Applications of Robust Optimization to Queueing and Inventory Systems

by

Alexander Anatolyevich Rikun

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

June 2011

Author .....
Sloan School of Management
May 16, 2011

Certified by .....
Dimitris Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center
Thesis Supervisor

Certified by .....
David Gamarnik
Associate Professor of Operations Research
Thesis Supervisor

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

Applications of Robust Optimization to Queueing and Inventory Systems

by

Alexander Anatolyevich Rikun

Submitted to the Sloan School of Management
on May 16, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

This thesis investigates the application of robust optimization in the performance analysis of queueing and inventory systems.

In the first part of the thesis, we propose a new approach for performance analysis of queueing systems based on robust optimization. We first derive explicit upper bounds on performance for tandem single class, multiclass single server, and single class multiserver queueing systems by solving appropriate robust optimization problems. We then show that these bounds derived by solving deterministic optimization problems translate to upper bounds on the expected steady-state performance for a variety of widely used performance measures such as waiting times and queue lengths. Additionally, these explicit bounds agree qualitatively with known results.

In the second part of the thesis, we propose methods to compute (s,S) policies in supply chain networks using robust and stochastic optimization and compare their performance. Our algorithms handle general uncertainty sets, arbitrary network topologies, and flexible cost functions including the presence of fixed costs. The algorithms exhibit empirically practical running times. We contrast the performance of robust and stochastic (s,S) policies in a numerical study, and we find that the robust policy is comparable to the average performance of the stochastic policy, but has a considerably lower standard deviation across a variety of networks and realized demand distributions. Additionally, we identify regimes when the robust policy exhibits particular strengths even in average performance and tail behavior as compared with the stochastic policy.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Professor of Operations Research
Co-Director, Operations Research Center

Thesis Supervisor: David Gamarnik
Title: Associate Professor of Operations Research

Acknowledgments

I would like to take a moment to mention and thank the people who have played an important role in my life over the last five years. This section alone can easily be the length of a Chapter. The amount of people who have played a positive role in my life over the last five years, along with their specific acts of kindness, are too many to recount. I have tried my best to name all of the people I can think of, but I apologize ahead of time to those I may have overlooked.

Firstly, I would like to thank my advisers - Dimitris Bertsimas and David Gamarnik. As a result of your guidance and support, I was able to grow and develop as a researcher and as a communicator. You helped me learn how to frame complex problems and think in big picture terms, and your committment to high quality is a true role model. Most importantly, I was able to grow not just on a professional, but on a personal level as well.

I never would have come to the ORC, if it were not for Chris Wiggins, Patrick Gallagher, and Ioannis Karatzas. Thank you for providing me with valuable advice, inspiration, and encouragement to pursue a Ph.D. Thank you Kostas Kardaras for your help and advice about graduate school. And thank you Burton Levine for your support and advice.

I would like to specially thank Gabriel Bitran and Eleni Pratsini for providing me with wonderful support, mentorship, and encouragement over the last few years. I am very fortunate that I was able to get to know you. I would like to thank Rama Ramakrishnan, Don Rosenfield, and Leonid Kogan for their valuable advice and encouragement.

90 % of this work was done in the Dewey Library, and I would like to thank the staff of Dewey for providing such a nice environment for thinking and working. A very special thank you to the staff of the ORC - Andrew, Laura, and Paulette. You provided me with wonderful help and advice in important times. I would also like to thank Patrick Jaillet for being a great co-director of the ORC.

Thank you to my colleagues and friends at the ORC including Matt Fontana, Maxime, Phil, Joline, Andre, Jason, Adam, Chaithanya, Nick, Shubham, Gareth, Shashi, Xin, Diana, Matthieu, Andy, Eric, Theo and all other pals in the ORC. Thank you to ORC alums Ilan Lobel, Margret Bjarnadottir, Doug Fearing, Dan Iancu, Karima Nigmatulina, Rajiv Menjoge, and Lavanya Marla for your friendship and advice in important times.

In addition to doing research and classwork, I was able to find some free time for other activies such as tennis and coffee. As a result, I have shared wonderful and engaging moments with Ruben Lobel, Nikos Trichakis, Jon Kluberg, Dave Goldberg, Adrian Becker, and Martin Quinteros. Thank you for your friendship. Working on projects with Nikos was awesome. Special thanks to Ruben, Nikos, and Martin for introducing me (multiple times) to fine South American meat specialties. A special thanks to Jon for his support over the years and for introducing me to Harvard Chabad. On the tennis front, a very special thanks to Julianne Dunkel, Charlie Maher, and Florin Ciocan for being great tennis partners and great pals. A special thanks to Dima Katz for his personal friendship and collaboration on work in Chapter 3. I learned a great deal from you.

My non-ORC friends including Andres Abin-Fuentes, Gene, Pasha Lavitas, Yura Tarasula, Nika Gelfand, Artem Stavisky, Dmytro Karabash, Rob Toth, and Jim Kincade made my last five years fulfilling and fun. Thank you for your friendship and for being part of my life. Thank you to Rabbis Posner, Gluckin, and Wiener for your friendship and guidance.

Thank you to my high school homeys Carl Wivagg, Vlad Turzhitsky, and Alex Gordon. I am truly lucky for your support and friendship. The bike trips, camping trips, math team, and business are great times in my life.

Thank you Dasha. I am really happy and lucky you decided to be a Shliach in Boston in 2009.

I would like to thank my family. Thank you to my aunts Galya and Liuda and my cousin Alena for your support and love. Thank you to my mom Vera, dad Anatoliy, and grandparents Bella, Mitya and David for providing me with encouragement and support in times that matter most. I am lucky and grateful for you.

Cambridge, May 2011 *Alexander Rikun*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this thesis is to propose a new method of analysis for queueing systems that leads to explicit upper bounds on performance and to investigate the effectiveness of this method of analysis to the performance of inventory systems. The key approach is to utilize robust optimization, a tractable method to optimize systems under uncertainty that has been widely developed in the last decade. For this reason, we discuss in Section 1.1 optimization under uncertainty and robust optimization, in particular. In Section 1.2, we provide an overview of the thesis and of our contributions.

1.1 Optimization Under Uncertainty

Capturing uncertainty in optimization problems provides a powerful modeling framework. Portfolio optimization, stochastic shortest paths, queueing systems, and revenue management are among the numerous potential problems that can be modeled as optimization problems under uncertainty. The downside is that barring very specific examples and models, optimization problems under uncertainty are hard and there are no general and simple tools for solving them. However, throughout the optimization literature there have been several major lines of research to tackle this area, and we outline some below.

One approach known as Stochastic Programming refers to methods that represent uncertain data through scenarios. These scenarios are generated as a result of assuming an underlying

probability distribution for the uncertain parameters. For instance, stochastic linear programming finds an optimal solution that produces the best average objective function value over all scenarios. One may then extend this approach to model multi-stage problems using techniques such as Benders decomposition or incorporate a notion of risk into the objective function. The book by Shapiro et al. (Shapiro, Dentcheva, and Ruszczynski 2009) is a standard reference on stochastic programming.

Dynamic Programming introduced by Richard Bellman is another method designed to deal with uncertain systems. For the most part, dynamic programming also models uncertainty with a probability distribution but is geared towards problems with multiple stages. The spirit of the approach is to solve the problem recursively - starting with the last stage. Often times, the major power of the Dynamic Programming approach is two-fold: it allows the user to prove that a particular policy or solution is optimal, or it allows the user to prove that the optimal policy has a special structure which may greatly reduce the search space for the optimal policy or give rise to good heuristics. We refer the reader to the seminal book by Bellman (Bellman 1957) and the books by Bertsekas (Bertsekas 1995).

Perhaps the greatest drawback of the Stochastic and Dynamic Programming approaches is that they suffer from the curse of dimensionality. In other words, barring specialized models, the solution time of these problems increases exponentially with the size of the problem (e.g. number of stages). Another method for incorporating uncertainty into optimization problems is known as Robust Optimization. For a review of robust optimization see the survey by Bertsimas et al. (Bertsimas, Brown, and Caramanis 2011) and the book by Ben-Tal et al. (Ben-Tal, Ghaoui, and Nemirovski 2009). Robust optimization provides a tractable framework for incorporating uncertainty into the optimization problem. Robust Optimization does not model uncertainty with a specific probability distribution, but instead models uncertainty with uncertainty sets (polyhedra or ellipsoids). The main impact of robust optimization is two fold: First, it allows the decision maker to include uncertainty information into the optimization problem, whereas other methods may fail completely due to tractability issues. Secondly, robust optimization provides a particular advantage for modeling in low data environments (forecast is poor) by avoiding strong assumptions about the underlying probability distribution for uncertain parameters.

Currently, robust optimization is a rapidly growing area of academic research both on the

theory and application fronts. Researchers are still trying to figure out the extent to which this framework can be applied to model real world problems. Additionally, robust optimization has a deep underlying connection to risk theory (Natarajan, Pachamanova, and Sim 2009; Bertsimas and Brown 2009) and as a result provides a tractable framework for which to model problems that have hitherto been attacked with traditional stochastic approaches. Thus, it is a very exciting and potentially rewarding pursuit to push the envelope of the robust optimization modeling framework to see how well one can model and capture complex random behavior (i.e. in queueing systems or finance) in a tractable manner (i.e. in a linear or quadratic program) that agrees qualitatively with probabilistic methods.

1.2 Thesis Overview and Contributions

This thesis is composed of three self-contained essays illustrating the applications of robust optimization approaches. Chapters 2 and 3 both deal with queueing systems and Chapter 4 with inventory theory. The motivation for the research is two fold:

- To use the robust optimization modeling framework for performance analysis in queueing systems. In particular, the goal is to compute bounds on performance measures for several types of queueing systems using robust optimization in a way that translates to meaningful bounds and insights for the underlying stochastic system.
- Understand the benefits and drawbacks of the policies and solutions to optimization problems with uncertainty based on the robust optimization approach as compared with tradititional stochastic optimization, particularly in the context of inventory systems.

We next give a brief overview of each chapter and its specific contributions below:

Chapter 2 considers the question of performance analysis of queueing systems. In this chapter, we propose a new performance analysis method, which is based on robust optimization. The basic premise of our approach is as follows: rather than assuming that the stochastic primitives of a queueing model satisfy certain probability laws, such as, for example, i.i.d. interarrival and service times distributions, we assume that the underlying primitives are deterministic and satisfy the *implications* of such probability laws. These implications take the form of simple linear

constraints, namely, those motivated by the Law of the Iterated Logarithm (LIL). Using this approach we are able to obtain performance bounds on some key performance measures. Furthermore, these performance bounds imply similar bounds in the underlying stochastic queueing models.

We demonstrate our approach on two types of queueing systems: Tandem Single Class (TSC) queueing network and the Multiclass Single Server queueing network. In both cases, using the proposed robust optimization approach, we are able to obtain *explicit* upper bounds on some steady-state performance measures. For example, for the case of TSC system we obtain a bound of the form

$$C(1-\rho)^{-1}\ln\ln((1-\rho)^{-1})$$

on the expected steady-state sojourn time, where C is an explicit constant and ρ is the bottleneck traffic intensity. This qualitatively agrees with the correct heavy traffic scaling of this performance measure up to the $\ln\ln((1-\rho)^{-1})$ correction factor.

Chapter 3 considers the performance analysis of the single class m-parallel server network ($GI/GI/m$) with general, but independent interarrival and service times. In particular, we apply the approach developed in Chapter 2 to address the question of computing waiting times and queueing lengths for the $GI/GI/m$ queueing system. Using this approach we are able to obtain *explicit* bounds on waiting times and queueing lengths of the form

$$C(1-\rho)^{-1}\ln\ln((1-\rho)^{-1})$$

that qualitatively agree with Kingman's bounds up to the the $\ln\ln((1-\rho)^{-1})$ correction factor. Additionally, we analyze the waiting time of the $GI/GI/m$ robust model in the Halfin-Whitt regime and compare to how it performs to traditional stochastic analysis. In particular, we explicitly construct and prove an upper and lower bound on the waiting time of the steady state customer in the robust $GI/GI/m$ system. These results indicate that as more servers are added to the system, the steady state customer in the robust $GI/GI/m$ system experiences a decline (upper bound result) in the expected waiting time that is similar to the steady state customer in the stochastic $GI/GI/m$ system. However, as more and more servers are added to the system, the stochastic steady state waiting time is driven to zero, while the robust steady

state waiting time remains strictly above zero.

Chapter 4 addresses the question of computing (s,S) policies in supply chain networks. In particular, we propose methods to compute (s,S) policies in supply chain networks using robust and stochastic optimization and compare their performance. Our algorithms handle general uncertainty sets, arbitrary network topologies, and flexible cost functions including the presence of fixed costs. The algorithms exhibit empirically practical running times. In a numerical study, we contrast the performance of robust and stochastic (s,S) policies, and we find that the robust policy is comparable to the average performance of the stochastic policy, but has a considerably lower standard deviation across a variety of networks and realized demand distributions. Additionally, we identify regimes when the robust policy exhibits particular strengths even in average performance and tail behavior as compared with the stochastic policy.

Chapter 2

Performance Analysis of Queueing Networks via Robust Optimization

2.1 Introduction

Performance analysis of queueing networks is one of the most challenging areas of queueing theory. The difficulty stems from the presence of network feedback, which introduces a complicated multidimensional structure into the stochastic processes underlying the key performance measures. Short of specialized cases, such as product form networks, which typically rely on Poisson arrival/exponential service time distributional assumptions, the problem is largely unresolved. Specifically, given the topological description of a queueing network and given the description of the underlying stochastic primitives such as interarrival and service times distributions, we do not have good tools for computing exactly or obtaining upper and lower bounds on key performance measures, such as, for example average queue lengths and waiting times. Some of results which provide non-asymptotic bounds on performance measures can be found in (Bertsimas, Paschalidis, and Tsitsiklis 1994), (Kumar and Kumar 1994), (Kumar and Morrison 2004), (Jin, Ou, and Kumar 1997), (Bertsimas, Gamarnik, and Tsitsiklis 1996), (Bertsimas and Nino-Mora 1999), all of which require Markovian (Poisson arrival/exponential service time) distributional assumptions. Moreover, some of these bounds become quite weak as traffic intensity (of some of the network

components) approach unity. For example, a bound of the form $O((1-\rho^*)^{-2})$ is obtained in (Bertsimas, Gamarnik, and Tsitsiklis 2001), where ρ^* is the bottleneck (real or virtual, see the reference) traffic intensity. The other references can lead to infinite upper bounds even in the cases where stationary distribution exists. The approaches in these papers also do not extend to the case of non-Markovian systems. As a consequence, most of the known performance analysis results are of an asymptotic nature, which apply to queueing networks in various limiting regimes, such as the heavy traffic regime (Harrison 1990),(Whitt 2002),(Chen and Yao 2001), large deviations methods (Ganesh, O'Connell, and Wischik 2004),(Shwartz and Weiss 1995), approximations by phase-type distributions (Kleinrock 1975),(Latouche and Ramaswami 1987).

In this thesis, we partially fill this gap by developing a new performance analysis approach based on robust optimization methods. The theory of robust optimizaiton emerged recently as a very successful and constructive approach for the analysis of certain stochastic modeling problems (Soyster 1973),(Ben-Tal and Nemirovski 1998), (Ben-Tal and Nemirovski 1999), (Bertsimas and Sim 2004). The main premise of our approach in the queueing context is that, rather than assuming probabilistic laws for the underlying stochastic primitives, such as, for example, i.i.d. interarrival and service times, we consider a deterministic queueing model and we will assume only the *implications* of these laws. Specifically we consider implications of the Law of the Iterated Logarithm (LIL). The objective is to find laws which on the one hand hold in the underlying stochastic queueing model and, on the other hand, lead to linear constraints in the formulation of the robust optimization problem, and LIL accomplishes this. We illustrate our approach using two queueing models, namely the Tandem Single Class (TSC) queueing system operating under the First-In-First-Out (FIFO) scheduling policy, and the Multiclass Single Server (MCSS) queueing system operating under an arbitrary work-conserving policy. Motivated by the LIL, we consider constraints of the form $\sum_{1\leq i\leq k} U_i \leq \lambda^{-1}k + \Gamma\sqrt{k\ln\ln k}$, for all $k \geq 1$. Here $(U_k, k \geq 1)$ is any of the stochastic primitives of the underlying queueing system, such as, for example, the sequence of interarrival times and λ stands for the rate of this stochastic primitive. Using these bounds, we derive *explicit* bounds on some performance measures such as sojourn time in the TSC system, namely, the time it takes for a job to be processed by all the servers, and the virtual workload (virtual waiting time) in the MCSS system, namely, the time required to clear the current backlog in the absence of future arrivals. In both models we derive upper bounds on the aforementioned performance measures for the

corresponding deterministic counterpart models and prove that similar bounds also hold for the same performance measures in the underlying stochastic models. In both cases the bounds are of the order $O(\frac{1}{1-\rho}\ln\ln\frac{1}{1-\rho})$, where ρ is the (bottleneck for the case of TSC model) traffic intensity. This matches the correct $O(\frac{1}{1-\rho})$ order short of $\ln\ln((1-\rho)^{-1})$ error. While the technical derivation of these bounds is involved, the conceptual approach is very simple. An interesting distinction of our approach from other robust optimization type results is that our results are explicit, as opposed to numeric results one typically obtains from the formulating and solving a robust optimization model. These explicit bounds however, come at a price of not caring much for the constants corresponding to the leading coefficient. In order to keep things simple we sometimes use very crude estimates for such constants.

Our approach bears similarity with some earlier works in the queueing literature. Specifically, the pioneering work of Cruz (Cruz 1991a), (Cruz 1991b) used a similar non-probabilistic approach to performance analysis by deriving bounds based on placing deterministic constraints on the flow of traffic called "burstiness constraints". The method could be applied to fairly general network topologies and led to more research in the area. In (Gallager and Parekh 1993), (Gallager and Parekh 1994), tighter performance bounds were obtained assuming a "Leaky Bucket" rate admission control from (Turner 1986) and particular service disciplines. In addition, there is some similarity between the philosophy of our approach and the *adversarial queueing network approach* (Andrews, Awerbuch, Fernandez, Kleinberg, Leighton, and Liu 1996), (Borodin, Kleinberg, Raghavan, Sudan, and Williamson 2001), (Goel 1999), (Gamarnik 2003), (Gamarnik 2000), which emerged in the last decade in the computer science literature and also replaces the stochastic assumptions with adversarial deterministic ones. The deterministic constraints used in the aforementioned works are of the form of $A(t) \leq \lambda t + B$ where $A(t)$ is the number of external arrivals into the queueing system up to time t and λ represents the arrival rate. As it turns out, these types of assumptions are too restrictive from the probabilistic point of view and do not lead to bounds on the underlying stochastic network: observe that every renewal process $A(t)$ arising from an i.i.d. sequence with positive variance violates this assumption almost surely for every B for large enough t. As we demonstrate in this chapter, the constraints motivated by the LIL, namely $A(t) \leq \lambda t + B\sqrt{t\ln\ln t}$, can indeed be served to obtain performance bounds, which can be translated into the underlying stochastic network. In fact, the key contribution of our approach is that the deterministic constraints we place on the

service and arrival processes are rich enough to lead to stochastic results. The results based on "Leaky Buckets", bounded burstiness and adversarial queueing theory address very general queueing networks. It would be an interesting research project to extend our results based on robust optimization to these general network structures.

The rest of the chapter is structured as follows. In the following section we describe two queueing models under the consideration, namely the tandem single class queueing network and the single server multiclass queueing network, as well as their robust optimization counterpart models. Our main results, namely the performance bounds in robust optimization type queueing systems and their implications for stochastic queueing systems are stated in Section 3.3. The proofs of our main results are in Sections 2.4 and 2.5. Some concluding thoughts and directions for further research are outlined in Section 2.6. Several technical results necessary for proofs of main theorems are delayed untill the Appendix section.

We close this section with some notational conventions. ln stands for the logarithm with natural base. The notation $(x)^{\frac{1}{2}}$ for a non-negative vector $x \in \mathbb{R}^d$ means applying the square root operator coordinate-wise: $(x)^{\frac{1}{2}} = (x_i^{\frac{1}{2}}, 1 \leq i \leq d)$. A^T denotes a transposition operator applied to the matrix A.

2.2 Model description

We now describe the two queueing models analyzed in this chapter, both very well studied models in the literature. We begin by describing these models in the stochastic setting, and then we describe their deterministic robust optimization counterparts.

2.2.1 A tandem single class (TSC) queueing network. Stochastic model

The model is a tandem of single servers $S_1, \ldots, S_J$ processing a single stream of jobs arriving from outside and requiring services at $S_1, \ldots, S_J$ in this order. The jobs arrive from outside according to an i.i.d. renewal process. Let $U_1, U_2, U_3, \ldots$ denote i.i.d. interarrival times with a common distribution function $F_a(t) = \mathbb{P}(U_1 \leq t)$, where U_1 is the time at which the first job

arrives. The external arrival rate is defined to be $\lambda \triangleq 1/\mathbb{E}[U_1]$ and the variance of U_1 is denoted by σ_a^2.

The jobs arriving externally join the buffer corresponding to server S_1 where they are served using First-In-First-Out (FIFO) scheduling policy. We assume that all buffers are of infinite capacity. After service completion, jobs are routed to the buffer of server S_2, where they are also served using FIFO scheduling policy, then they are routed to servers S_3, S_4, etc. After service completion in server S_J the jobs depart from the network. Let V_k^j denote the service time requirement for job k in server j. We assume that the sequence $(V_k^j, k \geq 1)$ is i.i.d. for each j, and is independent from all other random variables in the network. The distribution of the service time in server j is $F_{s,j}(t) = \mathbb{P}(V_1^j \leq t), t \geq 0$. The service rate in server S_j is defined to be $\mu_j \triangleq 1/\mathbb{E}[V_1^j]$, and we denote by $\mu_{\min} = \min_{1 \leq j \leq J} \mu_j$ the rate of the slowest server. $\sigma_{s,j}^2$ denotes the variance of V_1^j for each $j = 1, \ldots, J$. The traffic intensity in server S_j is defined to be $\rho_j = \lambda/\mu_j$, and the bottleneck traffic intensity is defined to be $\rho^* = \max_j \rho_j = \lambda/\mu_{\min}$.

Denote by W_k^j the waiting time experienced by job k in server j not including the service time V_k^j. Let $W_k = \sum_j (W_k^j + V_k^j)$ be the sojourn time of the job k. Namely, this is time between the arrival of job k into buffer 1 and service completion of the same job in buffer J. Denote by $Q_j(t)$ the queue length in server j (the number of jobs in buffer j) at time t. We assume that initially all queues are empty: $Q_j(0) = 0, 1 \leq j \leq J$, although most of our results can either be easily adopted to the case of non-zero queues at time zero, or apply to the steady-state measures where the initializations of the queues is irrelevant. Let I_k^j denote the idle time of server j in between servicing jobs $k-1$ and k for $k = 2, ..., N$. We define $I_1^j = 0 \ \ \forall j = 1, \ldots, J$.

The model just described will be denoted by TSC(St) (Tandem Single Class Stochastic) for short. It is known (Sigman 1990),(Dai 1995),(Dai and Meyn 1995),(Chen and Yao 2001) that as long as $\rho^* < 1$, and some additional mild conditions hold, such as finiteness of moments, TSC(St) is stable and the stochastic processes underlying the performance measures such as queue lengths, workloads, sojourn times are mixing. Namely, these processes are positive Harris recurrent (Dai 1995),(Meyn and Tweedie 1993), and the transient performance measures converge to the (unique) steady-state performance measures both in distributions and in moments. Computing these performance measures is a different matter, however. We denote by W_∞^j, W_∞ the steady state versions of the random variables W_k^j, W_k. Thus provided that $\rho^* < 1$ and some

additional technical assumptions hold, we have

$$\lim_{n\to\infty} \mathbb{E}[W_n] = \mathbb{E}[W_\infty]. \tag{2.1}$$

We will assume that $\rho^* < 1$ holds without explicitly stating it. Rather than describing the assumptions required to make (3.1) true, we will simply assume when stating our results that (3.1) holds as well.

2.2.2 A multiclass single server (MCSS) queueing system. Stochastic model

We now describe our second queueing model. Consider a single server queueing system which processes J classes of jobs. The jobs of class $j = 1, 2, \ldots, J$ arrive from outside according to a renewal process with i.i.d. interarrival times $U_k^j, k \geq 1$ and distribution function $F_{a,j}(t) = \mathbb{P}(U_1^j \leq t)$. The arrival rate for class j jobs is $\lambda_j \triangleq 1/\mathbb{E}[U_1^j]$. It is possible that some classes j do not have an external arrival process, in which case $U_k^j = \infty$ almost surely and $\lambda_j = 0$. Let $\sigma_{a,j}^2$ be the variance of U_1^j. The sequences $(U_k^j, k \geq 1)$ are also assumed to be independent for different j. Let $\lambda = (\lambda_j)$ denote the J-vector of arrival rates. We let $\lambda_{\max} = \max_{1\leq j\leq J} \lambda_j$ and $\lambda_{\min} = \min_{1\leq j\leq J} \lambda_j$. We let $A(t) = (A_j(t))$ denote the vector of cumulative number of external arrivals up to time t where $A_j(t) = \max\{k : \sum_{1\leq i\leq k} U_i^j \leq t\}$.

The jobs corresponding to class j are stored in buffer B_j until served. As in the single class case, we assume all buffers are of infinite capacity. The service time for the k-th job arriving to buffer B_j is denoted by V_k^j and the sequence $(V_k^j, k \geq 1)$ is assumed to be i.i.d. with a common distribution function $F_{s,j}(t) = \mathbb{P}(V_1^j \leq t)$. Additionally, these sequences are assumed to be independent for all j and independent from the interarrival times sequences $(U_k^j, k \geq 1)$. The average service time for class j is $m_j \triangleq E[V_1^j]$ and the service rate is $\mu_j \triangleq 1/\mathbb{E}[V_1^j]$. $\sigma_{s,j}^2$ denotes the variance of V_1^j. Let $\bar{m} = (m_j)$ denote the J-vector of average service times and let $\mu = (\mu_j)$ be the J-vector of service rates. Let M denote the diagonal matrix with j-th entry equal to μ_j and let $\mu_{\max} = \max_{1\leq j\leq J} \mu_j$.

We assume that the jobs in buffer B_j are served using FIFO rule, but prioritizing jobs between different buffers B_j is done using some scheduling policy θ. The only assumption we

make about θ is that it is a *work-conserving* policy. Namely, the server is working full time as long as there is at least one job in one of the buffers B_j, $1 \leq j \leq J$. The only performance measure we will consider is the workload (defined below) for which it is well known that the details of the scheduling policy are unimportant for us, as long as the policy is work-conserving.

The routing of jobs after service completions is determined using a routing matrix P, which is an J by J 0,1 matrix $P = (P_{i,j}, 1 \leq i,j \leq J)$. It is assumed that $\sum_j P_{i,j} \leq 1$ for each i. (Namely, the sum is either 1 or 0). Upon service completion in buffer B_i, the job of class i is routed to buffer j if $P_{i,j} = 1$. Otherwise, if $\sum_j P_{i,j} = 0$, the jobs of class i leave the network. It is assumed that $P^n = 0$ for some positive integer n. It is easy to see that this condition is equivalent to saying that all jobs eventually leave the network.

It is known (Chen and Yao 2001) that the traffic equation $\bar{\lambda}_i = \lambda_i + \sum_{1 \leq j \leq J} \bar{\lambda}_j P_{j,i}$ has a unique solution $\bar{\lambda} = (\bar{\lambda}_j)$ given simply as $\bar{\lambda} = [I - P^T]^{-1}\lambda$, where I is the J by J identity matrix. Let $\bar{\lambda}_{\max} = \max_j(\bar{\lambda}_j)$ (observe that $\lambda_j \leq \bar{\lambda}_j$ for every j and hence $\bar{\lambda}_{\max} \geq \lambda_{\max}$). Let $\bar{A}(t) = (\bar{A}_j(t))$ denote the vector of number of arrivals by time t that will eventually route to server j: $\bar{A}_j(t) = e_j^T(I + (P^T)^1 + (P^T)^2 + \ldots)A(t) = e_j^T[I - P^T]^{-1}A(t)$ and e_j denotes the $j-th$ unit vector.

The traffic intensity vector is defined to be $\bar{\rho} = M^{-1}\bar{\lambda} = M^{-1}[I - P^T]^{-1}\lambda$. The traffic intensity of the entire server is $\rho = e^T\bar{\rho}$, where e is the J vector of ones. Let $Q_j(t)$ denote the queue length in buffer j at time t, let $Q(t) = (Q_j(t))$. We assume that $Q(0) = 1$. As for the case of TSC model, our results can be extended to the case $Q(0) \geq 0$, but for the results regarding steady-state behavior, the initialization of queues is irrelevant. Denote by W_k^j the waiting time of the k-th job arriving into buffer j. We let W_t denote the workload at time t. Namely, W_t is the time required to process all the jobs present in the system at time t, in the absence of the future arrivals. Note that W_t is also the virtual waiting time at time t when the scheduling policy is FIFO. Observe that if t_0 marks the beginning of a busy period and t_1 belongs to the same busy period (namely, the server was working continuously during the time interval $[t_0, t_1]$), then almost surely

$$W_{t_1} = \sum_{i=\bar{A}_1(t_0)}^{\bar{A}_1(t_1)} V_i^1 + \ldots + \sum_{i=\bar{A}_J(t_0)}^{\bar{A}_J(t_1)} V_i^J - (t_1 - t_0). \tag{2.2}$$

The model described above is denoted by MCSS(St) (Multiclass Single Server Stochastic) for short. It is known (Dai 1995) that if $\rho < 1$, and some additional technical assumption on interarrival and service time distributions hold then MCSS(St) is stable and enters the steady state in the same sense as described for the tandem queueing network. While in this case the steady-state distribution of many performance measures usually depends on the details of work-conserving policy used, the steady-state distribution of the workload does not depend on the policy, as we have discussed above. Let W_∞ denote the workload in steady state, and let B_∞ and I_∞ denote the steady-state duration of the busy and idle periods, respectively. Additionally, denote by $I_0, B_1, I_1, B_2, I_2, \ldots$ the alternating sequence of the lengths of the busy and idle periods of the MCSS(St) system, assuming that time zero initiates a busy period. Under the same technical assumptions as above the following ergodic properties hold almost surely:

$$\lim_{t\to\infty} \frac{\int_0^t W_s ds}{t} = \mathbb{E}[W_\infty], \tag{2.3}$$

$$\lim_{n\to\infty} \frac{\sum_{1\le i\le n} B_i}{n} = \mathbb{E}[B_\infty], \tag{2.4}$$

$$\lim_{n\to\infty} \frac{\sum_{1\le i\le n} I_i}{n} = \mathbb{E}[I_\infty], \tag{2.5}$$

$$\lim_{n\to\infty} \frac{\sum_{1\le i\le n} B_i^2}{n} = \mathbb{E}[B_\infty^2]. \tag{2.6}$$

We denote by $n(t)$ the number of busy periods that have been initiated up to time t. Mathematically, we define $n(t)$ to satisfy $\sum_{1\le i\le n(t)-1}(B_i + I_i) < t \le \sum_{1\le i\le n(t)}(B_i + I_i)$. When $t \in [\sum_{1\le i\le n(t)-1}(B_i+I_i), \sum_{1\le i\le n(t)-1}(B_i+I_i)+B_{n(t)}]$, t falls on a busy period and using the definition of $n(t)$, we have $W(t) \le B_{n(t)}$. When $t \in [\sum_{1\le i\le n(t)-1}(B_i+I_i)+B_{n(t)}, \sum_{1\le i\le n(t)}(B_i+I_i)]$, t falls on idle period $I_{n(t)}$ and hence $W(t) = 0$. We let τ_i denote the beginning of the i-th busy period. This implies

$$\frac{\int_0^t W(s)ds}{t} = \frac{\sum_{i=1}^{n(t)} \int_{\tau_i}^{\min(\tau_i+B_i,t)} W(s)ds}{t} \le \frac{\sum_{1\le i\le n(t)} B_i^2}{\sum_{1\le i\le n(t)-1}(B_i + I_i)}$$

If (2.3),(2.4),(2.5) and (2.6) hold, then we also obtain

$$\mathbb{E}[W_\infty] \leq \frac{\mathbb{E}[B_\infty^2]}{\mathbb{E}[B_\infty] + \mathbb{E}[I_\infty]} \leq \frac{\mathbb{E}[B_\infty^2]}{\mathbb{E}[B_\infty]}. \tag{2.7}$$

This bound will turn useful when we apply our results for robust optimization models to the underlying stochastic model. As for the TSC case, we assume from now on $\rho < 1$. Rather than listing the assumptions leading to ergodic properties (2.3),(2.4),(2.5) and (2.6) we assume when stating our results, that the stochastic process W_t enters the steady-state as $t \to \infty$ and that the properties (2.3),(2.4),(2.5) and (2.6) holds almost surely.

2.2.3 Robust optimization type queueing systems

We now describe deterministic robust optimization type counterparts of the two stochastic queueing models described in the previous subsections.

We begin with TSC model and describe the corresponding model which we denote by TSC(RO) (Tandem Single Class Robust Optimization). The description of the network topology is the same as for TSC(St). However, it is not assumed that U_k, V_k^j and, as a result $Q(t), W_k^j, W_k$ are random variables. Rather we assume that these quantities are *arbitrary* subject to certain linear constraints detailed below. Additionally, we assume that the system starts empty $Q(0) = 0$ and only n jobs go through the system.

Specifically, consider a sequence of non-negative deterministic interarrival and service times $(U_k, 1 \leq k \leq n), (V_k^j, 1 \leq k \leq n), 1 \leq j \leq J$. Let

$$\phi(x) = \begin{cases} \sqrt{x \ln \ln x}, & x \geq e^e; \\ 1, & x < e^e. \end{cases} \tag{2.8}$$

We assume that there exist λ, Γ_a and $\mu_j, \Gamma_{s,j} \geq 0, 1 \leq j \leq J$ such that

$$\Big| \sum_{k+1 \leq i \leq n} U_k - \lambda^{-1}(n-k) \Big| \leq \Gamma_a \phi(n-k), \qquad k = 0, 1, \ldots, n-1, \tag{2.9}$$

$$\Big| \sum_{k+1 \leq i \leq n} V_i^j - \mu_j^{-1}(n-k) \Big| \leq \Gamma_{s,j} \phi(n-k), \qquad k = 0, 1, \ldots, n-1,\ j = 1, 2, \ldots, J. \tag{2.10}$$

It is because we need to consider tail summation $\sum_{k+1\leq i\leq n}$ we assume that only n jobs going through the system, though we will be able to apply our results in the stochastic setting where infinite number of jobs pass through the system. Let $\Gamma = \max(\Gamma_a, \Gamma_{s,j})$. Borrowing from the robust optimization literature terminology (Bertsimas and Sim 2004), the parameters $\Gamma_a, \Gamma_{s,j}, \Gamma$ are called *budgets of uncertainty.* Note, that the values $U_k, V_k^j, k \geq 1$ uniquely define the corresponding performance measures $Q_j(t), W_k^j, W_k, k = 1, \ldots, n$. There is no notion of steady state quantities $Q_j(\infty), W_\infty$ for the model TSC(RO). The motivation for constraints (2.9) and (2.10) comes from the Law of the Iterated Logarithm, and we discuss the connection in a separate subsection.

We denote the robust optimization counterpart of the MCSS(St) model by MCSS(RO). In this case it turns out to be convenient to consider infinite sequence of jobs. Thus consider infinite sequences of deterministic non-negative values $(U_k^j, k \geq 1), (V_k^j, k \geq 1), 1 \leq j \leq J$. It is assumed that values $\lambda_j, \mu_j, \Gamma_{a,j}, \Gamma_{s,j} \geq 0,\ 1 \leq j \leq J$ exist such that

$$\Big|\sum_{1\leq i\leq k} U_k^j - \lambda_j^{-1}k\Big| \leq \Gamma_{a,j}\phi(k), \quad k = 1, 2, \ldots,\ j = 1, 2, \ldots, J, \tag{2.11}$$

$$\Big|\sum_{1\leq i\leq k} V_i^j - \mu_j^{-1}k\Big| \leq \Gamma_{s,j}\phi(k), \quad k = 1, 2, \ldots,\ j = 1, 2, \ldots, J. \tag{2.12}$$

For convenience we assume that at time zero the system begins with exactly one job in every class $j = 1, \ldots, J$: $Q_j(0) = 1$. Then the first after time zero external arrival into buffer j occurs at time U_1^j. As before, we let $\Gamma = \max(\Gamma_{a,j}, \Gamma_{s,j})$.

For technical reasons, we also assume that Γ in TSC(RO), MCSS(RO) constraints satisfies

$$\lambda\Gamma \geq e^{2e} \quad \text{and} \quad \min_j \lambda_j\Gamma \geq e^{2e}, \quad \text{respectively.} \tag{2.13}$$

2.2.4 The Law of the Iterated Logarithm

One of the cornerstones of the probability theory is the Law of the Iterated Logarithm (LIL) (Chung 2001), which states that given a i.i.d. sequence of random variables $X_1, \ldots, X_n, \ldots$ with

zero mean and finite variance σ, the following holds almost surely,

$$\limsup_{n\to\infty} \frac{\sum_{1\le k\le n} X_k}{\sigma\sqrt{2n\ln\ln n}} = 1, \quad \liminf_{n\to\infty} \frac{\sum_{1\le k\le n} X_k}{\sigma\sqrt{2n\ln\ln n}} = -1.$$

The LIL extends immediately to non-zero mean i.i.d. sequences by subtracting $n\mathbb{E}[X_1]$ from $\sum_{1\le k\le n} X_k$. Furthermore, LIL implies (in the case of zero-mean variables) that

$$\Gamma_{\rm LIL} \triangleq \sup_{n\ge 1} \frac{|\sum_{1\le k\le n} X_k|}{\sigma\sqrt{2}\phi(n)} < \infty, \tag{2.14}$$

where ϕ is defined in (3.2). Note that $\Gamma_{\rm LIL}$ is a random variable. Thus when we consider stochastic queueing models such as TSC(St) or MCSS(St), the constraints (2.9),(2.10),(2.11),(2.12) hold with probability one, with $\Gamma = \sqrt{2}\Gamma_{\rm LIL}\sigma$, where $\Gamma_{\rm LIL}$ is defined in (2.14) for the corresponding random sequence. Specifically, let $\Gamma_a = \Gamma_{a,\rm LIL} = \Gamma_{\rm LIL}$ and $\sigma = \sigma_a$, when $X_k = U_{n-k} - \lambda^{-1}, 0 \le k \le n-1$ and U_k is the sequence of interarrival times in the TSC(St) model. Similarly define $\Gamma_{s,j} = \Gamma_{s,j,\rm LIL}$ when $X_k = V^j_{n-k} - \mu_j^{-1}, 0 \le k \le n-1, 1 \le j \le J$. Observe, that for $\Gamma_a, \Gamma_{s,j}$ thus defined, the constraints (2.9),(2.10) hold for an infinite sequences of jobs (that is jobs which would have indices $-1, -2, \ldots$), even though we need it only for the first n jobs. For the MCSS(St) model define $\Gamma_{a,j} = \Gamma_{a,j,\rm LIL}, \Gamma_{s,j} = \Gamma_{s,j,\rm LIL}$ corresponding to the sequences $U^j_k - \lambda_j^{-1}, V^j_k - \lambda_j^{-1}, k \ge 1$, respectively. We obtain

Proposition 2.1 *Constraints (2.9),(2.10),(2.11),(2.12) hold with probability one for* $\Gamma_a = \sqrt{2}\Gamma_{a,LIL}\sigma_a$, $\Gamma_{s,j} = \sqrt{2}\Gamma_{s,j,LIL}\sigma_{s,j}$, $\Gamma_{a,j} = \sqrt{2}\Gamma_{a,j,LIL}\sigma_{a,j}$, *and* $\Gamma_{s,j} = \sqrt{2}\Gamma_{s,j,LIL}\sigma_{s,j}$, *respectively, where* $\Gamma_{\cdot,\cdot,LIL}$ *is defined in (2.14) for the corresponding sequence.*

As a conclusion, for *every* property derivable on the basis of these constraints in our deterministic robust optimization queueing network models, such as, for example, bounds on the sojourn time of the n-th job in TSC, the *same property* applies with probability one for the underlying stochastic network. This observation underlies the main idea of the work.

2.3 Main results

In this section we state our main results on the performance bounds for robust optimization type queueing networks TSC(RO) and MCSS(RO), and the implications of our results for their stochastic counterparts TSC(St) and MCSS(St). We begin with TSC(RO) with the goal of obtaining a bound on the sojourn time.

Theorem 2.2 *The sojourn time of the n-th job in the TSC(RO) queueing system with constraints (2.9),(2.10) satisfies*

$$W_n \leq \frac{7J^2\Gamma^2\lambda}{1-\rho^*} \ln \ln \frac{J\lambda\Gamma}{1-\rho^*} + J\lambda^{-1}. \tag{2.15}$$

Observe that the bound on the sojourn time is explicit. It is expressed directly in terms of the primitives of the queueing system such as arrival and service rates. Observe also that the upper bound is independent from n. One can think of this bound as a "steady-state" bound on the sojourn time in the robust optimization model of the TSC system. Additionally, the constant Γ^2 is related to the "variances" of interarrival and service times viz a vi the LIL (2.14). It is known that in the stochastic GI/GI/1 queueing system the expected waiting time in steady state is approximately $(\sigma_a^2+\sigma_s^2)/(2\lambda(1-\rho))$, when the system is in heavy traffic, namely $\rho \to 1$. Namely, the expected waiting time depends linearly on the variances of interarrival and service time. Our bound (2.15) is thus consistent with this type of dependence. On the other hand, unfortunately, our bound depends quadratically on the number of servers J, whereas the correct dependence is known to be linear, at least in some special cases (Reiman 1984),(Gamarnik and Zeevi 2006).

The bound above does not have a correct $O((1-\rho^*)^{-1})$ scaling, which is known to be correct from the heavy-traffic theory perspective (Reiman 1984),(Gamarnik and Zeevi 2006). However, the correction factor is a very slowly growing function ln ln. The upshot is that we can use this bound to obtain a bound on W_n and W_∞ in the underlying stochastic system. This is what we do next.

Corollary 2.3 *For every $n \geq 1$ the sojourn time of the n-th job in the TSC(St) queueing*

network satisfies

$$\mathbb{E}[W_n] \leq \mathbb{E}\Big[\frac{7J^2\Gamma^2\lambda}{1-\rho^*} \ln\ln \frac{J\lambda\Gamma}{1-\rho^*}\Big] + J\lambda^{-1}. \tag{2.16}$$

where $\Gamma = \max_j(\sqrt{2}\sigma_a\Gamma_{a,LIL}, \sqrt{2}\sigma_{s,j}\Gamma_{s,j,LIL}, e^{2e}\lambda^{-1})$. *If in addition the assumption (3.1) holds then*

$$\mathbb{E}[W_\infty] \leq \mathbb{E}\Big[\frac{7J^2\Gamma^2\lambda}{1-\rho^*} \ln\ln \frac{J\lambda\Gamma}{1-\rho^*}\Big] + J\lambda^{-1}. \tag{2.17}$$

Proof. We first assume Theorem 2.2 is established. Note, in the context of the stochastic system, both W_n and Γ in Theorem 2.2 are random variables. We take $\Gamma = \max_j(\sqrt{2}\sigma_a\Gamma_{a,\mathrm{LIL}}, \sqrt{2}\sigma_{s,j}\Gamma_{s,j,\mathrm{LIL}}, e^{2e}\lambda^{-1})$ to satisfy (3.6), where $\Gamma_{\cdot,\cdot,\mathrm{LIL}}$ is defined in (2.14) for the corresponding sequence. Applying Proposition 2.1 we have that (2.15) holds with probability one for the underlying stochastic network. The bound (2.16) now follows from taking expectations of both sides of (2.15). The bound (2.17) follows from applying (3.1) to (2.16). □

We now turn our attention to the MCSS queueing model. Our approach for deriving a bound on the workload is based on first obtaining an upper bound on the duration of the busy period. Thus, we first give a bound on the duration of the busy period and then turn to the workload. Recall our assumption $Q(0) = 1$, though our results can readily be extended to the general case of $Q(0) \geq 0$. Thus, time $t = 0$ marks the beginning of a busy period.

Theorem 2.4 *Given a MCSS(RO) queueing system with constraints (2.11),(2.12), let B be the duration of the busy period initiated at time 0. Then*

$$B \leq \frac{5(4J+3)^2\bar{\lambda}^3_{\max}\Gamma^4}{(1-\rho)^2} \ln\ln \frac{2(4J+3)\bar{\lambda}^2_{\max}\Gamma^2}{1-\rho}, \tag{2.18}$$

$$\text{and} \quad \sup_{0\leq t\leq B} W(t) \leq \frac{2(4J+3)^2\bar{\lambda}^3_{\max}\Gamma^4}{1-\rho} \ln\ln \frac{(4J+3)\bar{\lambda}^2_{\max}\Gamma^2}{1-\rho} + \Gamma + 3\bar{\lambda}^2_{\max}\Gamma^3. \tag{2.19}$$

While the bound (2.19) corresponds to the maximum workload during a given busy period, the actual value of the bound does not depend on the busy period length explicitly. As it will

become apparent from the proof, we use the same technique for obtaining a bound simultaneously on the duration of the busy period and maximum workload during the busy period. Let us now discuss the implications of these bounds for the underlying stochastic model MCSS(St).

Corollary 2.5 *Given a MCSS(St) model, suppose the relations (2.3),(2.4),(2.5) and (2.6) hold. Then*

$$\mathbb{E}[B_\infty] \leq \mathbb{E}\Big[\frac{5(4J+3)^2\bar{\lambda}_{\max}^3\Gamma^4}{(1-\rho)^2}\ln\ln\frac{2(4J+3)\bar{\lambda}_{\max}^2\Gamma^2}{1-\rho}\Big], \tag{2.20}$$

$$\mathbb{E}[W_\infty] \leq \mathbb{E}\Big[\frac{25(4J+3)^4\bar{\lambda}_{\max}^6\mu_{\max}\Gamma^8}{(1-\rho)^4}\Big(\ln\ln\frac{2(4J+3)\bar{\lambda}_{\max}^2\Gamma^2}{1-\rho}\Big)^2\Big], \tag{2.21}$$

where $\Gamma = \max_j(\sqrt{2}\sigma_{a,j}\Gamma_{a,j,LIL}, \sqrt{2}\sigma_{s,j}\Gamma_{s,j,LIL}, e^{2e}\lambda_{\min}^{-1})$.

Unfortunately, in this case the scaling of our bounds as $\rho \to 1$ deviates significantly from the correct behavior. From the heavy traffic theory (Dai and Kurtz 1995), the correct behavior for the steady-state workload should be $O((1-\rho)^{-1})$. As for the steady-state busy period, the theory of M/G/1 queueing system (Kleinrock 1975) suggests the behavior $O((1-\rho)^{-\frac{3}{2}})$ as opposed to $O((1-\rho)^{-2}\ln\ln(1-\rho)^{-1})$ which we obtain. On the positive side, however, we managed to obtain explicit bounds on the performance measures which are expressed directly in terms of the stochastic primitives of the model, which we do not believe was possible using prior methods. We leave it as an interesting open problem to derive the performance bounds based on the robust optimization technique, which lead to the correct scaling behavior as $\rho \to 1$.

While the proofs of our main results are technically involved, conceptually they are not complicated. Before we turn to formal proofs, in order to help the reader, we outline below informally some of the key proof steps for our results.

For the TSC queueing network we first replace the constraints (2.9),(2.10) with more general constraints, see (2.22) and (2.23) below. Our results for the TSC network rely mostly on the Lindley's type recursion which in a single server queueing system recursively represents in the waiting time of the n-th job in terms of the interarrival and service times of the first n jobs. It is classical result of the queueing theory that this waiting time can be thought of as maximum of a random walk, with steps equalling in distribution to the difference between the interarrival

and service times. We derive a similar relation in the form of a bound on the sojourn time of the n-th job in the TSC network. This bound is given in Theorem 2.6. Then we view this bound as an optimization problem and obtain a bound on the objective value by proving the concavity of the objective function and substituting explicit bounds from constraints (2.9),(2.10).

Our proofs for the MCSS queueing system rely on the relation (2.2). Namely, we take advantage of the fact that the workload is depleted with the unit rate during the busy period. Then we take advantage of the constraints (2.11),(2.12) to show that in the MCSS(RO) system the workload at time t during the busy period can be upper bounded by an expression of the form $-at+b\sqrt{t\ln\ln t}+c$ with strictly positive a, b. It is then not hard to obtain an explicit estimated t_0 such that this expression is negative for $t > t_0$. Since this expression is an upper bound on a non-negative quantity (workload), then the duration of the busy period cannot be larger than t_0. This leads to an upper bound on the duration of the busy period in the MCSS(RO) system. In order to obtain a bound on the workload, we again take advantage of (2.2) and further obtain explicit upper bounds on the terms involving the sums of service times. We show that the workload at time t is at most $-at + b\sqrt{t\ln\ln t} + c$. We then obtain an upper bound on the workload during the busy period by obtaining explicit bounds on $\max_{t\geq 0} -at + b\sqrt{t\ln\ln t} + c$.

Our derivation of the bounds for the stochastic model MCSS(St) relies on the ergodic representation (2.3). We consider a modified system in which each busy period is initiated with simultaneous arrival of one job into *every* buffer j. This leads to a alternating renewal process with alternating i.i.d. busy and idle periods. We then obtain a bound on the steady-state workload in terms of the second moment of the busy period in the modified queueing system, using the renewal theory type arguments. It is this necessity to look at the second moment of the busy period which leads to a conservative scaling $O\Big((1-\rho)^{-4}(\ln\ln(1-\rho)^{-1})^2\Big)$ in our bound (2.21) on the steady-state workload.

2.4 Tandem single class queueing system: proof of Theorem 2.2

In order to prove Theorem 2.2 we first generalize constraints (2.9),(2.10) and obtain a method for bounding W_n under more general uncertainty assumptions.

2.4.1 General upper bound on the sojourn times

Given a sequence of non-negative real values $\Gamma^j_{\min}(k), \Gamma^j_{\max}(k)$ $1 \le j \le J, 1 \le k \le n$, $\Gamma_{\min}(k), \Gamma_{\max}(k)$ $1 \le k \le n$, we consider the set of all sequences of service times and inter-arrival times $(V^j_i), (U_i)$ $j = 1, \ldots, J$, $i = 1, \ldots, n$ satisfying for all $k = 1, \ldots, n$

$$\Gamma^j_{\min}(k) \le \sum_{i=k}^{n} V^j_i \le \Gamma^j_{\max}(k), \tag{2.22}$$

$$\Gamma_{\min}(k) \le \sum_{i=k}^{n} U_i \le \Gamma_{\max}(k), \tag{2.23}$$

$$V^j_i, U_i \ge 0.$$

In the next theorem we obtain a bound on the sojourn time of the n-th job in TSC(RO) system in terms of values $\Gamma^j_{\min}(k), \Gamma^j_{\max}(k), \Gamma_{\min}(k), \Gamma_{\max}(k)$.

Theorem 2.6 *Suppose the relations (2.22) and (2.23) hold. Then*

$$W_n \le \max_{n \ge k_J \ge \ldots \ge k_1 \ge 1} \sum_{j=1}^{J-1} \left(\Gamma^j_{\max}(k_j) - \Gamma^j_{\min}(k_{j+1} + 1) \right) + \Gamma^J_{\max}(k_J) - \Gamma_{\min}(k_1 + 1) \tag{2.24}$$

We now show how Theorem 2.6 implies our main result Theorem 2.2.

Proof of Theorem 2.2. The proof consists of two steps: the first step uses Theorem 2.6 to bound W_n with uncertainty sets (2.9),(2.10). The second step involves solving some associated maximization problem.

We set $\Gamma_{\min}(k) = \lambda^{-1}(n + 1 - k) - \Gamma_a \phi(n + 1 - k), \Gamma_{\max}(k) = \lambda^{-1}(n + 1 - k) + \Gamma_a \phi(n + 1 - k), \Gamma^j_{\min}(k) = \mu_j^{-1}(n + 1 - k) - \Gamma_{s,j} \phi(n + 1 - k), \Gamma^j_{\max}(k) = \mu_j^{-1}(n + 1 - k) +$

$\Gamma_{s,j}\phi(n+1-k)$, where ϕ is defined by (3.2). From Theorem 2.6 we obtain:

$$W_n \le \max_{n\ge k_J\ge\ldots\ge k_1\ge 1} \sum_{j=1}^{J-1} \left(\mu_j^{-1}(n+1-k_j) + \Gamma_{s,j}\phi(n+1-k_j)\right)$$
$$- \sum_{j=1}^{J-1} \left(\mu_j^{-1}(n+1-k_{j+1}-1) - \Gamma_{s,j}\phi(n+1-k_{j+1}-1)\right)$$
$$+ \left(\mu_J^{-1}(n+1-k_J) + \Gamma_{s,J}\phi(n+1-k_J)\right) \; - \; \left(\lambda^{-1}(n+1-k_1-1) - \Gamma_a\phi(n+1-k_1-1)\right)$$

Since $n \ge k_{j+1} \ge k_j \;\; \forall j$, we can replace μ_j^{-1} by $\mu_{\min}^{-1} = \max(\mu_1^{-1}, \mu_2^{-1} \ldots, \mu_J^{-1}) < \lambda^{-1}$ and preserve inequality. Similarly, we can replace $\Gamma_{s,1}, \Gamma_{s,2}, \ldots, \Gamma_{s,J}, \Gamma_a$ by Γ. We obtain:

$$\begin{aligned}
W_n &\le \max_{n\ge k_J\ge\ldots\ge k_1\ge 1} \sum_{j=1}^{J-1} \left[\mu_{\min}^{-1}(k_{j+1}+1-k_j) + \Gamma\big(\phi(n+1-k_j) + \phi(n-k_{j+1})\big)\right] \\
&\quad + \left(\mu_{\min}^{-1}(n+1-k_J) + \Gamma\phi(n+1-k_J)\right) - \left(\lambda^{-1}(n-k_1) - \Gamma\phi(n-k_1)\right) \\
&\le \max_{n\ge k_1\ge 1} \mu_{\min}^{-1}(n-k_1) + 2J\Gamma\phi(n+1-k_1) \\
&\quad + J\mu_{\min}^{-1} - \lambda^{-1}(n-k_1) \quad \text{where we used } k_1 \le k_2 \le \ldots \le k_J \text{ to combine } \Gamma \text{ terms} \\
&= \max_{n\ge k_1\ge 1} (n+1-k_1)(\mu_{\min}^{-1} - \lambda^{-1}) + 2J\Gamma\phi(n+1-k_1) + (J-1)\mu_{\min}^{-1} + \lambda^{-1} \\
&\le \max_{n\ge k_1\ge 1} (n+1-k_1)(\mu_{\min}^{-1} - \lambda^{-1}) + 2J\Gamma\phi(n+1-k_1) + J\lambda^{-1} \quad \text{since } \lambda^{-1} > \mu_{\min}^{-1}
\end{aligned}$$

We let $x = n+1-k_1$. Since $1 \le k_1 \le n$ we have that $1 \le x \le n$ and obtain:

$$\begin{aligned}
W_n &\le \max_{n\ge x\ge 1} x(\mu_{\min}^{-1} - \lambda^{-1}) + 2J\Gamma\phi(x) + J\lambda^{-1} \\
&\le \max_{x\ge 1} x(\mu_{\min}^{-1} - \lambda^{-1}) + 2J\Gamma\phi(x) + J\lambda^{-1} \qquad (2.25)
\end{aligned}$$

Putting $a = \lambda^{-1} - \mu_{\min}^{-1}, b = J\Gamma, c = J\lambda^{-1}$, and using the assumption (3.6), we have $b/a = \lambda J\Gamma/(1-\rho^*) \ge e^{2e}$, namely, the condition (A.1) is satisfied. Applying Proposition A.3 from Appendix we obtain

$$W_n \le \frac{7\lambda J^2\Gamma^2}{1-\rho^*} \ln\ln \frac{\lambda J\Gamma}{1-\rho^*} + J\lambda^{-1}.$$

This completes the proof of the theorem. □

2.4.2 Proof of Theorem 2.6

Job 1 enters the system first, followed by jobs $2, 3, \ldots, n$. Let U_i^j be the time between the arrival of job i and job $i-1$ into server j for $i = 2, \ldots, n$ and $j = 1, \ldots, J$. Specifically, $U_i^1 = U_i$, and we define $U_1^j = V_1^{j-1}$ for $j = 2, \ldots, J$. The following relations are well known in the queueing theory (Kleinrock 1975).

$$W_i^j = \max(W_{i-1}^j + V_{i-1}^j - U_i^j, 0) \qquad \forall\, i = 2, \ldots, n,\ j = 1, \ldots, J, \tag{2.26}$$

$$U_i^j = V_i^{j-1} + I_i^{j-1} \qquad \forall\, i = 2, \ldots, n,\ j = 2, \ldots, J, \tag{2.27}$$

$$W_i^j = \max\Big\{ \max_{1 \le k \le i-1} \sum_{l=k}^{i-1} (V_l^j - U_{l+1}^j), 0 \Big\} \qquad \forall\, i = 2, \ldots, n,\ j = 1, \ldots, J, \tag{2.28}$$

$$W_{i-1}^j = W_i^j - I_i^j - (V_{i-1}^j - U_i^j) \qquad \forall\, i = 2, \ldots, n,\ j = 1, \ldots, J. \tag{2.29}$$

We now prove some more detailed results regarding the dynamics of our queueing system.

Corollary 1. *The following relations hold for $k = 2, \ldots, n-1$:*

$$\sum_{i=k+1}^{n} U_i^2 = \sum_{i=k+1}^{n} (V_i^1 + I_i^1) = W_n^1 - W_k^1 + \sum_{i=k+1}^{n} U_i^1 + V_n^1 - V_k^1.$$

Proof. The first equality follows from (2.27). To prove the second equality we use (2.29) to obtain

$$\sum_{i=k+1}^{n} (V_i^1 + I_i^1) = \sum_{i=k+1}^{n} (W_i^1 - W_{i-1}^1 + U_i^1) + V_n^1 - V_k^1,$$

and the result follows. □

Lemma 2.7

$$W_n = \max_{n \ge k_J \ge \ldots \ge k_1 \ge 1} \sum_{i=k_1}^{k_2} V_i^1 + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J - \sum_{i=k_1+1}^{n} U_i^1. \tag{2.30}$$

Proof. We prove Lemma 2.7 by induction. We let $W_i^{j,S} = W_i^j + V_i^j$ denote the sojourn time of customer i in server j.

Case $J = 1$: We first define $\sum_{i=j+1}^{j} \equiv 0$ for all j. Using (2.28) and $V_i^j \geq 0$ we have for any $n = 2, \ldots, n$:

$$\begin{aligned} W_n^{1,S} &= \max\Big(\max_{n-1\geq k_1\geq 1} \sum_{i=k}^{n-1} (V_i^1 - U_{i+1}^1), 0\Big) + V_n^1 \\ &= \max\Big(\max_{n\geq k_1\geq 1} \sum_{i=k_1}^{n} V_i^1 - \sum_{i=k_1+1}^{n} U_i^1, V_n^1\Big) \\ &= \max_{n\geq k_1\geq 1}\Big(\sum_{i=k_1}^{n} V_i^1 - \sum_{i=k_1+1}^{n} U_i^1\Big) \quad \text{and this completes case } J = 1. \end{aligned}$$

Case $J > 1$: Note that $W_n = W_n^{1,S} + (W_n^{2,S} + \ldots + W_n^{J,S})$ and denotes the sojourn time of job n in J-server system. We suppose that the result holds for a $J-1$ tandem system and proceed by induction:

$$\begin{aligned} &\max_{n\geq k_J\geq\ldots\geq k_1\geq 1} \sum_{i=k_1}^{k_2} V_i^1 + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J - \sum_{i=k_1+1}^{n} U_i^1 \\ = &\max_{n\geq k_J\geq\ldots\geq k_1\geq 1}\Big(\sum_{i=k_1}^{k_2} V_i^1 - \sum_{i=k_1+1}^{k_2} U_i^1\Big) - \sum_{i=k_2+1}^{n} U_i^1 + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J \\ = &\max_{n\geq k_J\geq\ldots\geq k_2\geq 1}\Big[\max_{k_1:k_2\geq k_1\geq 1}\Big(\sum_{i=k_1}^{k_2} V_i^1 - \sum_{i=k_1+1}^{k_2} U_i^1\Big) - \sum_{i=k_2+1}^{n} U_i^1 + \sum_{i=k_2}^{k_3} V_i^2 + \\ &\qquad \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J\Big] \\ = &\max_{n\geq k_J\geq\ldots\geq k_2\geq 1} W_{k_2}^{1,S} - \sum_{i=k_2+1}^{n} U_i^1 + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J \\ &\qquad \text{the base case } J = 1 \text{ is used} \\ = &\max_{n\geq k_J\geq\ldots\geq k_2\geq 1}\Big(W_{k_2}^{1,S} - \sum_{i=k_2+1}^{n} U_i^1\Big) + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J \end{aligned}$$

$$= \max_{n\geq k_2\geq\ldots\geq k_J\geq 1}\Big(W_n^{1,S} - \sum_{i=k_2+1}^{n} U_i^2\Big) + \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J$$

we used Corollary 1 and $W_{k_2}^{1,S} = W_{k_2}^1 + V_{k_2}^1$

$$= W_n^{1,S} + \max_{n\geq k_J\geq\ldots\geq k_2\geq 1} \sum_{i=k_2}^{k_3} V_i^2 + \ldots + \sum_{i=k_{J-1}}^{k_J} V_i^{J-1} + \sum_{i=k_J}^{n} V_i^J - \sum_{i=k_2+1}^{n} U_i^2$$

$$= W_1^{1,S} + (W_n^{2,S} + \ldots + W_n^{J,S}) \text{ by inductive assumption on } J-1 \text{ server system}$$

and the proof follows from definition of sojourn time W_n. □

Proof of Theorem 2.6. The result follows immediately from Lemma 2.7. □

2.5 Multiclass single server queueing system: proofs of main results

2.5.1 Proof of Theorem 2.4

Lemma 2.8 *For every t satisfying*

$$t \geq \max_j(\lambda_j^{-1}e^e, \lambda_j^{-1} + 3\lambda_j^{-1}\lambda_{\max}^2\Gamma^2), \tag{2.31}$$

the following holds: $A_j(t) \leq t\lambda_j + 3\lambda_j^2\Gamma^2\phi(t\lambda_j)$.

Proof. Assume first $A_j(t) < e^e$. Then applying (2.11) corresponding to the case $A_j(t) < e^e$, we obtain $A_j(t)\lambda_j^{-1} - \Gamma_{a,j} \leq t$, namely $A_j(t) \leq \lambda_j t + \lambda_j\Gamma_{a,j} \leq \lambda_j t + \lambda_j\Gamma$. Since $\lambda_j\Gamma, \phi(t\lambda_j) \geq 1$ from (3.6) and (3.2), the desired result is obtained. For the rest of the proof assume $A_j(t) \geq e^e$. Applying (2.11), we obtain $A_j(t)\lambda_j^{-1} - \Gamma_{a,j}\sqrt{A_j(t)\ln\ln A_j(t)} \leq t$. Which gives

$$\frac{A_j(t) - t\lambda_j}{\sqrt{A_j(t)\ln\ln A_j(t)}} \leq \lambda_j\Gamma_{a,j} \leq \lambda_j\Gamma. \tag{2.32}$$

Define b_j by: $b_j = t\lambda_j + 3\lambda_j^2\Gamma^2\sqrt{t\lambda_j\ln\ln t\lambda_j}$. Observe that:

$$
\begin{aligned}
\frac{b_j - t\lambda_j}{\sqrt{b_j \ln\ln b_j}} &= \frac{3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j}}{\left((t\lambda_j + 3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j})\ln\ln(t\lambda_j + 3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j})\right)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j}}{\left((t\lambda_j + 3\lambda_j^2\Gamma^2\sqrt{t^2\lambda_j^2})\ln\ln(t\lambda_j + 3\lambda_j^2\Gamma^2\sqrt{t^2\lambda_j^2})\right)^{\frac{1}{2}}} \\
&\quad \text{since } t\lambda_j \geq \ln\ln t\lambda_j \text{ for } t\lambda_j \geq e^e \text{ from (2.31)} \\
&= \frac{3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j}}{\left((t\lambda_j)(1+3\lambda_j^2\Gamma^2)\ln\ln(t\lambda_j)(1+3\lambda_j^2\Gamma^2)\right)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda_j^2\Gamma^2\sqrt{t\lambda_j \ln\ln t\lambda_j}}{\left((t\lambda_j)(1+3\lambda_j^2\Gamma^2)\ln\ln(t\lambda_j)^2\right)^{\frac{1}{2}}} \quad \text{since } t\lambda_j > 1 + 3\lambda_j^2\Gamma^2 \text{ from (2.31)} \\
&\geq \frac{3\lambda_j^2\Gamma^2\sqrt{\ln\ln t\lambda_j}}{\sqrt{(4\lambda_j^2\Gamma^2)(2\ln\ln t\lambda_j)}} \quad \text{since } 2\ln\ln t\lambda_j > \ln\ln(t\lambda_j)^2 \text{ for } t\lambda_j \geq e^e \text{ and } \lambda_j\Gamma \geq 1 \\
&\geq \lambda_j\Gamma \quad \text{by simplifying above expression.}
\end{aligned}
$$

Since $\frac{x-t\lambda_j}{\sqrt{x\ln\ln x}}$ is an increasing function for $x \geq e^e$ and from (3.24), we have that $b_j \geq A_j(t)$ and the result is obtained. □

We now obtain an upper bound on the cumulative arrival processes $\bar{A}_j(t), 1 \leq j \leq J$.

Lemma 2.9 *For every t satisfying (2.31), the following holds*

$$\phi(\bar{A}_j(t)) \leq \left((2 + 6\lambda_{\max}^2\Gamma^2)\right)^{\frac{1}{2}}\phi(\bar{\lambda}_j t)$$

Proof. Consider first the case $\bar{A}_j(t) < e^e$. From (3.2), we have that $\phi(\bar{A}_j(t)) = 1$ and applying (2.31), the lemma follows. Now we consider the case $\bar{A}_j(t) \geq e^e$. Recall that $\bar{A}_j(t) = e_j^T[I - P^T]^{-1}A(t)$. Applying Lemma 2.8

$$\bar{A}_j(t) \leq e_j^T[I - P^T]^{-1}\lambda t + e_j^T[I - P^T]^{-1}\begin{bmatrix} 3\lambda_1^2\Gamma^2\phi(t\lambda_1) \\ \vdots \\ 3\lambda_J^2\Gamma^2\phi(t\lambda_J) \end{bmatrix}$$

$$\leq e_j^T[I - P^T]^{-1}\lambda t + 3\lambda_{\max}^2\Gamma^2 e_j^T[I - P^T]^{-1}\lambda t \qquad \text{applying (2.31) and } x \geq \phi(x) \text{ for } x \geq e^e$$
$$= \bar{\lambda}_j t(1 + 3\lambda_{\max}^2\Gamma^2), \qquad \text{applying the definition of } \bar{\lambda}_j.$$

Applying this bound we also obtain

$$\begin{aligned}
\ln\ln \bar{A}_j(t) &\leq \ln\ln(\bar{\lambda}_j t(1 + 3\lambda_{\max}^2\Gamma^2)) \\
&\leq \ln\ln(\bar{\lambda}_j t)^2 \qquad \text{using assumption (2.31)} \\
&= \ln\ln\bar{\lambda}_j t + \ln 2 \\
&\leq 2\ln\ln\bar{\lambda}_j t, \qquad \text{using } \bar{\lambda}_j t \geq \lambda_j t \geq e^e \text{ from (2.31).}
\end{aligned}$$

Combining the previous bounds with definition of $\phi(x)$, the lemma follows. □

Lemma 2.10 *For every t satisfying (2.31), we have: $\bar{m}^T\bar{A}(t) - t \leq (\rho - 1)t + 3\lambda_{\max}\Gamma^2\phi(\lambda_{\max}t)$.*

Proof. Applying definition of $\bar{A}_j(t)$, we have

$$\begin{aligned}
\bar{m}^T\bar{A}(t) - t &= \bar{m}^T[I - P^T]^{-1}A(t) - t \\
&\leq m^T[I - P^T]^{-1}\big(\lambda t + 3\lambda_{\max}\Gamma^2\phi(\lambda_{\max}t)\lambda\big) - t \qquad \text{from Lemma 2.8} \\
&= \sum_j m_j\bar{\lambda}_j t + 3\lambda_{\max}\Gamma^2\phi(\lambda_{\max}t)\sum_j m_j\bar{\lambda}_j - t \qquad \text{applying the definition of } \bar{\lambda}_j \\
&= (\rho - 1)t + 3\lambda_{\max}\Gamma^2\phi(\lambda_{\max}t)\rho
\end{aligned}$$

and the lemma follows from applying the condition $\rho < 1$ to the second term. □

We now obtain an upper bound in the duration of the busy period. Recall the identity (2.2). Since the busy period begins at time zero its duration is upper bounded by the first time t such that

$$\sum_{i=1}^{\bar{A}_1(t)} V_i^1 + \ldots + \sum_{i=1}^{\bar{A}_J(t)} V_i^J - t < 0. \tag{2.33}$$

Consider any t satisfying the lower bound (2.31). We have

$$\begin{aligned}
&\sum_{i=1}^{\bar{A}_1(t)} V_i^1 + \ldots + \sum_{i=1}^{\bar{A}_J(t)} V_i^J - t \\
&\qquad \leq \sum_{j=1}^{J} \mu_j^{-1} \bar{A}_j(t) + \sum_{j=1}^{J} \Gamma_{a,j} \phi(\bar{A}_j(t)) - t \quad \text{applying (2.11),(2.12)} \\
&\qquad \leq \bar{m}^T \bar{A}(t) - t + \sum_{j=1}^{J} \Gamma_{a,j} \big((2 + 6\lambda_{\max}^2 \Gamma^2)\big)^{\frac{1}{2}} \phi(\bar{\lambda}_j t) \quad \text{applying Lemma 2.9} \\
&\qquad \leq t(\rho - 1) + 3\lambda_{\max} \Gamma^2 \phi(\lambda_{\max} t) + \sum_{j=1}^{J} \Gamma (2 + 6\lambda_{\max}^2 \Gamma^2)^{\frac{1}{2}} \phi(\bar{\lambda}_j t) \quad \text{applying Lemma 2.10} \\
&\qquad \leq t(\rho - 1) + (4J + 3) \bar{\lambda}_{\max} \Gamma^2 \phi(\bar{\lambda}_{\max} t),
\end{aligned}$$

where we have used a crude estimate $2 + 6\lambda_{\max}^2 \Gamma^2 < 16 \lambda_{\max}^2 \Gamma^2$, justified by (3.6). We now apply Lemma A.2 with $x = \bar{\lambda}_{\max} t, a = \bar{\lambda}_{\max}^{-1}(1 - \rho), b = (4J + 3)\bar{\lambda}_{\max} \Gamma^2/2$ and $c = 0$. The condition (A.1) is implied by assumption (3.6), and the second condition of Lemma A.2 is satisfied since $c = 0$. We obtain that (2.33) holds for all t satisfying (2.31) and

$$\begin{aligned}
t &\geq \frac{18(4J + 3)^2 \bar{\lambda}_{\max}^2 \Gamma^4}{4 \bar{\lambda}_{\max} \bar{\lambda}_{\max}^{-2} (1 - \rho)^2} \ln \ln \frac{3(4J + 3) \bar{\lambda}_{\max} \Gamma^2}{2 \bar{\lambda}_{\max}^{-1} (1 - \rho)} \\
&\geq \frac{5(4J + 3)^2 \bar{\lambda}_{\max}^3 \Gamma^4}{(1 - \rho)^2} \ln \ln \frac{2(4J + 3) \bar{\lambda}_{\max}^2 \Gamma^2}{1 - \rho}.
\end{aligned}$$

Observe using (3.6) that the right-hand side of the last expression is larger than the right-hand side of (2.31). Combining two cases we obtain (2.18).

We now turn to (2.19). First suppose t does not satisfy (2.31). Denote the right-hand side of (2.31) by C. That is $t < C$. Observe that $W(t) \leq (C - t) + W(C) \leq C + W(C)$ as the workload at time C corresponds in addition to arrivals during $[t, C]$. So now we focus on the case when t satisfies (2.31). We use Proposition A.3 from Appendix and obtain

$$\sup_{C \leq t \leq B} W(t) \leq \frac{7(4J + 3)^2 \bar{\lambda}_{\max}^2 \Gamma^4}{4 \bar{\lambda}_{\max}^{-1} (1 - \rho)} \ln \ln \frac{(4J + 3) \bar{\lambda}_{\max} \Gamma^2}{2 \bar{\lambda}_{\max}^{-1} (1 - \rho)}$$

$$\leq \frac{2(4J+3)^2\bar{\lambda}_{\max}^3\Gamma^4}{1-\rho}\ln\ln\frac{(4J+3)\bar{\lambda}_{\max}^2\Gamma^2}{1-\rho}.$$

From (3.6), we have $\Gamma \geq \lambda_{\min}^{-1}$. We conclude that

$$\sup_{0\leq t\leq B} W(t) \leq \frac{2(4J+3)^2\bar{\lambda}_{\max}^3\Gamma^4}{1-\rho}\ln\ln\frac{(4J+3)\bar{\lambda}_{\max}^2\Gamma^2}{1-\rho} + \Gamma + 3\bar{\lambda}_{\max}^2\Gamma^3.$$

This completes the proof of the theorem.

2.5.2 Proof of Corollary 2.5

First we establish bound (2.20). Let $t = 0$ mark the beginning of a busy period with (random) length B_∞ in steady state. This means that there is an arrival into one of the classes j_0 at time 0. Consider a modified system where the first arrivals into classes $j \neq j_0, \lambda_j > 0$ after time 0 are artificially pushed down to exactly time 0. Namely, now at time zero there is an arrival into every class j with $\lambda_j > 0$. The subsequent arrivals into these classes are also pushed earlier by the same amount, thus creating an i.i.d. renewal process initiated at time 0. Let $\hat{B}$ be the busy period initiated in the modified system at time 0. It is easy to see that almost surely $\hat{B} \geq B_\infty$. However, now that we have arrivals in every class at time zero, applying Proposition 2.1 and our result for the robust optimization counterpart queueing system, namely applying part (2.18) of Theorem 2.4, we obtain the required bound by taking the expected values of both sides of (2.18). This establishes part (2.20).

In order to prove (2.21), we use a bound (2.7). Using our earlier argument for the proof of (2.20) but applying it to the second moment of $\hat{B}$ we obtain

$$\mathbb{E}[B_\infty^2] \leq \mathbb{E}[\hat{B}^2] \leq \mathbb{E}\Big[\frac{25(4J+3)^4\bar{\lambda}_{\max}^6\Gamma^8}{(1-\rho)^4}\Big(\ln\ln\frac{2(4J+3)\bar{\lambda}_{\max}^2\Gamma^2}{1-\rho}\Big)^2\Big].$$

On the other hand, we trivially have have $\mathbb{E}[B_\infty] \geq \min_{1\leq j\leq J} m_j = 1/\mu_{\max}$, since every busy period involves at least one service completion. The result then follows.

2.6 Conclusion

Using ideas from the robust optimization theory we have developed a new method for conducting performance analysis of queueing networks. The essence of our approach is replacing stochastic primitives of the underlying queueing system with deterministic quantities which satisfy the implications of some probability laws. These implications take the form of linear constraints and for the case of two queueing systems, namely Tandem Single Class queueing networks and Multiclass Single Server queueing system, we have managed to derive explicit upper bounds on some performance measures such as sojourn times and workloads. Then we showed that the bounds implied by the Law of the Iterated Logarithm are applicable for the underlying stochastic queueing system leading to explicit and non-asymptotic performance bounds on the same performance measures.

We have just scratched the surface of possibilities in this work and we certainly expect that our approach can be strengthened and extended in multiple directions, some of which we outline below. First we expect that our approach extends to even more general models, such as, for example multiclass queueing networks or more general processing networks (Harrison 2000). The performance bounds can be obtained perhaps again by introducing linear constraints implied by probability laws and using some sort of a Lyapunov function for obtaining bounds in the resulting robust optimization type queueing model. Another important direction is identifying new probability laws which lead to tighter constraints than the ones implied by the LIL. Ideally, one would like to be able to obtain bounds which faithfully represent the scaling behavior of the performance measures of interest in the heavy traffic regime as the (bottleneck) traffic intensity ρ converges to the unity. Further, it would be interesting to obtain performance bounds on the tail probability of the performance measure of interest, perhaps by constructing constraints implied by bounds on the tail probabilities of the underlying stochastic processes. For example, perhaps one can obtain large deviations type bounds by considering the linear constraints implied by the large deviations bounds on the underlying stochastic processes. Deeper connection between the results of this chapter and the results in the adversarial queueing theory and the related queueing literature is worth investigating as well.

Finally, we expect that the philosophy of replacing the *probability model* with *implications of the probability model* will prove useful in non-queueing contexts as well, whenever one has to

deal with the issues of stochastic analysis of complicated functionals of stochastic primitives.

Chapter 3

Robust Optimization Analysis of the $GI/GI/m$ Queue

3.1 Introduction

Parallel server queueing systems model a variety of important phenomena in the real world. Examples of applications include hospitals and call centers. As such, they have received significant academic interest starting with the foundational work of Erlang on the $M/M/m$ queueing model. The central questions of interest in this area are those precisely of performance analysis - computing probability distributions and bounds on key performance measures such as waiting times, queue lengths, and busy period lengths.

An interesting feature of parallel server systems is its ability to operate in a variety of regimes that balance between efficiency and quality of offered service. In their seminal paper, Halfin and Whitt (Halfin and Whitt 1981) formally introduced a new unconventional heavy traffic regime (Halfin-Whitt regime) for queueing models for $M/M/m$ and $G/M/m$ models. In particular, in the Halfin-Whitt regime, service rate remains constant, but the number of servers m and arrival rate λ increase to infinity simulataneously to make the traffic intensity ρ approach 1, see Section 3.4 for details. The special aspect of this regime is that the steady state probability of delay has a nontrivial limit if and only if it is in the Halfin-Whitt Regime. Additionally, it is known that in the Halfin-Whitt Regime the steady-state queue length and waiting time scale respectively

as $O(\sqrt{m})$ and $O(\frac{1}{\sqrt{m}})$. This is another attractive feature of the regime since it balances the system utilization and quality of service, and for this reason, the systems are also referred to as Quality- and Efficiency-Driven (QED). Erlang (Erlang 1948) was the first to consider the QED regime, but the work by Halfin and Whitt brought a great deal of renewed interest in the area. Additionally, queueing models in the QED regime have found applications including modeling large-scale call and customer contact centers in (Aksin, Armony, and Mehrotra 2007) and (Gans, Koole, and Mandelbaum 2003). Most of the aforementioned results assume exponential service times, which significantly simplifies the analysis as one does not need to keep track of residual service times.

In this chapter, we utilize a performance analysis approach proposed in Chapter 2 to conduct a performance analysis of $GI/GI/m$ queueing systems. This approach uses ideas from robust optimization in an attempt to build a tractable optimization framework for analyzing queueing systems with general service and interarrival time distributions. The main premise of our approach in the queueing context is that, rather than assuming probabilistic laws for the underlying stochastic primitives, such as, for example, i.i.d. interarrival and service times, we consider a deterministic queueing model and we will assume only the implications of these laws. In summary, the contributions of this chapter are as follows:

- We illustrate our performance analysis approach based on robust optimization to address the question of computing waiting times and queueing lengths for the $GI/GI/m$ queueing system. Using this approach we are able to obtain *explicit* bounds on waiting times and queueing lengths of the form

$$C(1-\rho)^{-1}\ln\ln((1-\rho)^{-1})$$

that qualitatively agree with stochastic approaches such as the bounds in (Kingman 1970) up to the the $\ln\ln((1-\rho)^{-1})$ correction factor. One can also use this optimization framework to more precisely numerically compute the waiting time and queueing length bounds for an arbitrary number of jobs that are scheduled to arrive to the system by solving an associated linear program. The advantage of our approach is that the bounds obtained for the robust (deterministic) problem directly translate to bounds on the steady state performance measures in the underlying stochastic system.

- We analyze the waiting time of the $GI/GI/m$ robust analogue in the Halfin-Whitt regime and compare our analysis with bounds computed from traditional stochastic analysis. In particular, we explicitly construct an upper and lower bound on the maximum waiting time of customers in the robust $GI/GI/m$ system. These results indicate that as more servers are added to the system, customers in the robust $GI/GI/m$ system experience a decline (upper bound result) in the waiting time that is similar to the steady state waiting time in the stochastic $GI/GI/m$ system. However, as more and more servers are added to the system, we show that the waiting time in the robust system can remain strictly above zero (lower bound result), while the stochastic steady state waiting time is known to be driven to zero.

The rest of this chapter is organized as follows: Section 3.2 describes the stochastic $GI/GI/m$ model and its robust counterpart. In Section 3.3, we state our main results, namely the performance bounds in robust optimization type queueing systems and their implications for stochastic queueing systems. Section 3.4 presents the analysis of the waiting time in the Halfin-Whitt regime. Section 3.5 contains the proofs of the main results. Several technical results necessary for proofs of main theorems are delayed until the Appendix section.

3.2 $GI/GI/m$ model description

3.2.1 Stochastic model

The model is a system of m parallel identical single servers $\sigma_1, \ldots, \sigma_m$ processing a single stream of jobs arriving from the outside. We assume a total of $N+1$ jobs arrive from outside according to an i.i.d. renewal process. Let $U_N, U_{N-1}, \ldots, U_2, U_1$ denote i.i.d. interarrival times with a common distribution function $F_a(t) = \mathbb{P}(U \leq t)$, where U_k denotes the time between arrival of job $k-1$ and k. For clarity of exposition, we label the jobs in reverse order so that jobs arrive in the order $N, N-1, N-2, \ldots, 2, 1, 0$, and job N is the first job to arrive. The arrival rate is defined to be $\lambda \triangleq 1/\mathbb{E}[U_1]$ and the variance of U_k is denoted by σ_a^2.

The jobs arriving externally join the main buffer and are served using the First-In-First-Out (FIFO) scheduling policy by the first available server. Upon completion of service in any server

$1, \ldots, m$, jobs exit the system. Let $V_N, V_{N-1}, \ldots, V_2, V_1, V_0$ denote i.i.d. service times of jobs $N, N-1, \ldots, 1, 0$ with a common distribution function $F_s(t) = \mathbb{P}(V \leq t)$. Additionally, we assume that the sequence $(V_k, 0 \leq k \leq N)$ is independent from all other random variables in the network. The servers have a common rate of service defined to be $\mu \triangleq 1/\mathbb{E}[V]$ and the variance of V_k is denoted by σ_s^2 for $k = N, \ldots, 0$. We denote by $\rho = \frac{\lambda}{m\mu}$ the traffic intensity.

We denote by W_k the waiting time experienced by job k not including its service time V_k. Denote by $Q(t)$ the queue length in the system (number of jobs waiting to be served in the main buffer) at time t. Additionally, we denote by Q_k the queue length in the system upon the arrival of job k (not including k). We assume that the first job N arrives to the system at time $t = 0$, and that initially the system is empty: $Q_N = Q(0) = 0$.

The model just described will be denoted by GGm(St), see (Pollaczek 1961), (de Smit 1983b; de Smit 1983a), and (Bertsimas 1990). It is known (Kiefer and Wolfowitz 1955; Loynes 1962; Whitt 1972) that as long as $\rho < 1$, and some additional mild conditions hold, such as $\mathbb{P}(U_n - V_n > 0) > 0$, GGm(St) is stable and the transient performance measures converge to the (unique) steady-state performance measures in distribution. However, computing these performance measures is a different matter. We denote by W_∞, Q_∞ the steady state versions of the random variables W_k, Q_k. Thus provided that $\rho < 1$ and some additional technical assumptions hold, we have

$$\lim_{N\to\infty} \mathbb{E}[W_0] = \mathbb{E}[W_\infty] \qquad \lim_{N\to\infty} \mathbb{E}[Q_0] = \mathbb{E}[Q_\infty]. \tag{3.1}$$

We will assume that $\rho < 1$ holds without explicitly stating it. Rather than describing the assumptions required to make (3.1) true, we will simply assume when stating our results that (3.1) holds as well.

3.2.2 Robust model

We now describe the deterministic robust optimization type counterpart of the stochastic queueing model GGm(St), which we denote by GGm(RO).

The description of the network topology is the same as for GGm(St). However, it is not

assumed that U_k, V_k and, as a result $Q(t), W_k$ are random variables. Rather we assume that these quantities are *arbitrary* subject to certain linear constraints detailed below. Additionally, we assume that the system starts empty $Q(0) = 0$, job N arrives at time $t = 0$, and only $N + 1$ jobs go through the system. For clarity of exposition, we label the jobs in reverse order so that jobs arrive in the order $N, N-1, N-2, \ldots, 2, 1, 0$ so that job $N+1-j$ is the j-th job to arrive to the system.

Specifically, consider a sequence of non-negative deterministic interarrival and service times $(U_k, 1 \leq k \leq n), (V_k, 1 \leq k \leq N)$. In particular, U_k represents the interarrival time between job $k-1$ and k for $k = 1, \ldots, N$, and V_k represents the service time of job k for $k = 1, \ldots, N$. Let

$$\phi(x) = \begin{cases} \sqrt{x \ln \ln x}, & x \geq e^e; \\ 1, & x < e^e. \end{cases} \tag{3.2}$$

It is assumed that values $\lambda, \mu, \Gamma_a, \Gamma_s \geq 0$ exist such that

$$\Big| \sum_{1 \leq i \leq k} U_k - \lambda^{-1} k \Big| \leq \Gamma_a \phi(k), \quad k = 1, 2, \ldots, \tag{3.3}$$

$$\Big| \sum_{1 \leq i \leq k} V_i - \mu^{-1} k \Big| \leq \Gamma_s \phi(k), \quad k = 1, 2, \ldots, \tag{3.4}$$

$$V_i \leq B, \quad i = 1, 2, \ldots, N. \tag{3.5}$$

These assumptions and constraints are similar to the ones used to describe TSC and MCSS systems (2.9)-(2.12) in Chapter 2. (3.5) means that we assume that the service times are bounded with probability 1. We let $\Gamma = \max(\Gamma_a, \Gamma_s)$.

For technical reasons, we also assume that Γ_a and λ in GGm(RO) constraints satisfy

$$\lambda \Gamma_a \geq 2e^{2e}, \quad \Gamma_a \rho \geq 1, \quad \text{and} \quad \lambda \geq 1. \tag{3.6}$$

While we assume $N + 1$ jobs going through the system, we will be able to apply our results in the stochastic setting where infinite number of jobs pass through the system. Borrowing from the robust optimization literature terminology (Bertsimas and Sim 2004), the parameters Γ_a, Γ_s are called *budgets of uncertainty*. Note, that the values U_k and V_k, for $1 \leq k \leq N$, uniquely

define the corresponding performance measures $Q(t), W_k,\ k = 0, \ldots, N$. There is no notion of steady state quantities Q_∞, W_∞ for the model TSC(RO). The motivation for constraints (3.3) and (3.4) comes from the Law of the Iterated Logarithm. We refer the reader to section 2.2.4 for a discussion on the Law of the Iterated Logarithm.

3.3 Main results

In this section we state our main results on the bounds for waiting time and queue lengths for the robust optimization system GGm(RO), and the implications of our results for its stochastic counterpart GGm(St).

Theorem 3.1 *Given a GGm(RO) queueing system with constraints (3.3)-(3.6), let W_0 be the waiting time of job 0. Then*

$$W_0 \leq \frac{2\lambda(\frac{\Gamma_s}{m} + \Gamma_a)^2}{1-\rho} \ln\ln \frac{\lambda(\frac{\Gamma_s}{m} + \Gamma_a)}{2(1-\rho)} + B. \tag{3.7}$$

Observe that the bound on the waiting time is explicit. It is expressed directly in terms of the primitives of the queueing system such as arrival and service rates. Observe also that the upper bound is independent from N - the number of jobs that passed through the system. One can think of this bound as a "steady-state" bound (for large enough N) on the waiting time in the GGm(RO) system. Additionally, the constants Γ_s, Γ_a are related to the standard deviations of service and interarrival times viz a vi the LIL (2.14). It is known from Kingman's bound (Kingman 1970) that in the stochastic $GI/GI/m$ queueing system, the expected waiting time in steady state is at most $\lambda(\frac{\sigma_s^2}{m} + \sigma_a^2 + (m^{-1} - m^{-2})\mu^{-2})/2(1-\rho)$ in heavy traffic. Namely, the expected waiting time depends linearly on the variances of interarrival and service times. Our bound (3.7) is thus consistent with this type of dependence. On the other hand, the bound above does not have the familiar $O((1-\rho)^{-1})$ scaling, which is known to be correct from (Kingman 1970). However, the correction factor is a very slowly growing function $\ln\ln$. Additionally, while the bound above also has a constant B, this constant is much less significant for ρ close to 1. The upshot is that we can use this bound to obtain a bound on W_0 and W_∞ in

the underlying stochastic system. This is what we do next.

Recall $\Gamma_{a,\text{LIL}}$ and $\Gamma_{s,\text{LIL}}$ from Proposition 2.1, where $\Gamma_{\cdot,\text{LIL}}$ is defined for the corresponding sequence in (2.14). Define $\Gamma_a = \max(\sqrt{2}\sigma_a\Gamma_{a,\text{LIL}}, 2e^{2e}\lambda^{-1}, \rho^{-1})$ and $\Gamma_s = \max(\sqrt{2}\sigma_s\Gamma_{s,\text{LIL}}, 2e^{2e}\lambda^{-1})$ and observe that (3.6) is satisfied.

Corollary 3.2 *For every $N \geq 1$ the sojourn time of the N-th job (W_0) in the GGm(St) queueing network satisfies*

$$\mathbb{E}[W_0] \leq \mathbb{E}\Big[\frac{2\lambda(\frac{\Gamma_s}{m} + \Gamma_a)^2}{1-\rho} \ln\ln \frac{\lambda(\frac{\Gamma_s}{m} + \Gamma_a)}{2(1-\rho)}\Big] + B. \tag{3.8}$$

If in addition Assumption (3.1) holds, then

$$\mathbb{E}[W_\infty] \leq \mathbb{E}\Big[\frac{2\lambda(\frac{\Gamma_s}{m} + \Gamma_a)^2}{1-\rho} \ln\ln \frac{\lambda(\frac{\Gamma_s}{m} + \Gamma_a)}{2(1-\rho)}\Big] + B. \tag{3.9}$$

Proof. We first assume Theorem 3.1 is established. Note, in the context of the stochastic system, W_0, Γ_a, and Γ_s in Theorem 3.1 are random variables. Applying Proposition 2.1 in section 2.2.4 we have that (3.7) holds with probability one for the underlying stochastic network. The bound (3.8) now follows from taking expectations of both sides of (3.7). The bound (3.9) follows from applying (3.1) to (3.8). □

Theorem 3.3 *Given a GGm(RO) queueing system with constraints (3.3)-(3.6), let Q_0 be the number of people in the queue when job 0 arrives into the system. Then*

$$Q_0 \leq \frac{2\rho\Psi^2}{1-\rho} \ln\ln \frac{\rho\Psi}{2(1-\rho)} + \mu(m-1)B \tag{3.10}$$

where $\Psi = 3\lambda^2\Gamma_a^2 + 2\Gamma_s\mu\big((2 + 6\lambda^2\Gamma_a^2)\big)^{\frac{1}{2}}$.

Corollary 3.4 *For every $N \geq 1$, the queueing length (Q_0) at the time of arrival of job 0 into*

the GGm(St) queueing network satisfies

$$\mathbb{E}[Q_0] \leq \mathbb{E}\Big[\frac{2\rho\Psi^2}{1-\rho} \ln \ln \frac{\rho\Psi}{2(1-\rho)}\Big] + \mu(m-1)B \tag{3.11}$$

where $\Psi = 3\lambda^2\Gamma_a^2 + 2\Gamma_s\mu\big((2+6\lambda^2\Gamma_a^2)\big)^{\frac{1}{2}}$. *If in addition Assumption (3.1) holds, then*

$$\mathbb{E}[Q_\infty] \leq \mathbb{E}\Big[\frac{2\rho\Psi^2}{1-\rho} \ln \ln \frac{\rho\Psi}{2(1-\rho)}\Big] + \mu(m-1)B \tag{3.12}$$

Proof. We first assume Theorem 3.3 is established. Note, in the context of the stochastic system, Q_0, Γ_a, and Γ_s in Theorem 3.3 are random variables. Applying Proposition 2.1 we have that (3.10) holds with probability one for the underlying stochastic network. The bound (3.11) now follows from taking expectations of both sides of (3.10). The bound (3.12) follows from applying (3.1) to (3.11). □

3.4 Performance bounds in the Halfin-Whitt Regime

In this section, we will formally introduce the Halfin-Whitt Regime and present results which compare the performance of our robust system GGm(Rob) in the Halfin-Whitt Regime to classical analysis.

Formally, the Halfin-Whitt Regime is defined by setting $\lambda = m\mu - \beta\mu\sqrt{m}$. As a result, $\rho = 1 - \frac{\beta}{\sqrt{m}}$.

We define the uncertainty sets of the robust queueing model in the Halfin-Whitt Regime GGm(RO-HW) by:

$$\Big|\sum_{1\leq i\leq k} U_k - \lambda^{-1}k\Big| \leq \frac{\Gamma_a}{m}\phi(k), \quad k = 1,2,\ldots, \tag{3.13}$$

$$\Big|\sum_{1\leq i\leq k} V_i - \mu^{-1}k\Big| \leq \Gamma_s\phi(k), \quad k = 1,2,\ldots, \tag{3.14}$$

$$V_i \leq B \quad i = 1,2,\ldots,N. \tag{3.15}$$

Note the only difference between uncertainty sets for GGm(RO-HW) and GGm(RO) is in the constraints on the arrival process where Γ_a in (3.3) is replaced by $\frac{\Gamma_a}{m}$ in (3.13). In fact, because the arrival *rate* $\lambda \approx O(m)$, the standard deviation of the interarrival *times* (absorbed by the Γ_a parameter) must accordingly scale by $(\frac{1}{m})$. For technical purposes, we will additionally assume that

$$B \geq 2\mu^{-1} \quad \text{and} \quad \Gamma_s \geq \mu^{-1}. \tag{3.16}$$

We now state and prove the following theorem which characterizes the behavior of the waiting time of the $N + 1^{\text{st}}$ job, W_0, in the GGm(RO-HW) system characterized by (3.13)-(3.16).

Theorem 3.5 *Given a GGm(RO-HW) queueing system defined by (3.13)-(3.16), let W_0 denote the maximum feasible waiting time of job 0. Then*

$$0.45\mu^{-1} < W_0 \leq \frac{C \ln \ln (C'\sqrt{m})}{\sqrt{m}} + B \tag{3.17}$$

where $C = 2\mu(\Gamma_s + \Gamma_a)^2/\beta$ and $C' = \mu(\Gamma_s + \Gamma_a)/2\beta$.

Before delving into the proof of Theorem 3.5, we would first like to discuss its implications.

First, observe that the upper bound in (3.17) implies a similar result for the expected waiting time of the $N+1^{\text{st}}$ job ($\mathbb{E}(W_0)$) for the underlying stochastic system. The proof follows similarly to the proof of Corollary 3.2. Additionally, similar to the behavior of the steady state waiting time in Halfin-Whitt Regime, increasing the number of servers causes the main term in the upper bound (3.7) to decay almost as $O(\frac{1}{\sqrt{m}})$.

The key difference between the behavior of the waiting time in GGm(RO-HW) as compared to the classical approaches is that in our system, while W_0 decays as the number of servers increases (in particular the term corresponding to $\frac{1}{1-\rho}$ in (3.7)), a non-zero waiting time still remains and can be achieved for arbitrary large m. The intuitive reason behind this difference is that our approach is inherently transient analysis based, as opposed to the standard steady state analysis used in classical queueing work. In fact, we show that for arbitrary high m, it is possible to achieve a positive waiting time for the $N + 1^{\text{st}}$ job, W_0. In particular, using a sequence of

interarrival and service times similar to the one used in our proof (3.18), a similar phenomenon would result in a stochastic system for transient jobs. However, as this phenomenon is transient, it disappears in steady state and $\mathbb{E}[W_0] \to 0$ in classical analysis.

We now prove Theorem 3.5.

Proof. The proof consists of two parts: $W_0 \leq C \ln\ln(C'\sqrt{m})/\sqrt{m} + B$ and $W_0 > 0.45\mu^{-1}$.

Case $W_0 \leq C \ln\ln(C'\sqrt{m})/\sqrt{m} + B$.
From Theorem 3.1 and (3.13-3.15), we obtain:

$$\begin{aligned} W_0 &\leq \frac{2\lambda(\frac{\Gamma_s}{m}+\frac{\Gamma_a}{m})^2}{1-\rho} \ln\ln \frac{\lambda(\frac{\Gamma_s}{m}+\frac{\Gamma_a}{m})}{2(1-\rho)} + B \\ &= \frac{\mu(2m-2\beta\sqrt{m})(\frac{\Gamma_s}{m}+\frac{\Gamma_a}{m})^2}{\frac{\beta}{\sqrt{m}}} \ln\ln\left(\frac{\mu(m-\beta\sqrt{m})(\frac{\Gamma_s}{m}+\frac{\Gamma_a}{m})}{2\frac{\beta}{\sqrt{m}}}\right) + B \\ &\leq \frac{2\mu(\Gamma_s+\Gamma_a)^2}{\beta\sqrt{m}} \ln\ln\left(\frac{\mu(\Gamma_s+\Gamma_a)\sqrt{m}}{2\beta}\right) + B \\ &= \frac{C\ln\ln(C'\sqrt{m})}{\sqrt{m}} + B \end{aligned}$$

where $C = 2\mu(\Gamma_s+\Gamma_a)^2/\beta$ and $C' = \mu(\Gamma_s+\Gamma_a)/2\beta$.

Case $W_0 > 0.45\mu^{-1}$. We will do this by explicitly constructing a sequence of service and interarrival times that satisfies (3.13), (3.14), and (3.15) and achieves $W_0 > 0.45\mu^{-1}$.

We consider a system with $N+1$ jobs where $N = 2m+0.49m$. The jobs arrive to the system in the order $N, N-1, \ldots, N-i+1, \ldots, 1, 0$ where $(N-i+1)$ is the i-th job to arrive to the system. For technical purposes, we assume that m is large enough to satisfy $m-\beta\sqrt{m} \geq 0.99m$, $m \geq 10000$, and $m/100$ is an integer. Since we are interested in limiting behavior as $m \to \infty$, these are appropriate assumptions.

Consider the following sequence of interarrival and service times:

$$U_i = \frac{1}{\lambda} = \frac{1}{\mu(m-\beta\sqrt{m})} \leq \frac{\mu^{-1}}{0.99m} \leq \frac{\mu^{-1}}{0.99 \cdot 10000} \leq 0.01\mu^{-1}$$

$$V_i = \begin{cases} \mu^{-1}, & i \leq 0.49m; \\ 2\mu^{-1}, & 0.49m < i \leq N \text{ and } i \text{ even}; \\ 0, & 0.49m < i \leq N \text{ and } i \text{ odd}; \end{cases} \tag{3.18}$$

Let t_{N-2m+1} denote the arrival of job $N-2m+1$. Observe, that for jobs i, for $0.49m < i \leq N$, the service times alternate between $2\mu^{-1}$ and 0 units. Since the system is initially empty, without loss of generality suppose that jobs enter service in the first available server, and in the smallest indexed server if more than one server is available. It follows from (3.18) that the workloads remaining in σ_i at time t_{N-2m+1} is $W_{\sigma_i}(t_{N-2m+1}) = \max\{0, \frac{2}{0.99m\mu}(i-m)+2\mu^{-1}\}$ for $i = 1, \ldots, m$, and

$$W_{\sigma_i}(t_{N-2m+1}) \geq W_{\sigma_{0.49m}}(t_{N-2m+1}) = \frac{2}{0.99m\mu}(0.49m - m) + 2\mu^{-1} > 0.96\mu^{-1}$$

for servers $i = 0.49m, 0.49m+1, \ldots, m$.

Now consider the arrival of $0.49m$ more jobs and let $t_{N-(2m+0.49m)+1}$ denote the time of the arrival of job $(N - (2m + 0.49m) + 1)$. By (3.18), these additional $0.49m$ jobs have initial service time of length μ^{-1}. First observe that $t_{N-(2m+0.49m)+1} - t_{N-2m+1} < .5\mu^{-1}$ since it takes $\frac{0.49m}{\lambda} = \frac{0.49m}{\mu(m-\beta\sqrt{m})} \leq \frac{0.49m}{0.99m\mu} < 0.5\mu^{-1}$ units for $0.49m$ jobs to arrive. Hence, the work remaining in servers σ_i for $i = 0.49m, 0.49m+1, \ldots, m$ at time $t_{N-(2m+0.49m)+1}$ is $W_{\sigma_i}(t_{N-(2m+0.49m)+1}) > 0.96\mu^{-1} - 0.5\mu^{-1} = 0.46\mu^{-1}$. Additionally, there is currently $0.49m$ other jobs that arrived in interval $(t_{N-2m+1}, t_{N-(2m+0.49m)+1}]$ with remaining service at least $\mu^{-1} - 0.5\mu^{-1} = 0.5\mu^{-1}$ units (since they all arrived at most $(0.5\mu^{-1})$ units ago). Thus, it will be at least $\min(0.5\mu^{-1}, 0.46\mu^{-1}) = 0.46\mu^{-1}$ more units of time until any server becomes available. Hence, the waiting time of the next job, job $(N - (2m + 0.49m + 1) + 1)$ is at least

$$W_0 = W_{N-(2m+0.49m+1)+1} \geq 0.46\mu^{-1} - \frac{1}{\lambda} \geq 0.45\mu^{-1} \text{ by (3.18).}$$

□

It is worthwhile to highlight the underlying dynamics of the sequence of service times in (3.18) that results in the lower bound on the waiting time for the last job. Intuitively, the sequence of service times used in the proof represents sequences of jobs with service times from two types of distributions - having the same average service time μ^{-1}, but different variance. First, a large number of jobs with high variance of service time (alternating sequence of 0's and $2\mu^{-1}$'s) arrive. This results in some jobs being processed quickly, while others take longer than expected and have tails (in our case $\frac{1}{2}$ jobs are length 0 and $\frac{1}{2}$ are length $2\mu^{-1}$). Immediately following these jobs, for a short period of time all of the jobs that arrive have low variance of service time (i.e. all jobs have μ^{-1} service time). As a result, this sequence of jobs with low variance combined with the extra long jobs left over from the initial set of arrivals end up clogging up the system for a short period of time. As a result, some jobs that arrive immediately after have to wait. However, this phenomenon washes away in steady state as the extra long jobs are on average well balanced with the extra short jobs in a way that the average waiting time in steady state approaches zero.

3.5 Proofs of Main Results

3.5.1 Waiting Time: Proof of Theorem 3.1

We are interested in computing the maximum possible waiting time of job 0. We remind the reader that our notation is reverse with respect to sequence of jobs where job 0 is the last job out of a total of $N+1$ jobs to arrive into the system. The general approach of the proof will be to compute a bound on the average workload remaining in the system that is $\left(\frac{\text{total workload}}{m}\right)$ at the time of job 0 arrival. This serves as a natural upper bound on $W_0 = \min_j(\text{workload remaining in } s_j)$ since jobs are served according to FIFO and enter service as soon as a server becomes available.

For the rest of the proof, we will assume that when job 0 arrives, all servers are busy, since otherwise job 0 immediately enters service and $W_0 = 0$.

Notation:

- Let job n denote the earliest job (highest index) to arrive that initiated a period of all servers being continuously busy up to and including the arrival of job 0.
- Let t_0 and t_n denote the time of arrival of jobs 0 and n, respectively, into the system.
- Let $t = (U_n + U_{n-1} + \ldots + U_1) = t_0 - t_n$ denote the time between arrival of job 0 and job n.
- Let $\{k_2, k_3, \ldots, k_m\}$ be the set of jobs in service at time t_n. Naturally, $n < k_i \leq N$ for $i \in 2, \ldots, m$.
- Let $V_{k_i}^R$ for $i = 2, \ldots, m$ denote the remaining service of job k_i at time t_n.

We state the following obvious Claim without proof:

Claim 3.6 *There are exactly $(m-1)$ jobs in service and no jobs in the queue at the time of arrival of job n.*

We will now present a lemma that upper bounds the W_0 in terms of jobs that arrive before it:

Lemma 3.7

$$W_0 \leq \frac{\sum_{i=1}^{n} V_i + (V_{k_2}^R + \ldots + V_{k_m}^R)}{m} - (U_1 + \ldots + U_n)$$

Proof. Let W denote the sum of the total workload remaining in the system at time t_n and all future work that arrives to the system after job n (not including job 0).

By Claim 3.6, the workload in the system at time t_n is only due to job n and remaining service times of jobs in all the other servers which is $V_n + (V_{k_2}^R + V_{k_3}^R + \ldots + V_{k_m}^R)$. All future work that arrives to the system after job n (not including job 0) is $(V_1 + V_2 + \ldots + V_{n-1})$. From this we see that

$$W = (V_1 + V_2 + \ldots + V_n) + (V_{k_2}^R + V_{k_3}^R + \ldots + V_{k_m}^R) \tag{3.19}$$

The additional time it takes until at least one server becomes free is less than the average additional time it takes for all of the servers to process the above work. Since all m servers are continuously busy beginning with the arrival of job n, the latter quantity is $\frac{W}{m}$.

Since job 0 arrives t units after job n, we obtain

$$W_0 \leq \frac{W}{m} - t,$$

and the statement of the Lemma follows from (3.19) and the definition of t. □

From Lemma 3.7, we obtain:

$$\begin{aligned} W_0 &\leq \frac{\sum_{i=1}^{n} V_i + (V_{k_2}^R + \ldots + V_{k_m}^R)}{m} - (U_1 + \ldots + U_n) \\ &\leq \frac{n\mu^{-1} + \Gamma_s\phi(n) + (m-1)B}{m} - (n\lambda^{-1} - \Gamma_a\phi(n)) \text{ from uncertainty sets (3.3), (3.4), (3.5)} \\ &\leq -\lambda^{-1}(1-\rho)n + \phi(n)\left(\frac{\Gamma_s}{m} + \Gamma_a\right) + B \quad \text{by definition of } \rho = \frac{\lambda}{m\mu}. \end{aligned}$$

Observe that the last expression is of the form $U(x) = -ax + 2b\phi(x) + c$ where $a = \lambda^{-1}(1-\rho)$, $b = \frac{1}{2}\left(\frac{\Gamma_s}{m} + \Gamma_a\right)$, $c = B$. Also, $\frac{b}{a} \geq \frac{\Gamma_a}{2\lambda^{-1}} \geq e^{2e}$ by (3.6).

We invoke Proposition A.3 to obtain the final result:

$$W_0 \leq \max_{x \geq 0} U(x) \leq \frac{7\lambda(\frac{\Gamma_s}{m} + \Gamma_a)^2}{4(1-\rho)} \ln\ln \frac{\lambda(\frac{\Gamma_s}{m} + \Gamma_a)}{2(1-\rho)} + B.$$

3.5.2 Queue Length: Proof of Theorem 3.3

The question we are interested in is computing the maximum number of jobs (Q_0) waiting in the queue (not including job 0) when the last job - 0 arrives to the system subject to our uncertainty sets (3.3), (3.4), (3.5).

Without loss of generality, we assume that upon arrival of job 0 all of the servers are busy. Otherwise, if one of the servers is idle, this implies that the queue must be empty and hence $Q_0 = 0$. We use the same notation as in the previous section (3.5.1) and also introduce

two additional notations:

- Let $S = \{i_1, i_2, \ldots, i_k\}$, $1 \leq i_j \leq n$ for $j = 1, \ldots, k$ denote the set of jobs (not including 0) in the queue at time t_0.
- Let $k = \arg\max S$ denote the highest index (earliest) of the jobs in the queue at time t_0.

Observe that $Q_0 = k$.

The following lemma bounds Q_0 in terms of n and t.

Lemma 3.8

$$Q_0 \leq n + 2\Gamma_s\phi(n)\mu + \mu(m-1)B - tm\mu$$

Proof. Observe that when job 0 arrives into the system, two things must be true:

- $\{k, k-1, k-2, \ldots, 3, 2, 1\}$ are still in the queue
- servers have processed tm units of work in the last t units (since they have all been operating continuously for the last t units).

Also observe that:

$$tm \leq (V_{k_2}^R + V_{k_3}^R + \ldots + V_{k_m}^R) + (V_n + V_{n-1} + \ldots + V_{k+2} + V_{k+1}) \tag{3.20}$$

From (3.20), we obtain

$$\begin{aligned} tm &\leq (V_{k_2} + V_{k_3} + \ldots + V_{k_m}) + (V_n + V_{n-1} + \ldots + V_{k+2} + V_{k+1}) \\ &\leq (m-1)B + \mu^{-1}(n-k) + \Gamma_s(\phi(n) + \phi(k)) \\ &\leq (m-1)B + \mu^{-1}(n-k) + 2\Gamma_s\phi(n) \\ k &\leq n + 2\Gamma_s\phi(n)\mu + \mu(m-1)B - tm\mu \end{aligned}$$

Since $Q_0 = k$, the proof is complete. □

To complete the proof of the theorem, we consider two cases: $\lambda t < 1 + 3\lambda^2\Gamma_a^2$ and $\lambda t \geq 1 + 3\lambda^2\Gamma_a^2$:

Lemma 3.9 *For every t satisfying $\lambda t < 1 + 3\lambda^2\Gamma_a^2$, the following holds:*

$$Q_0 \leq \frac{2\rho\Psi^2}{1-\rho} \ln\ln \frac{\rho\Psi}{2(1-\rho)} + \mu(m-1)B \tag{3.21}$$

where $\Psi = 3\lambda^2\Gamma_a^2 + 2\Gamma_s\mu\big((2 + 6\lambda^2\Gamma_a^2)\big)^{\frac{1}{2}}$.

Proof. In this proof, we will show that the condition $\lambda t < 1 + 3\lambda^2\Gamma_a^2$ implies that the bound in Theorem 3.3 is an upper bound on n, and hence also an upper bound on Q_0.

Assume first $n < e^e$. Then applying (3.3) corresponding to the case $n < e^e$, we obtain $n\lambda^{-1} - \Gamma_a \leq t$, namely

$$\begin{aligned} n &\leq \lambda t + \lambda\Gamma_a \\ &\leq (1 + 3\lambda^2\Gamma_a^2) + \lambda\Gamma_a \\ &\leq (\lambda^2\Gamma_a^2)(0.002 + 3 + 1) \quad \text{since } \lambda\Gamma \geq 2e^e \text{ by (3.6)} \\ &\leq (\lambda\Gamma_a)^3 \quad \text{since } \lambda\Gamma \geq 2e^e \text{ by (3.6)} \end{aligned}$$

which is less than the bound (3.21).

For the rest of the proof assume $n \geq e^e$. Applying (3.3), we obtain $n\lambda^{-1} - \Gamma_a\sqrt{n \ln\ln n} \leq t$. This in addition to condition $(t\lambda < 1 + 3\lambda^2\Gamma_a^2)$ give:

$$\frac{n - t\lambda}{\sqrt{n \ln\ln n}} \leq \lambda\Gamma_a \;\Rightarrow\; \frac{n - (1 + 3\lambda^2\Gamma_a^2)}{\sqrt{n \ln\ln n}} \leq \lambda\Gamma_a. \tag{3.22}$$

Let $\Delta = (1 + 3\lambda^2\Gamma_a^2)$ and $b = \Delta + 3\lambda^2\Gamma_a^2\sqrt{\Delta \ln\ln\Delta}$.

Observe that:

$$\begin{aligned}
\frac{b-\Delta}{\sqrt{b\ln\ln b}} &= \frac{3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta}}{\Big((\Delta+3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta})\ln\ln(\Delta+3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta})\Big)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta}}{\Big((\Delta+3\lambda^2\Gamma_a^2\Delta)\ln\ln(\Delta+3\lambda^2\Gamma_a^2\Delta)\Big)^{\frac{1}{2}}} \quad \text{since } \Delta \geq \ln\ln\Delta \text{ for } \Delta \geq e^e \\
&= \frac{3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta}}{\Big((\Delta)(1+3\lambda^2\Gamma_a^2)\ln\ln(\Delta)(1+3\lambda^2\Gamma_a^2)\Big)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta}}{\Big((\Delta)(1+3\lambda^2\Gamma_a^2)\ln\ln(\Delta)^2\Big)^{\frac{1}{2}}} \quad \text{since } \Delta = 1+3\lambda^2\Gamma_a^2
\end{aligned}$$

$$\begin{aligned}
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{\ln\ln\Delta}}{\sqrt{(4\lambda^2\Gamma_a^2)(2\ln\ln\Delta)}} \quad \text{since } 2\ln\ln\Delta > \ln\ln(\Delta)^2 \text{ for } \Delta \geq e^e \text{ and } \lambda\Gamma_a \geq 1 \\
&\geq \lambda\Gamma_a.
\end{aligned}$$

Since $\frac{x-\Delta}{\sqrt{x\ln\ln x}}$ is an increasing function for $x \geq e^e$ and from (3.22), we have that $b \geq n$.

The final step follows from comparing bound on n in terms of Δ with the upper bound in the Lemma (3.21):

$$\begin{aligned}
n &\leq b \\
&= \Delta + 3\lambda^2\Gamma_a^2\sqrt{\Delta\ln\ln\Delta} \\
&\leq \Delta(1+\sqrt{\Delta\ln\ln\Delta}) \quad \text{by definition of } \Delta \\
&= (1+3\lambda^2\Gamma_a^2)\big(1+\sqrt{(1+3\lambda^2\Gamma_a^2)\ln\ln(1+3\lambda^2\Gamma_a^2)}\big) \\
&\leq (3.1\lambda^2\Gamma_a^2)(1.1\sqrt{3.1\lambda^2\Gamma_a^2\ln\ln(3.1\lambda^2\Gamma_a^2)} \quad \text{since } \lambda\Gamma_a \geq 2e^{2e} \text{ by (3.6)} \\
&\leq 7.2(\lambda\Gamma_a)^3\sqrt{\ln\ln\lambda\Gamma_a} \quad \text{since } \lambda\Gamma_a \geq 2e^{2e} \text{ by (3.6)} \\
&\leq \frac{2\rho\Psi^2}{1-\rho}\ln\ln\frac{\rho\Psi}{2(1-\rho)} + \mu(m-1)B
\end{aligned}$$

where $\Psi = \Big(3\lambda^2\Gamma_a^2 + 2\Gamma_s\mu\big((2+6\lambda^2\Gamma_a^2)\big)^{\frac{1}{2}}\Big)$. □

For the rest of the proof, we assume

$$\lambda t \geq 1 + 3\lambda^2\Gamma_a^2 \tag{3.23}$$

Lemma 3.10 *Assuming condition (3.23) holds, then*

$$n \leq t\lambda + 3\lambda^2\Gamma_a^2\phi(t\lambda).$$

Proof. First suppose $n < e^e$: In this case, $n \leq \lambda t + \lambda\Gamma_a$ by (3.3). The statement of the Lemma follows trivially since $\lambda\Gamma_a \geq e^e$ by (3.6) and $\phi(x) \geq 1$.

For the rest of the proof, suppose $n \geq e^e$: Applying (3.3), we obtain $n\lambda^{-1} - \Gamma_a\sqrt{n \ln\ln n} \leq t$. Which gives

$$\frac{n - t\lambda}{\sqrt{n \ln\ln n}} \leq \lambda\Gamma_a \leq \lambda\Gamma. \tag{3.24}$$

Define b by: $b = t\lambda + 3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}$.

Observe that:

$$\begin{aligned}
\frac{b - t\lambda}{\sqrt{b \ln\ln b}} &= \frac{3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}}{\left((t\lambda + 3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}) \ln\ln(t\lambda + 3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda})\right)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}}{\left((t\lambda + 3\lambda^2\Gamma_a^2\sqrt{t^2\lambda^2}) \ln\ln(t\lambda + 3\lambda^2\Gamma_a^2\sqrt{t^2\lambda^2})\right)^{\frac{1}{2}}} \\
&\quad \text{since } t\lambda \geq \ln\ln t\lambda \text{ because } t\lambda \geq 1 + 3\lambda^2\Gamma_a^2 \geq (2e^{2e})^2 \text{ by (3.6)} \\
&= \frac{3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}}{\left((t\lambda)(1 + 3\lambda^2\Gamma_a^2) \ln\ln (t\lambda)(1 + 3\lambda^2\Gamma_a^2)\right)^{\frac{1}{2}}} \\
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{t\lambda \ln\ln t\lambda}}{\left((t\lambda)(1 + 3\lambda^2\Gamma_a^2) \ln\ln(t\lambda)^2\right)^{\frac{1}{2}}} \quad \text{since } t\lambda > 1 + 3\lambda^2\Gamma_a^2 \\
&\geq \frac{3\lambda^2\Gamma_a^2\sqrt{\ln\ln t\lambda}}{\sqrt{(4\lambda^2\Gamma_a^2)(2 \ln\ln t\lambda)}} \quad \text{since } 2\ln\ln t\lambda > \ln\ln(t\lambda)^2 \text{ for } t\lambda \geq e^e \text{ and } \lambda\Gamma_a \geq 1 \\
&\geq \lambda\Gamma_a \quad \text{by simplifying above expression.}
\end{aligned}$$

Since $\frac{x-t\lambda}{\sqrt{x \ln \ln x}}$ is an increasing function for $x \geq e^e$ and from (3.24), we have that $b \geq n$ and the result is obtained. □

Lemma 3.11 *Assuming condition (3.23) holds, then*

$$\phi(n) \leq \left((2 + 6\lambda^2\Gamma^2)\right)^{\frac{1}{2}} \phi(t\lambda).$$

Proof.

$$\begin{aligned}
\phi(n) &\leq \phi(t\lambda + 3\lambda^2\Gamma_a^2 \phi(t\lambda)) \quad \text{by Lemma 3.10} \\
&= \phi(t\lambda + 3\lambda^2\Gamma_a^2 \sqrt{t\lambda \ln \ln t\lambda}) \quad \text{condition (3.23) and definition of } \phi(x) \\
&= \sqrt{(t\lambda + 3\lambda^2\Gamma_a^2 \sqrt{t\lambda \ln \ln t\lambda}) \cdot \ln \ln (t\lambda + 3\lambda^2\Gamma_a^2 \sqrt{t\lambda \ln \ln t\lambda})} \\
&\leq \sqrt{(t\lambda)(1 + 3\lambda^2\Gamma_a^2) \ln \ln \left((t\lambda)(1 + 3\lambda^2\Gamma_a^2)\right)} \\
&\leq \sqrt{(t\lambda)(1 + 3\lambda^2\Gamma_a^2) \ln \ln \left((t\lambda)^2\right)} \quad \text{since } t\lambda \geq 1 + 3\lambda^2\Gamma_a^2 \\
&\leq \sqrt{(t\lambda)(1 + 3\lambda^2\Gamma_a^2) 2 \ln \ln (t\lambda)} \quad \text{since } t\lambda \geq 1 + 3\lambda^2\Gamma_a^2 \geq e^e \text{ by (3.6)} \\
&= \sqrt{(t\lambda)(2 + 6\lambda^2\Gamma_a^2) \ln \ln (t\lambda)} \\
&= (2 + 6\lambda^2\Gamma_a^2)^{\frac{1}{2}} \phi(t\lambda).
\end{aligned}$$

□

We now use Lemmas 3.10 and 3.11 to complete the statement of our proof.

Applying Lemma 3.8, we obtain

$$\begin{aligned}
Q_0 &\leq n + 2\Gamma_s \phi(n)\mu + \mu(m-1)B - tm\mu \\
&\leq \left(t\lambda + 3\lambda^2\Gamma_a^2 \phi(t\lambda)\right) + 2\Gamma_s \mu \left((2 + 6\lambda^2\Gamma_a^2)\right)^{\frac{1}{2}} \phi(t\lambda) + \mu(m-1)B - tm\mu \\
&\qquad \text{where we apply Lemmas 3.10 and 3.11 to bound } n \text{ and } \phi(n) \\
&= -m\mu(1-\rho)t + \phi(t\lambda)\left(3\lambda^2\Gamma_a^2 + 2\Gamma_s \mu \left((2 + 6\lambda^2\Gamma_a^2)\right)^{\frac{1}{2}}\right) + \mu(m-1)B \\
&= -\frac{1-\rho}{\rho}(t\lambda) + \phi(t\lambda)\left(3\lambda^2\Gamma_a^2 + 2\Gamma_s \mu \left((2 + 6\lambda^2\Gamma_a^2)\right)^{\frac{1}{2}}\right) + \mu(m-1)B \\
&\qquad \text{where we apply definition } \rho = \frac{\lambda}{m\mu}
\end{aligned}$$

We denote the RHS of the last expression by $U(x) = -ax + 2b\phi(x) + c$ where $a = \frac{1-\rho}{\rho} > 0$, $b = \frac{1}{2} \cdot \left(3\lambda^2\Gamma_a^2 + 2\Gamma_s\mu\big((2 + 6\lambda^2\Gamma_a^2)\big)^{\frac{1}{2}}\right)$, and $c = \mu(m-1)B$. Observe that

$$\frac{b}{a} \geq \frac{\rho 3\lambda^2\Gamma_a^2}{2(1-\rho)} \geq \rho\Gamma_a \cdot \lambda\Gamma_a \cdot \lambda \geq e^{2e}$$

where the last inequality follows by (3.6).

We invoke Proposition A.3 to obtain the desired result:

$$Q_0 \leq \max_{x \geq 0} U(x) \leq \frac{7\rho\Psi^2}{4(1-\rho)} \ln\ln \frac{\rho\Psi}{2(1-\rho)} + \mu(m-1)B.$$

3.6 Conclusion

We have built upon the approach developed in Chapter 2 and applied it to the performance analysis of $GI/GI/m$ system. The essence of the approach lies in replacing stochastic primitives of the underlying queueing system with deterministic quantities that satisfy the implications of some probability laws. Using this approach, we have managed to derive explicit upper bounds on waiting times and queueing lengths. We also showed that the bounds implied by the Law of the Iterated Logarithm are applicable for the underlying stochastic queueing system leading to explicit and non-asymptotic performance bounds on the same performance measures. Overall, this suggests that this type of modeling approach for performance analysis is both tractable and is capturing underlying stochastic behavior (derived bounds qualitatively agree with Kingman bounds up to $\ln\ln(1-\rho)^{-1}$ factor). Additionally, we have shown that this type of analysis yields bounds for waiting times in the Halfin-Whitt regime, which agrees with asymptotics obtained in the stochastic setting up to a constant additive factor. It would be an interesting research endeavor to see if one can reproduce exactly the stochastic steady state results in Halfin-Whitt regime through tractable robust formulations that hold with high probability.

Chapter 4

(s,S) Policies in Supply Chain Networks: Robust vs. Stochastic Optimization

4.1 Introduction

Supply chain management is a significant problem which has received considerable attention both in industry and academia. In 1960, Scarf (Scarf 1960) first proved the optimality of (s,S) policies in a single installation model. In the same year, the pioneering work of Clark and Scarf (Clark and Scarf 1960) showed that basestock type policies are optimal for serial supply chains in the absence of capacity constraints, and that the optimal ordering policy for the entire multiechelon system can be decomposed into decisions based solely on echelon inventories. In addition to being optimal in a variety of theoretical settings, basestock type policies are also preferred by companies due to their innate simplicity in implementation. Further work in generalizing, extending, and refining optimality results of basestock policies has been done by Federgruen and Zipkin (Federgruen and Zipkin 1984), Rosling (Rosling 1989), Langenhoff and Zijm (Langenhoff and Zijm 1990), Muharremoglu and Tsitsiklis (Muharremoglu and Tsitsiklis 2008), Huh and Janakiraman (Huh and Janakiraman 2008), among many others. Sethi and Cheng (Sethi and Cheng 1997) proved the optimality of (s,S) policies in a more general setting - with

Markovian demand. For a thorough review of inventory theory, see Zipkin (Zipkin 2000).

The question of computing optimal basestocks in general supply chain networks is a difficult problem for two reasons. First, it is a complex stochastic optimization problem in parameter space for which there is not a general exact algorithm available. Second, in reality, only data of demand histories are available, and hence it is not clear which probability distribution is the true source of uncertainty.

Some work on computing basestocks in supply chain networks includes Glasserman and Tayur (Glasserman and Tayur 1994; Glasserman and Tayur 1995) and Fu (Fu 1994) who designed simulation based methods to compute basestock policies based on infinitesimal perturbation analysis (IPA). Other methods that aim to compute basestocks include Roundy and Muckstadt (Roundy and Muckstadt 2000) and Rong et al. (Rong, Bulut, and Snyder).

Robust optimization addresses the issue of data uncertainty without assuming specific probability distributions for the unknown parameters. Instead, the spirit of the approach is to use historical data to model randomness with uncertainty sets. The construction of such sets is informed by the laws of probability. Bertsimas and Thiele (Bertsimas and Thiele 2006) first applied robust optimization to inventory theory and proved that basestock type policies are optimal in the robust model. Ben-Tal et al. (Ben-Tal, Golany, Nemirovski, and Vial 2005) advanced this approach by efficiently computing affinely adjustable order policies for a two-echelon model. For a review of robust optimization see the survey by Bertsimas et al. (Bertsimas, Brown, and Caramanis 2011) and the book by Ben-Tal et al. (Ben-Tal, Ghaoui, and Nemirovski 2009). Some other work on distribution-free approaches to inventory theory, but not based on robust optimization, include Scarf (Scarf 1958), Kasugai and Kasegai (Kasugai and Kasegai 1961), and Gallego and Moon (Gallego and Moon 1993; Gallego and Moon 1994). Graves and Willems (Graves and Willems 2000) also develop a framework for optimizing safety stock placement in supply chains in a distribution-free manner.

Our goal in this chapter is twofold: a) to propose methods for computing (s,S) policies in supply chain networks using robust and stochastic optimization; b) to gain insights into the relative performance of robust (ROB) and stochastic (STO) policies. Our method builds on the technique of Bienstock and Özbay (Bienstock and Özbay 2008), who designed an algorithm to compute basestock parameters in a robust setting for a single echelon problem. While

(Bienstock and Özbay 2008) deals with the single echelon problem without fixed costs, our focus is to handle more realistic inventory problems including multiechelon systems with general topologies and cost functions (including fixed costs). As basestock type policies have enjoyed success in theoretical results and popularity among companies for ease and intuitive implementation, we focus on the problem of computing basestocks in a multiechelon model. In summary, the contributions of this chapter are as follows:

- By extending the applicability of (Bienstock and Özbay 2008) to general networks and cost structures, we propose algorithms based on simulated annealing that compute robust and stochastic (s,S) policies for supply chain networks. The algorithms tackle general network topologies and not just the standard serial or tree system networks. In addition, we assume general cost functions and the presence of fixed costs, which is usually ignored in most theoretical and computational results. The algorithms are implemented for networks (up to 8 installations) and show practical running times.

- In an extensive numerical study, we compare the performance of robust (ROB) vs. stochastic (STO) (s,S) policies and offer insight into their relative performance. In general, we find that ROB sacrifices little in average performance against STO, while having a considerably lower standard deviation and 5%-tail. Additionally, we identify regimes where ROB outperforms STO even in average performance.

The chapter is organized as follows. Section 4.2 presents notation and introduces the setup of the problem. Section 4.3 explains the algorithms and the implemention details. Section 4.4 discusses the performance of algorithms and presents a detailed numerical study comparing performance of the robust and stochastic (s,S) policies. Section 4.5 presents concluding remarks and directions for future research.

4.2 The Model

4.2.1 Notation and Dynamics of the General Assembly System

As discussed in the introduction, we consider a multi-echelon system in which every installation follows an (s,S) policy for echelon inventory. We begin by briefly introducing notation and dynamics of a multi-echelon system, explaining the cost structure and the mechanics of this basestock type policy, and finally consider two performance measures of interest.

The main storage hubs receive their supplies from outside manufacturing plants and send items throughout the network, each time bringing them closer to their final destination, until they reach the stores (the sinks of the network). Put another way, sinks are where the outside demand occurs, and it is the demand at the sinks that drives the orders within network. Note, our model easily allows for the possibility for external demand (sinks) to occur in intermediate stages of the network, as well. We let J be the number of installations, and S be the set of sink nodes. In the case where all installations face outside demand, $S = \{1, \ldots, J\}$. We consider a T period time horizon.

We define echelon $Ech(k)$ for $k = 1, \ldots, J$ to be the set of all the installations, including k itself, that can receive stock from installation k, and the links between them. This is the definition used by (Clark and Scarf 1960) when they consider tree networks. In the special case of series systems, we number the installations such that for $k = 1, \ldots, J$, the items transit from installation $k+1$ to k, with installation J receiving its supply from the plant and installation 1 being the only sink node, as in (Clark and Scarf 1960) and $Ech(k) = \{k, k-1, \ldots, 1\}$. In that case, the demand at installation $k+1$ at time t is the amount of stock ordered by installation k at time t.

Throughout the rest of the chapter, we will use boldface symbols, e.g. $\mathbf{x} \in \Re^T$, to denote a vector of scalar quantities $(x_1, x_2, \ldots, x_T)$. Let $N(k)$ be the set of installations supplied by installation k, $O(k)$ be the set of installations that supplies installation k, and $S(k)$ the set of sink nodes in echelon k. We also define for $k = 1, \ldots, J$,

- $I_k(t)$: stock available at the beginning of period t at installation k,
- $X_k(t)$: stock available at the beginning of period t at echelon k,

- $d_{i_k k}(t)$: stock ordered at the beginning of period t at installation k from its supplier $i_k \in 1, \ldots, J$,

- $w_s(t)$: demand at sink node $s \in S$ during period t,

- $\mathbf{w}$: a $|S| \times T$ vector of sink demands with $w(s,t) = w_s(t)$,

- $\mathbf{s}$, $\mathbf{S} \in \Re^J$: vector of lower and upper echelon basestock levels. We assume that the levels are time-invariant $\left(s_k = s_k(t_1) = s_k(t_2),\ \forall k, t_1, t_2\right)$.

Though our algorithm easily extends to nonzero lead times, we assume zero leadtimes throughout the network. In addition, we allow intermediate shortage of excess demand at all of the installations and backloging at the sinks. In Section 4.2.4, we discuss how to extend the standard model we consider to include no intermediate backlogging, non-zero leadtimes, and capacitated orders. By changing units if necessary, we may assume that all of the installations order parts in equal quantities from each of its suppliers, i.e., $d_{i_{k_1} k}(t) = d_{i_{k_2} k}(t)$ for $i_{k_1}, i_{k_2} \in O(k)$. Given a set of all echelon basestock levels (s_k, S_k) and sink demands $w_s(t)$ for $s \in S$, $t \in 1, \ldots, T$, the dynamics of the installation and echelon inventories are for $t = 1, \ldots, T$:

$$I_k(t+1) = I_k(t) + d_{i_k k}(t) - \sum_{j \in N(k)} d_{ij}(t) - w_k(t)\mathbb{1}_{[k \in S]}, \tag{4.1}$$

$$X_k(t+1) = \sum_{j \in Ech(k)} I_j(t). \tag{4.2}$$

d_{ij} are determined by the echelon inventory and basestock levels according to the following equation:

$$d_{i_k k}(t) = \begin{cases} S_k - X_k(t), & X_k(t) \leq s_k; \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

for $t = 1, \ldots, T$.

The dynamics (4.3) indicate that individual installations meet 100% of the demand to them and thus can have negative inventory or intermediate shortage. This is the model we will use in the numerical experiments. However, our setup is flexible to be adapted to the standard

multiechelon model described in (Clark and Scarf 1960) without intermediate shortage, as well as capacitated orders.

Finally, we specify the cost function. At each installation, we assume four types of costs present at period t:

- $c_k(t)$: variable cost per unit item ordered by installation k,
- $K_k(t)$: fixed cost of order by installation k,
- $h_k(t)$: holding cost per unit inventory at installation k,
- $p_k(t)$: backorder penalty cost per unit of negative inventory at installation k.

We denote by $\Pi(\mathbf{s}, \mathbf{S}, \mathbf{w})$ to be the total cost of operation with basestock parameters $\mathbf{s}, \mathbf{S}$ and realized demand $\mathbf{w}$. Thus, if installation k orders $d_k(t)$ from its suppliers at period t, the total cost incurred in period t by installation k is:

$$cost_k(t) = c_k(t) \cdot d_k(t) + K_k(t)\mathbb{1}_{[d_k(t)>0]} + \max\left(h_k(t)I_k(t+1), -p_k(t)I_k(t+1)\right),$$

and the total cost of operation is:

$$\Pi(\mathbf{s}, \mathbf{S}, \mathbf{w}) = \sum_{k=1}^{N}\sum_{t=1}^{T} c_k(t) \cdot d_k(t) + K_k(t)\mathbb{1}_{[d_k(t)>0]} + \max\left(h_k(t)I_k(t+1), -p_k(t)I_k(t+1)\right) \quad (4.4)$$

Note that the term $I_k(t+1)$ appears because this is the amount of inventory that remains at the end of period t and is either stored or backlogged.

4.2.2 Robust vs. Stochastic Optimization

In the robust problem, we assume that the vector of outside (sink) demands $\mathbf{w} \in \Re^{|S|\times T}$ is uncertain and lies in uncertainty set $P_{\mu,\sigma}$ that is specified by the user (see Section 4.2.3 and

Equations (4.7),(4.8)). We now present the complete robust optimization problem (4.5):

$$\min_{\mathbf{s},\mathbf{S}\in\Re^T} \max_{\mathbf{w}\in P_{\mu,\sigma}} \Pi(\mathbf{s},\mathbf{S},\mathbf{w}) \tag{4.5}$$

s.t.

Equations (4.1),(4.2),(4.3),(4.4).

In other words, Problem (4.5) describes the problem of minimizing the total operational cost Π subject to the worst case corresponding realization of demand vector $\mathbf{w}$. The constraints are simply the definition of Π and inventory dynamics from (4.1)-(4.3).

In contrast with the robust approach, the traditional way of inventory optimization has been to model the vector of sink demands $\mathbf{w}$ as a $|S|\times T$ dimensional random variable with distribution $\mathscr{W}$. Thus the stochastic version of Problem (4.5) is as follows, which we will henceforth refer to as Problem (4.6):

$$\min_{\mathbf{s},\mathbf{S}\in\Re^T} \mathbb{E}_{\mathscr{W}}\left[\Pi(\mathbf{s},\mathbf{S},\mathbf{w})\right] \tag{4.6}$$

s.t.

Equations (4.1),(4.2),(4.3),(4.4).

4.2.3 Designing Uncertainty Sets

For the robust formulation problem (4.5), we design uncertainty sets for the demand vector $\mathbf{w_s}$ at each sink $s \in S$, using a combination of interval uncertainty and the central limit theorem type uncertainty. In particular, assuming historical demand has mean μ and std. σ, we create the following polyhedral uncertainty set for $w_s(t)$, $t = 1,\ldots,T$:

$$w_s(t) \in [\mu-\sigma, \mu+\sigma], \tag{4.7}$$

$$\frac{\left|\sum_{i=1}^{t}(w_s(i)-\mu)\right|}{\sigma\sqrt{t}} \leq 3,\ \forall t = 1,\ldots,T. \tag{4.8}$$

Going forward, we denote by $P_{\mu,\sigma}$ the polyhedron of feasible demand vectors $\mathbf{w}_s$ with respect to constraints (4.7)-(4.8). The first constraint (4.7) implies that most demands will land within

one standard deviation from the average. However, this constraint alone still allows for the possibility that all of the realized demand will occur either one σ above or one σ below μ. This is probabilistically unlikely since we expect some demands to fall above the mean and others below the mean. To account for this, we introduce the second linear constraint (4.8) - which belongs to a class of uncertainty constraints known as "budgets of uncertainty" (Bertsimas and Sim 2004). Intuitively, (4.8) has the form of the central limit theorem which says that a sum of n zero-mean random variables normalized by $\sqrt{n}$ is 99% of time within 3σ. Since the robust problem considers only the worst case objective, the second constraint eliminates probabilistically unlikely corner point realizations that will result in a conservative policy. In practice, only using interval style constraints on demand may result in overly conservative solutions. Note, that we do not impose constraints that couple demands from different sinks.

4.2.4 Extensions

In this section, we outline how to extend our model (4.1) - (4.3) to include features such as no intermediate backlogging, capacititated orders, and non-zero leadtimes.

Extension to system without intermediate backlogging. We first explain how our model extends to the multiechelon assembly system without backlogging at intermediate echelons - such as the one described by (Clark and Scarf 1960), which we will call the standard model. We assume that there are no capacity constraints. Consider the N-installation network below in Figure 4-1.

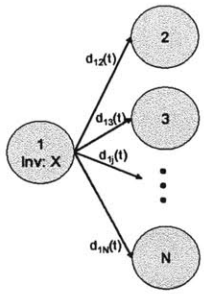

Figure 4-1: An N installation network.

In our model, we assume that Installation 1 meets all of the orders that are posed to it, regardless of the amount of inventory in Installation 1, which we denote by x. If $x \geq d_{12} + \ldots + d_{1N}$, then there is no backlogging in Installation 1 and the dynamics of the inventory of our model agree with the standard one. Now suppose $x \leq d_{12} + \ldots + d_{1N}$. Our model still assumes that Installation 1 meets all its demand, and in the event it will have negative inventory at the end of the period, it will be penalized with its corresponding backlogging cost p_1. However, the standard model will only send the number of items it has in inventory to Installations $2, 3, \ldots, N$. Since the standard model cannot satisfy all of the orders fully, it uses some demand allocation policy to decide how much to send to each installation. Examples of such demand allocation policies include: a) an apriori priority hierarchy that will meet demands in a greedy manner in the order $d_{12}, d_{13}, \ldots, d_{1N}$; b) a policy that can attempt to balance shortfalls, allocating inventory to minimize the resulting difference between inventories and basestock levels.

Extension to production capacities. Suppose we have arc capacities $c_{12}, \ldots, c_{1N}$ enforcing that each order $d_{1j} \leq c_{1j}$. This is handled similarly to the case of no intermediate backlogging. However, in this case we will also provide as input capacity constraints so that the demand allocation policy cannot send more than c_{1j} items across arc $(1, j)$ regardless of order amount.

Extension to non-zero leadtimes. We now explain how to extend our model (4.1) - (4.3) to include non-zero leadtimes. The two basic models that are the building blocks of any system and the ones that are addressed extensively in past literature are the cases of series systems and tree-like systems. In the case of a serial system with intermediate leadtimes, it is evident that there is no substantial change between our model with zero leadtimes and that with non-zero leadtimes, with the exception of having to keep better accounting of units that are yet to arrive and units that have already arrived downstream and are ready for assembly into larger pieces. In other words, in modeling a series system with non-zero leadtimes one has to make sure not to assemble products that have been ordered but have not yet arrived due to leadtimes.

We now address the case of non-zero leadtimes in a tree-like system. For illustration purposes, consider the three installation system in Figure 4-2, and suppose that there is a leadtime $l_{13} = 1$ on orders between Installations 1 and 3, and $l_{23} = 0$. If all installations follow $(\mathbf{s}, \mathbf{S})$ policies, then Installation 3 will result in equal order quantities of material 1 and 2. However, because $l_{13} = 1$, material from Installation 1 will always arrive at Installation 3 one period after the

material 2 order, and we will always end up with a surplus of material 2 (and hence a holding cost) at Installation 3. We model these types of systems as follows.

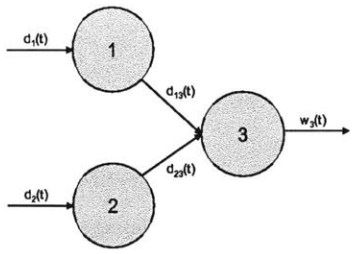

Figure 4-2: 3-Installation Setup

The first method involves having two different basestock parameters at Installation 3, (s_{31}, S_{31}) for orders from Installation 1 and (s_{32}, S_{32}) for orders from Installation 2. This will allow the flexibility of different order quantities from Installations 1 and 2 to compensate for the different leadtimes l_{13} and l_{23}. The downside of this approach is the increased complexity of the model since now we are optimizing over an additional basestock parameter. A second approach that one can use is to transform the assembly system into an equivalent series system via the approach of Rosling (Rosling 1989) according to leadtime quantity. In our example, to take into account that orders $d_{13}(t)$ will always arrive one period after $d_{23}(t)$, we can model the parallel system as a series system as shown in Figure 4-3. As mentioned earlier, a series system can be modeled easily with our standard approach.

4.3 The Algorithms for Robust and Stochastic (s,S) Policies

4.3.1 Robust Algorithm

In this section, we propose an algorithm to solve Problem (4.5). We follow the approach of (Bienstock and Özbay 2008) of using a Bender's Decomposition type routine. We divide

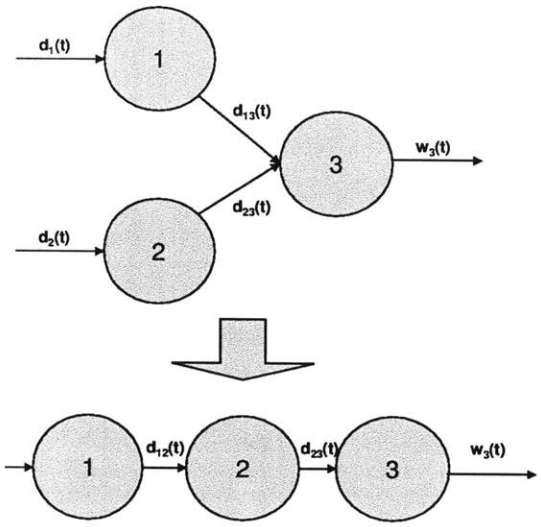

Figure 4-3: Parallel to series transformation.

Problem (4.5) into two parts: a decision maker's problem (DM) and the adversary's problem (ADV). The (DM) problem is to choose the best possible $(\mathbf{s}, \mathbf{S})$ that minimizes the highest cost $\Pi(\mathbf{s}, \mathbf{S}, \mathbf{w})$, when $\mathbf{w} \in \widetilde{W}$ and $\widetilde{W} = \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^n\}$ is a set of n possible demand realizations. The (ADV) problem involves finding the demand realization $\mathbf{w} \in P_{\mu,\sigma}$ (the original uncertainty set) that maximizes $\Pi(\mathbf{s},\mathbf{S},\mathbf{w})$ given the current basestock parameters are $(\mathbf{s}, \mathbf{S})$. The generic algorithm looks as follows (same as Generic Algorithm in (Bienstock and Özbay 2008)):

Algorithm 4.1 *Generic Algorithm.* ***Initialize:*** $\widetilde{W} = \{(\mu, \mu, \ldots, \mu) \in \Re^{|S|\times T}\}, L = 0, U = +\infty$ *tolerance* $\epsilon > 0$.

1. ***Decision Maker's Problem (DM).*** *Let* $(\mathbf{s}^*, \mathbf{S}^*)$ *and* L *be an optimal solution and the objective value, respectively, of the problem:*
   $L = \min\limits_{\mathbf{s},\mathbf{S}} \max\limits_{\mathbf{w}\in\widetilde{W}} \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w}), \quad (\mathbf{s}^*, \mathbf{S}^*) = \operatorname{argmin}\limits_{\mathbf{s},\mathbf{S}} \max\limits_{\mathbf{w}\in\widetilde{W}} \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w}).$
2. ***Adversarial Problem (ADV).*** *Let* $\overline{\mathbf{w}}$ *and* U *be the optimal solution and objective value, respectively, of the problem:*
   $U = \max\limits_{\mathbf{w}\in P_{\mu,\sigma}} \Pi(\mathbf{s}^*, \mathbf{S}^*, \mathbf{w}), \quad \overline{\mathbf{w}} = \operatorname{argmax}\limits_{\mathbf{w}\in P_{\mu,\sigma}} \Pi(\mathbf{s}^*, \mathbf{S}^*, \mathbf{w}).$
3. ***Termination Test:*** *If* $U - L < \epsilon$, *then* ***EXIT.***

4. ***Formulation Update:*** *Otherwise, add* $\overline{\mathbf{w}}$ *to* $\widetilde{W}$ *and return to Step 1.*

We begin the algorithm by initializing $\widetilde{W}$ to contain only the demand vector corresponding to mean historical demand μ. Then, we proceed to solve (DM), followed by (ADV), and either terminate the algorithm or increase the number of elements in $\widetilde{W}$. Note that in principle, Step 1 of the Algorithm looks like Problem (4.5) itself. However, it is in fact much easier. In (4.5), the maximum is taken over a polyhedron of possible sink demand vectors, while in Algorithm 4.1, the maximum is taken over a finite collection of possible realizations of sink demand vectors $\widetilde{W}$. Thus, if the set $\widetilde{W}$ is not too large, we can hope to be able to solve (DM) quickly. Observe that $|\widetilde{W}|$ = number of iterations of the algorithm. From the mechanics of Bender's Decomposition, it is reasonable to hope for convergence after a small number of iterations. In fact, convergence after few iterations ($\approx$ 6) was observed in (Bienstock and Özbay 2008), and we also observe similar behavior.

Before proceeding, we would like to comment on the difficulty of each subproblem:

1. The presence of fixed order costs make both (DM) and (ADV) require integer variables. The ordering constraint enforcing the (s,S) ordering rule (4.3) is not convex and further contributes to the difficulty of (DM) and (ADV) problems. In one type of uncertainty set, even without fixed-order costs and for a single echelon, (ADV) is shown to be NP-hard (Özbay 2006).

2. The optimal solution of (ADV) is not guaranteed to occur at the vertex of the original uncertianty set $P_{\mu,\sigma}$ (see Proposition 4.2). Otherwise, knowing that the optimal solution occurs at the corner point of the uncertainty polyhedron could potentially lead to faster heuristics.

3. One could attempt to solve (DM) as a mixed-integer optimization problem (MIP). However, we observed from numerical experiments that computational time for (DM) with a single installation quickly increased with each iteration of the algorithm and thus with the size of $\widetilde{W}$. Additionally, we observed that with higher number of periods, T, the MIP computation time of (DM) increased drastically. In fact, (Bienstock and Özbay 2008) reported similar large increases in computation time for MIP implementation of the Ad-

versarial Algorithm for their problem. This experience motivated the idea to look for another approach.

Proposition 4.2 *For a given* $(\mathbf{s}, \mathbf{S})$ *policy, the optimal solution to (ADV) is not guaranteed to lie on a corner of the uncertainty polyhedron* $P_{\mu,\sigma}$.

Proof. Consider a simple single factory newsvendor problem with $T = 2$ periods, and costs $h = 1, p = 3, K = 2, c = .5$. Suppose the policy is $s = 1, S = 2$. Let the demand uncertainty polyhedron $P_{\mu,\sigma}$ consist simply of the space $\mathbf{w} \in [0, 2]^2$. Given that we are forced to follow the (s,S) policy, Table 4.1 summarizes the cost of the policy for each corner realization of demand. Observe that in fact the highest cost occurs at $(1.00001, 0)$ - not a corner point of $P_{\mu,\sigma}$. □

w(1)	**w(2)**	**cost**
0	0	7
2	0	8
0	2	5
2	2	6
1.0001	**0**	**8.49995**

Table 4.1: Costs for various demand realizations.

As a result of Proposition 4.2, we should not expect corner points to occur under more complicated uncertainty descriptions and networks.

4.3.2 Simulated Annealing and the Robust Algorithm Implementation

Simulated annealing (SA) is a probabilistic metaheuristic and has had a long history of successful implementations (Ingber 1993). In particular, it is known to be good, particularly better than gradient based methods, in avoiding local optima. The optimality guarantee is that for any given problem, the probability that the simulated annealing algorithm terminates

with the global optimal solution approaches 1 as the annealing schedule is increased to infinity (Bertsimas and Tsitsiklis 1993). It is known that in practice simulated annealing can give good solutions with reasonable running time.

Recall that in (DM) we are computing a min max where the max is taken over $\widetilde{W} = \{\mathbf{w^1}, \ldots, \mathbf{w^n}\}$. The long computation time associated with a MIP formulation led us to believe that this may not be the best approach. Observe that once we fix $(\mathbf{s}, \mathbf{S})$ and $\widetilde{W}$, the cost $cost_{\widetilde{W}}(\mathbf{s}, \mathbf{S}) = \max\left(\Pi(\mathbf{s}, \mathbf{S}, \mathbf{w^1}), \ldots, \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w^n})\right)$ can be computed quickly and in parallel.

Since the shape of the cost function $\min_{\mathbf{s},\mathbf{S}} \max_{\mathbf{w}\in\widetilde{W}} \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w})$ has many local minima, we decided to use simulated annealing and not a gradient-descent based approach as the subroutine in Algorithm 4.1. In fact, we found that simulated annealing found better solutions in comparable time as the gradient-descent approach, which quickly got stuck at local minima. As mentioned earlier, simulated annealing is a probabilistic algorithm and is not guaranteed to converge in polynomial time to an optimal solution. Thus, in order to increase the probability of obtaining a good solution in reasonable time, we used the following rules which experimentally drastically improved both the convergence properties and the cost estimates:

1. For all of our experiments (up to network of 8 installations), we used up to 300 seconds per SA run and we found that this was more than enough to guarantee a good solution.

2. If we could solve both (DM) and (ADV) to optimality, then clearly, at each iteration we should have $cost(DM) \leq cost(ADV)$ or $L \leq U$ from the context of Algorithm 4.1. However, because the SA algorithm may not solve to optimality, the inequality might be reversed and we re-run the SA subroutine from a random starting point until we would have a solution such that $cost(DM) \leq cost(ADV)$.

3. Choosing good starting points was instrumental to making SA more effective. For our starting points - we used the optimal solution from the previous iteration. For example, in the case of (DM), we use as the starting point in iteration $i+1$ the optimal solutions from iteration i: $(\mathbf{s^0}, \mathbf{S^0}) = (\mathbf{s^*}, \mathbf{S^*})$. In the case of (ADV), we used as the starting point the arg max produced from the preceding iteration of (DM) $(\mathbf{w^M} = \max_{\mathbf{w}\in\widetilde{\mathbf{W}}} \Pi(\mathbf{s^*}, \mathbf{S^*}, \mathbf{w}))$.

4. In order to solve (ADV), we maximize over the demand uncertainty polyhedron $P_{\mu,\sigma}$. To do this, we include in the objective function a penalty term that contains the polyhedral

uncertainty constraints and a penalty multiplier. For example, if our demand uncertainty set $P_{\mu,\sigma}$ is of the form $A\mathbf{w} \leq b$. Then, we incorporate into the objective function a term $M \cdot (b - A\mathbf{w})$ where M is a large number.

4.3.3 Stochastic Algorithm

In order to compute the stochastic (s,S) policy, we formulate and solve the corresponding stochastic problem as follows. First, instead of specifying a polyhedral uncertainty set as we did in the robust case, we assume a normal distribution and generate M random demand vectors $(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^M) \sim N(\mu, \sigma)$ where μ, σ are the mean and standard deviation, respectively, of the corresponding sink demand. In our experiments we took $M = 1000$ because we observed that generating more than this amount did not substantially change the solution. We approximate problem (4.6) by solving the problem of minimizing the sample average objective $\min \frac{1}{1000} \sum_{1 \leq i \leq 1000} \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w}^i)$ using simulated annealing. Also, we use the same starting point as in Algorithm 1 - optimal $(\mathbf{s}, \mathbf{S})$ assuming constant demand with $w(t) = \mu$, which was generated by (DM) in the first step of the Algorithm 4.1.

4.4 Numerical results

In this section, we present numerical results. The section is outlined as follows: Section 4.4.1 discusses the effectiveness of simulated annealing. The networks used in the study are presented in Section 4.4.2. Running times are presented in 4.4.3, and the computational study of ROB vs. STO performance is presented in Section 4.4.4.

Going forward, we denote by ROB the robust policy computed by solving Problem (4.5) using Algorithm 4.1 subject to demands in $P_{\mu,\sigma}$ with $(\mu = 100, \sigma = 20)$. Analogously, we denote by STO the stochastic policy computed by solving, using the aforementioned stochastic approach, Problem (4.6) assuming demand is i.i.d. and normally distributed with $(\mu = 100, \sigma = 20)$.

4.4.1 Effectiveness of Simulated Annealing

In this section, we illustrate the effectiveness of simulated annealing in ROB and STO by comparing to an enumeration approach.

Comparison of SA to enumeration approach for the robust model. In order to compare the effectiveness of Algorithm 4.1, we consider a small multiechelon network with $J = 2$, $T = 10$, and polyhedral uncertainty $P_{\mu,\sigma} = \{(w(1), \ldots, w(10)) \in [80, 120] \times \ldots \times [80, 120]\}$ and solve it both with Algorithm 4.1 and with an enumeration approach. For the enumeration approach, we enumerate all of the possible corner points of $[80, 120]^{10}$ and use this as a set of possible demand realizations, $\mathcal{D} = \{$ corner point of $[80, 120]^{10}\}$. Note, this is still an approximation to the polyhedron $P_{\mu,\sigma}$ especially since we know that the optimum does not have to occur at a corner point from Proposition 4.2, but it is a reasonable approximation. Next we create a grid of possible values of basestock parameters $50 \leq s_1, S_1, s_2, S_2 \leq 150$ with increment of 2.5. For each $(\mathbf{s}, \mathbf{S})$ combination from the grid (≈ 0.75 million combinations) we evaluate the corresponding maximum cost $\max_{\mathbf{w} \in \mathcal{D}} \Pi(\mathbf{s}, \mathbf{S}, \mathbf{w})$.

Figure 4-4 shows the histogram of costs corresponding to each $(\mathbf{s}, \mathbf{S})$ combination from the discretization. This particular enumeration scheme took about 5 hours to run, and we compare in Figure 4-4 the histogram of costs produced from the enumeration scheme with the cost of the policy obtained by Algorithm 4.1, which in contrast took about 3 minutes to run. The solution produced by our algorithm is actually 0.02% better than the best one found by the enumeration scheme, probably due to discretization error. Clearly the number of corner points grows exponentially with T and the number of $(\mathbf{s}, \mathbf{S})$ values grows exponentially with the number of installations. Thus, it would be computationally intractable to enumerate even a slightly larger problem, i.e., with $N = 3$ installations or $T = 15$ periods.

Comparison of SA to enumeration approach for the stochastic model. The purpose of this experiment is to measure the effectiveness of SA as a subroutine for computing the simulation based stochastic heuristic. To achieve this, we generated a set of random normally distributed demands ($N(100, 20)$). Then, we solved for the optimal (s,S) policy for the same small multiechelon network with $J = 2$, $T = 10$ both by minimizing the sample average via SA and by a similar enumeration of basestocks $(\mathbf{s}, \mathbf{S})$. Figure 4-5 shows the histogram of costs produced by all of the $(\mathbf{s}, \mathbf{S})$ from the enumeration, as well as cost of the policy obtained by SA.

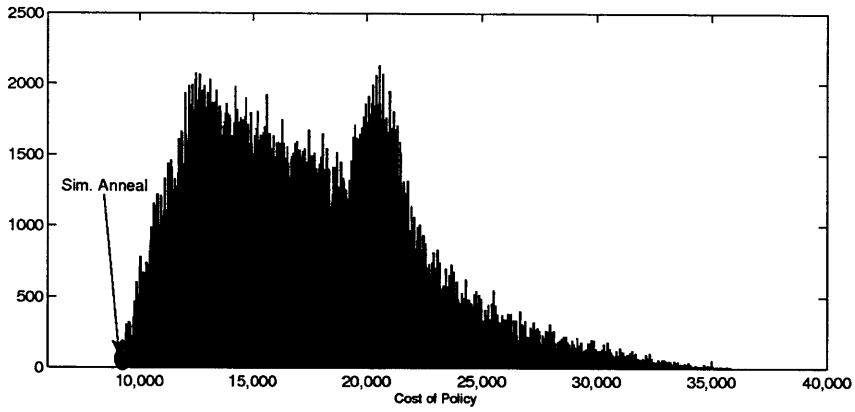

Figure 4-4: Comparison of enumeration scheme and SA for the robust model.

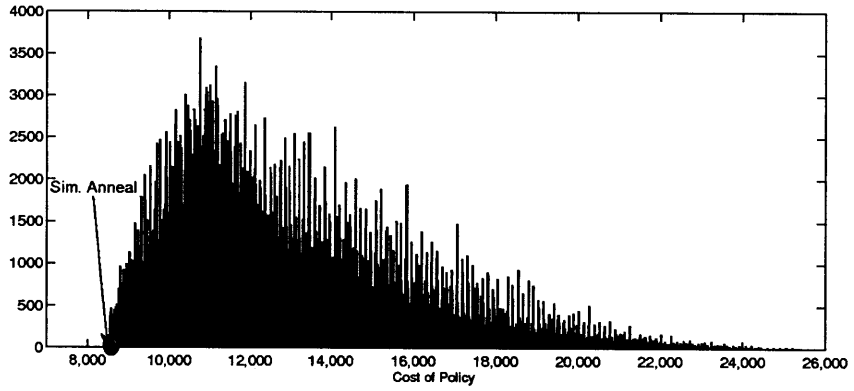

Figure 4-5: Comparison of enumeration scheme and SA for the stochastic model.

SA in a matter of minutes produced a policy that is 0.01% better than the best one found by the enumeration approach.

4.4.2 The Networks

A three-echelon station. Figure 4-6 depicts a three installation system, with a single sink at node 3 which orders material from nodes $1, 2$. Note, since orders placed by node 3 are

determined purely by its outside demand $\mathbf{w}_3$, current inventory, and basestock parameters $(\mathbf{s}_3, \mathbf{S}_3)$, the amount it orders from $1, 2$ are equal, $d_{13}(t) = d_{23}(t)$. We assume that demands $(w_{31}, w_{32}, \ldots, w_{3T})$ are i.i.d. with mean 100 and standard deviation 20. The number of periods we consider is $T = 15$. The order and inventory/backlog costs are summarized in Table 4.2.

A five-echelon station. Figure 4-7 depicts a five installation system containing two sink nodes: 4 and 5. For the sake of simplicity, we again assume that the period demands at sinks 4 and 5 are i.i.d. each with mean 100 and standard deviation 20. The number of periods we consider is $T = 10$. The order and inventory/backlog costs are summarized in Table 4.2. Note, that inventory costs - both holding and penalty increase downstream.

An eight-echelon station. Figure 4-8 depicts an eight installation system. This time we have three sink nodes $4, 7, 8$ with node 4 serving as both an intermediate node to 7 and a sink node with external demands. We again assume that the period demands at sinks 4, 7, and 8 are i.i.d. each with mean 100 and standard deviation 20. The number of periods we consider is T=10. The order and inventory/backlog costs are summarized in Table 4.2.

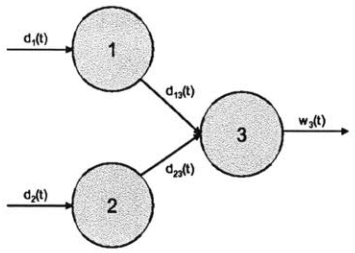

Figure 4-6: 3-Installation Setup

4.4.3 Running Time of the Algorithms

We found the run time of the robust algorithm to be practically reasonable. It takes approximately 1.5 hours for networks of size $J = 8$ and approximately ten minutes for smaller networks (see Table 4.3). Similarly, for the case of the stochastic algorithm, the 3 and 5-server networks were on the order of twenty minutes, and almost 2 hours to solve for the 8-server case.

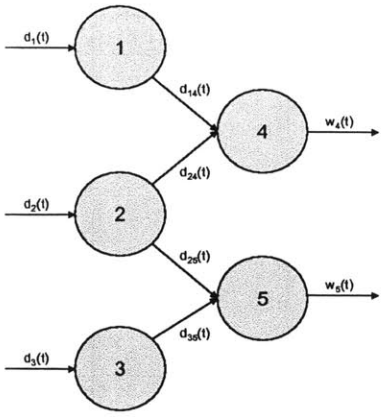

Figure 4-7: 5-Installation Setup

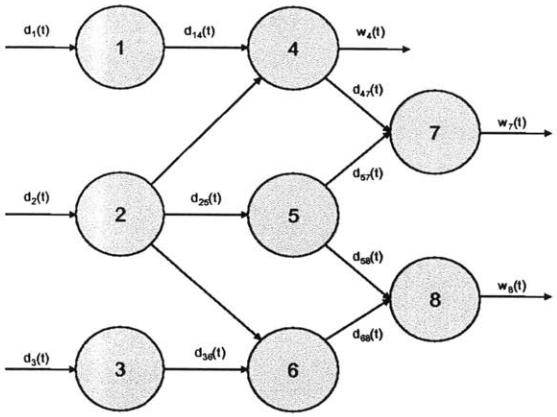

Figure 4-8: 8-Installation Setup

4.4.4 Performance of Robust and Stochastic (s,S) Policies

In this section we study the relative performance of ROB vs. STO. The questions we seek to address are:

- mean,variance, and tail analysis (mean of highest 5%) of robust and stochastic solutions,
- worst case performance of robust and stochastic solutions,
- regimes when robust is more favorable than stochastic approach, and vice-versa,

Network size	Installation	h	p	c	K
3-installation	1	6	10	1	10
	2	8	15	3	10
	3	10	28	4	25
5-installation	1	6	10	1	10
	2	8	15	2	10
	3	8	15	2	10
	4	14	30	10	10
	5	15	35	10	25
8-installation	1	6	10	1	10
	2	8	15	2	10
	3	8	15	2	10
	4	10	20	6	25
	5	16	28	6	25
	6	16	28	2	25
	7	25	45	8	40
	8	30	55	8	40

Table 4.2: Cost parameters for the network experiments.

network size	**# sinks**	**T**	**ROB runtime (hours)**	**STO runtime (hours)**
3	1	15	0.16	0.35
5	2	10	0.44	0.41
8	3	10	1.61	1.89

Table 4.3: Run time results (in hours).

- the robustness of both methods with respect to changes in probability distributions.

We next present the computational studies to compare the performance of ROB and STO in various scenarios.

Min-max comparison with polyhedral uncertainty. The purpose of this set of experiments is to measure the effectiveness of ROB vs. STO with respect to the worst case demand realization. This is especially relevant for highly risk-averse planners. To accomplish this, we took the corresponding policies ROB and STO of each of the network problems and computed the worst-case cost subject to demand in polyhedra of varying size $P_{\mu,\tilde{\sigma}}$ with $\tilde{\sigma} = \{\sigma, 2\sigma, 3\sigma\}$. As mentioned earlier, we use SA to compute the worst-case cost. These results are summarized

in Table 4.4.

From this table, it is clear that ROB offers consistently better worst case protection than STO. When compared on $P_{\mu,\sigma}$, the uncertainty polyhedron initially used to produce the robust policy, ROB has at least 12% lower cost for all three networks considered. In addition, as we vary the size of the polyhedral uncertainty, ROB continues to have lower cost than STO, throughout.

		$P_{\mu,\sigma}$	$P_{\mu,2\sigma}$	$P_{\mu,3\sigma}$
3-installation	STO	20,929	27,414	34,536
	ROB	18,165	26,293	33,159
	$\frac{STO-ROB}{STO}$	13.2%	4.1%	4.0%
5-installation	STO	41,929	52,763	69,522
	ROB	34,773	47,372	58,272
	$\frac{STO-ROB}{STO}$	17.1%	10.2%	16.2%
8-installation	STO	90,064	113,827	148,033
	ROB	78,508	107,181	129,937
	$\frac{STO-ROB}{STO}$	12.8%	5.8%	12.2%

Table 4.4: Max cost comparison for polyhedral uncertainty of varying size.

Discrete distribution of demand. The purpose of this set of experiments is to measure performance of ROB vs. STO when actual demand is drawn from a discrete random variable with the same $\mu = 100$ and $\sigma = 20$. In situations when actual demand is either high or low, discrete(μ, σ) random variable may be more appropriate for modeling demand than $N(\mu, \sigma)$. In doing so, we record: mean cost, standard deviation of cost, and the average of 5% highest costs. The results are summarized in Table 4.5 and in Figure 4-9.

Table 4.5 shows that ROB offers better risk protection in all areas across the three networks. In particular, ROB has lower average cost by approximately 4%. In addition, ROB performance has about 70% lower standard deviation than that of STO. STO also has heavier 5%-tails by the order of 10%. Figure 4-9 shows a histogram of the ROB and STO policies costs for the 5-installation network when realized demand is drawn from discrete(μ, σ) distribution. This figure clearly demonstrates that ROB has a smaller mean and is distributed more tightly around its mean.

		mean	std.	5% tail
3-installation	STO	18,794	782	20,371
	ROB	17,988	156	18,244
	$\frac{STO-ROB}{STO}$	4.3%	79.9%	10.4%
5-installation	STO	36,719	1,637	40,045
	ROB	34,255	384	34,814
	$\frac{STO-ROB}{STO}$	6.7%	76.5%	13.1%
8-installation	STO	78,824	3,240	85,683
	ROB	77,119	1,054	79,280
	$\frac{STO-ROB}{STO}$	2.1%	67.4%	7.5%

Table 4.5: Comparison of ROB and STO under discrete(μ, σ) random variable realization.

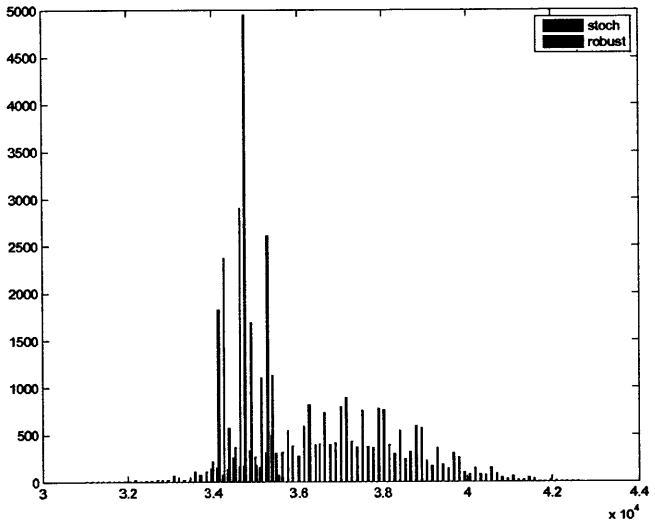

Figure 4-9: 5-installation: ROB vs. STO - discrete realization

Correlated demands. The goal of this set of experiments is to determine the effect that correlation in the realized demand has on comparative performance of ROB and STO. In this set of experiments, we tested the ROB and STO policies on demand drawn from $N(\mu, \sigma)$ with temporal serial correlation of $\rho = 50\%$. In other words, we model demands $w(1) \sim N(\mu, \sigma)$ and $w(t) = \mu + \rho \cdot (w(t-1) - \mu) + \sqrt{1-\rho^2} \cdot N(0, \sigma)$ for $t = 2, \ldots, T$. Then $w(t)$ has mean μ and standard deviation σ and $corr(w(t), w(t-1)) = \rho$ for all t. Note, that for networks with

multiple sinks, we assume that the sinks are not correlated. We did this for both positive and negative correlation, and results are presented in Table 4.6 and Figures 4-10 and 4-11.

		Positive (+50%) Correl			Negative (-50%) Correl		
		mean	std.	5% tail	mean	std.	5% tail
3-installation	STO	17,583	1,870	22,596	17,594	1,339	20,742
	ROB	17,687	1,193	21,024	17,707	1,084	20,411
	$\frac{STO-ROB}{STO}$	−0.6%	36.2%	7.0%	−0.6%	19.1%	1.6%
5-installation	STO	35,076	3,189	43,140	35,217	2,075	40,198
	ROB	35,569	1,443	39,718	35,564	1,141	38,445
	$\frac{STO-ROB}{STO}$	−1.4%	54.7%	7.9%	−1.0%	45.0%	4.4%
8-installation	STO	75,255	6,237	90,990	75,186	3,827	84,146
	ROB	75,512	3,471	84,757	75,496	2,788	82,370
	$\frac{STO-ROB}{STO}$	−0.3%	44.3%	6.9%	−0.4%	27.2%	2.1%

Table 4.6: Comparison of robust and stochastic policies under correlated realized demands.

From the experiments, we see that with respect to the mean, ROB underperforms STO by $\approx 0.6\%$ across the board, though the exact figure depends on the particular network. However, for all three networks at hand, ROB exhibits much better tail behavior. In particular, for the case of positive correlation, the standard deviation of ROB is $\sim 40\%$ lower than STO and 7.0% lower tails. This is is also well illustrated in Figure 4-10 where we see that ROB drops off sharply, while STO has a fatter tail. An organization that is concerned with tail protection might thus prefer ROB to STO. In the case of negatively correlated demand, ROB exhibits similar, albeit slightly lower, performance characteristics - $\sim 30\%$ lower standard deviation and 2.2% lower tails.

Intuitively, the reason ROB-STO outperformance is more pronounced in the positive correlation case is that the total inventory over time has a wider distribution and is more likely to take extreme values. On the other hand, in the presence of negative correlation, a high demand in one period is likely to be coupled with a low demand in the following period and hence total demand is more centered around its average value.

Continuous distribution of demands: Unimodal cases. The purpose of this set of experiments is to investigate the performance of ROB vs. STO when realized demand comes

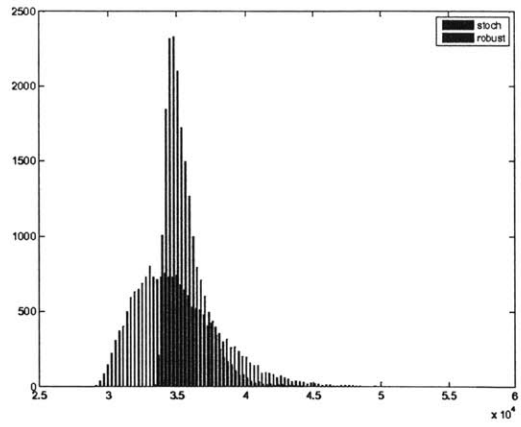

Figure 4-10: 50% realized demand correlation

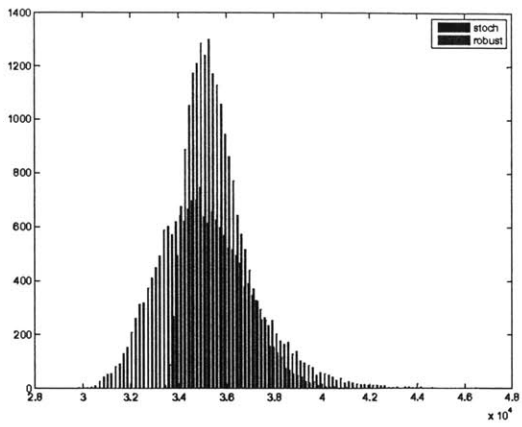

Figure 4-11: -50% realized demand correlation

from a continuous unimodal distribution that is similar to the original demand distribution assumed by STO ($N(\mu, \sigma)$). We draw demands from three such distributions - Normal(μ, σ), Lognormal(μ, σ), and Gamma(μ, σ) and measure the performance of ROB and STO. The results are presented in Table 4.7 and Figure 4-12.

		norm(μ, σ)			lognorm(μ, σ)			gamma(μ, σ)		
		mean	std.	5% tail	mean	std.	5% tail	mean	std.	5% tail
3-installation	STO	17,597	1,324	20,750	17,615	1,502	21,239	17,619	1,429	21,030
	ROB	17,707	941	20,016	17,901	1,124	20,777	17,842	1051	20,440
	$\frac{STO-ROB}{STO}$	-0.6%	29.0%	3.5%	-1.6%	25.4%	2.4%	-1.3%	26.7%	2.8%
5-installation	STO	35,192	2,134	40,081	35,085	2,377	40,672	35,130	2,298	40,497
	ROB	35,561	1,090	38,301	35,984	1,415	39,594	35,857	1,298	39,166
	$\frac{STO-ROB}{STO}$	-1.0%	48.8%	4.4%	-2.56%	40.4%	2.6%	-2.06%	43.4%	3.2%
8-installation	STO	75,184	4,109	84,687	75,020	4,613	85,975	75,113	4,441	85,446
	ROB	75478	2547	81,542	76,064	3,075	83,712	75,921	2,887	82,984
	$\frac{STO-ROB}{STO}$	-0.4%	38.0%	3.7%	-1.4%	33.3%	2.6%	-1.1%	35.0%	2.9%

Table 4.7: Comparison of robust and stochastic policies under random continuous demands.

The results indicate several interesting behaviors. First, we observe that for all three networks, ROB has higher mean costs than STO. However, this is expected since STO was trained to minimize the sample mean on a set of demands drawn from N(μ, σ). However, ROB exhibits

better standard deviation behavior than STO with roughly 35% lower values across the board, and thus, also lower tail costs varying $2.4 - 4.4\%$. Thus, while ROB has slightly worse mean behavior, it exhibits better second moment and tail behavior.

An interesting observation is that STO performs slightly worse when simulated with Normal, than with Lognormal or Gamma. This is somewhat counterintuitive since we trained the STO on Normal demands, and would expect STO-ROB outperformance to be highest on Normal demands. However, note that from Table 4.7 we see that this is happening not because STO is performing better, but because ROB costs are higher for Lognormal and Gamma demands, and this is driving the outperformance. The main reason that is causing ROB to perform worse on Lognormal or Gamma demands than compared to Normal demands is that robust policies inherently assume and are built upon the assumption of symmetric uncertainty. In fact, in our uncertainty sets, we did not model constraints to capture notions such as "there is more uncertainty to the right of the mean than to the left." While the Normal distribution is symmetric, Lognormal and Gamma distributions have positive skewness and hence increase the cost of ROB higher relative to STO. Thus, the modeler must keep in mind that robust policies are best suited for symmetric distributions, and may have worse performance if the true distribution is asymmetric. However, if this is known, we can update the uncertainty set accordingly as in Chen et al. (Chen, Sim, and Sun 2007).

Continuous distribution of demands: Multi-modal cases. The purpose of this set of experiments is to measure the relative performance of ROB vs. STO when the actual demand is realized by a continuous distribution that has qualitatively different properties from the original distribution assumed by STO ($N(\mu, \sigma)$). In particular, STO assumed a Normal distribution of demand, which is unimodal. The goal of these experiments is to determine whether this turns out to be a key assumption for good STO performance. For our purposes, we used uniform and mixture of two normal distributions to simulate realized demands maintaining that the mean and standard deviation of the realized demands are same as in the original assumption - ($\mu = 100, \sigma = 20$). We then observed the performance of both ROB and STO policies on these demands, and the results are presented in Table 4.8.

We observe very good outperformance of ROB. In the case of mixture of normals, we find that the robust policy beats the stochastic one in mean, standard deviation, and in tail behavior. While the same is true for the uniform distribution, the outperformance in terms of mean is not

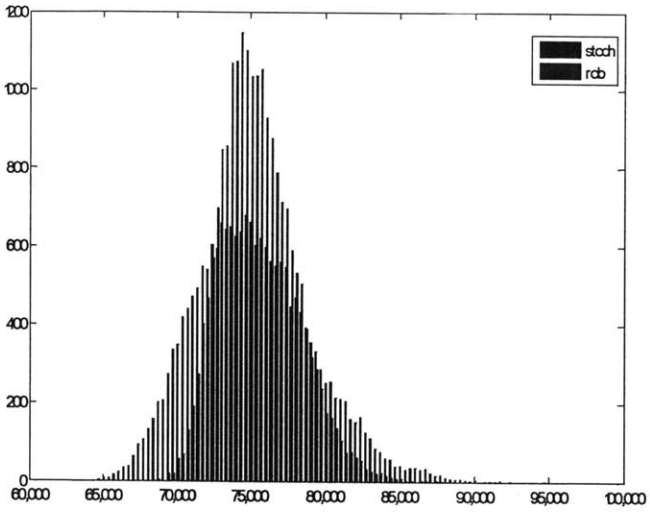

Figure 4-12: 8-installation: ROB vs. STO - Gamma(μ, σ) realization.

		Mixture of Two Normals			Uniform		
		mean	std.	5% tail	mean	std.	5% tail
3-installation	STO	18,216	1,092	20,653	18,001	1,154	20,525
	ROB	17,839	685	19,397	17,883	712	19,444
	$\frac{STO-ROB}{STO}$	2.1%	37.3%	6.1%	0.7%	38.3%	5.3%
5-installation	STO	35,940	1,900	40,040	35,581	1,945	39,828
	ROB	35,314	775	37,137	35,509	779	37,284
	$\frac{STO-ROB}{STO}$	1.7%	59.2%	7.3%	0.2%	59.9%	6.4%
8-installation	STO	77,145	3,722	85,460	76,451	3,755	84,740
	ROB	76,318	2,009	80,873	76,204	2,048	80,765
	$\frac{STO-ROB}{STO}$	1.1%	46.0%	5.4%	0.3%	45.5%	4.7%

Table 4.8: Comparison of robust and stochastic policies under multimodal demands.

as strong. Most importantly, this behavior is consistent across three types of networks. This is in contrast to the experiment in Table 4.7 where we found that robust performed worse in mean but had better standard deviation and tail behavior. Intuitively, this makes sense because in a way, Lognormal and Gamma distributions are very close to the Normal. Moreover, they are not symmetric which handicaps the robust approach. However, the mixture of two normal distributions and uniform distributions are substantially different from the normal distribution, and that is causing problems for the stochastic policy.

Uncertainty in σ. The purpose of this set of experiments is to measure the effect uncertainty in standard deviation of demands has on ROB and STO performance. STO assumed that demand is i.i.d. drawn from $N(\mu = 100, \sigma = 20)$, while ROB assumed that demand lies in $P_{\mu,\sigma}$ as defined by (4.7)-(4.8). We denote the realized standard deviation of demand by $\tilde{\sigma}$. In these experiments, we draw demand from Gamma distribution with mean and realized standard deviation $(\mu, \tilde{\sigma})$ where $\tilde{\sigma}$ varies $\tilde{\sigma} \in \{.5\sigma, .75\sigma, \ldots, 2\sigma\}$. Then, we record the performance of ROB and STO on demands drawn from Gamma distribution with mean μ and standard deviation $\tilde{\sigma}$.

Table 4.9 shows the performance of ROB and STO for the experiments described above with $\tilde{\sigma} = \{0.5\sigma, \sigma, 2\sigma\}$. Figure 4-13 is a plot of the relative performance metrics: % difference in mean, % difference in standard deviation, and % difference in 5%-tail between ROB and STO. For instance, from the figure we see that when realized std $\tilde{\sigma} = 1.25\sigma$, ROB has 35% lower standard deviation, 4% smaller 5%-tail, and the same mean as STO.

		gamma$(\mu, 0.5\sigma)$			gamma(μ, σ)			gamma$(\mu, 2\sigma)$		
		mean	std.	5% tail	mean	std.	5% tail	mean	std.	5% tail
5-installation	STO	31417	1003	33723	35130	2298	40497	43469	5194	55751
	ROB	34026	278.	34764	35857	1298	39166	42179	4094	52327
	$\frac{STO-ROB}{STO}$	-8.3%	72.2%	-3.1%	-2.1%	43.4%	3.2%	3.00%	21.1%	6.1%

Table 4.9: Performance of robust and stochastic policies as a function of realized demand σ.

The graph suggests three regimes. First, when the realized standard deviation turns out to be greater than 1.25σ, ROB outperforms STO both in mean cost and in having a significantly lower standard deviation ($\approx 25\%$) of costs, and lower tails ($\approx 5\%$). In the regime $.75\sigma \leq \tilde{\sigma} \leq 1.25\sigma$, while ROB has lower standard deviation ($\approx 40\%$) and tail costs ($\approx 3\%$), it has higher average costs than STO ($\approx 3\%$). The final case when $\tilde{\sigma} \leq .75\sigma$, ROB has significantly lower standard deviation of cost ($\approx 60\%$). However, ROB has worse average cost performance ($\approx 5\%$) and has slightly worse tail performance ($\approx 1.5\%$) than STO. Thus, in the first two scenarios, it may be advantageous to implement the robust policy, while in the scenario when $\tilde{\sigma}$ turns out to be lower than σ, the robust policy is too conservative.

Also, observe from Table 4.9 that when demand uncertainty is lower than expected, and ROB performs comparatively worse than STO, the mean cost for each is about \$32,000. However,

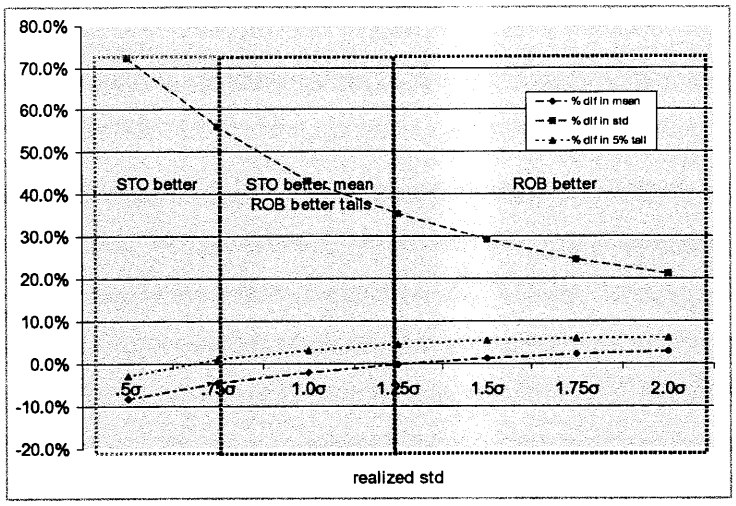

Figure 4-13: Relative % performance of ROB vs. STO as a function of realized $\tilde{\sigma}$.

when demand uncertainty is higher than expected, and ROB is more beneficial than STO, the mean cost is about $42,000 or 30% higher than in the previous scenario. Hence, we infer that ROB performs better than STO particularly in regimes that are higher cost for the user. These results are not surprising since we expect ROB to be relatively less sensitive to larger than expected demand variance since it inherently works by protecting us from corner points, whereas STO works by protecting us from average outcomes. Similarly, when demand variance shrinks, demand realizes closer to the mean, and this is relatively good for STO.

4.5 Conclusion

In this chapter, we have presented an algorithm for computing (s,S) policies in multiechelon supply chains based on robust and stochastic optimization. Our algorithm is designed to handle a broad class of network topologies, uncertainty sets, and cost structures. The algorithm computes (s,S) policies effectively, and thus makes it a reasonable tool for use in practice.

Additionally, we compared the performance of (s,S) policy ROB to STO when STO assumes the normal distribution but the actual distribution can be different. We summarize the key ROB-STO performance insights in Table 4.10. We note that these insights are consistent across

all of the networks presented. The name inside the box indicates which policy performs better across all networks in the given category for the particular set of realized demands.

actual demand distribution	mean	std.	5% tail
Discrete	ROB	ROB	ROB
Positively correlated	STO	ROB	ROB
Negatively correlated	STO	ROB	ROB
Unimodal continuous	STO	ROB	ROB
Multi-modal continous	ROB	ROB	ROB
σ higher than expected	ROB	ROB	ROB
σ lower than expected	STO	ROB	same
Min-max objective	ROB	NA	NA

Table 4.10: Key ROB-STO relative performance insights.

Table 4.10 suggests encouraging results for outperformance of the robust policy compared to the stochastic one across a variety of networks. Particularly in the area of standard deviation and tail behavior, the robust policy performs significantly better than the stochastic one. Additionally, we find that the stochastic policy performs worse than the robust policy when the realized distribution is substantially different from the assumed one.

Chapter 5

Concluding Remarks

In this thesis, we developed new approaches for incorporating uncertainty into optimization, as well as compared existing approaches to gain insights into their relative performance.

In Chapters 2 and 3, we worked in the context of queuing systems. Using ideas from robust optimization we developed a new method for conducting performance analysis of queueing networks. The essence of our approach lies in replacing stochastic primitives of the underlying queueing system with deterministic quantities that satisfy the implications of some probability laws. These implications take the form of linear constraints. As a result, out approach allows us to formulate questions in performance analysis of stochastic queueing systems as deterministic, tractable optimization problems.

We demonstrated our approach on three types of queueing systems: Tandem Single Class queueing networks, Multiclass Single Server queueing systems, and m parallel server system ($GI/GI/m$). Using our approach, we have managed to derive explicit upper bounds on some performance measures such as sojourn times and queue lengths. We also showed that the bounds implied by the Law of the Iterated Logarithm are applicable for the underlying stochastic queueing system leading to explicit and non-asymptotic performance bounds on the same performance measures. We are not aware of any other method of performance analysis which can provide similar performance bounds in queueing models of similar generality.

In Chapter 4, we addressed the question of computing (s,S) policies in multiechelon supply chains based on robust (ROB) and stochastic (STO) optimization. Our algorithms are designed

to handle a broad class of network topologies, uncertainty sets, and cost structures. The algorithms exhibit empirically practical running time. Extensive numerical experiments suggest that both the ROB and STO approaches have their benefits. In particular, we observed that the ROB policy always has better standard deviation and tail behavior of costs. Additionally, the average performance of STO is moderately better than that of ROB. In fact, in some cases when the realized distribution is substantially different from the normal assumption in the STO approach, ROB has lower average cost than STO. Overall, this research suggests that a little (if any) sacrifice in average performance can actually provide robustness and better downside protection.

Our thesis naturally leads to two research paths. The first path lies in strengthening our performance analysis approach and extending it to other areas. In particular, it would be interesting to extend our approach to more accurately capture performance measures such as waiting times and busy periods such that all of the bounds obtained from the robust model agree with existing stochastic results. Additionally, it may be worthwhile to investigate new probability laws or uncertainty set construction that result in a more tight analysis.

With the recent connection established between certain risk measures and robust optimization, it is perhaps not surprising that our robust approach yielded results and bounds that are consistent with traditional stochastic analysis. In essence, the spirit of our approach involves analyzing complex stochastic systems by translating them to tractable deterministic (robust) optimization problems in a way such that insights and bounds obtained for the deterministic problems transfer directly to the underlying stochastic systems. As a result, it would be interesting to extend the "spirit of our approach" to problems in other areas including finance, inventory theory, and other stochastic optimization problems.

Appendix A

Technical Results

In this section we establish some preliminary technical results. Using ϕ as defined by (3.2), we let $U(x) = -ax + 2b\phi(x) + c$ for some positive constants a, b, c satisfying

$$\frac{b}{a} \geq e^{2e}. \tag{A.1}$$

Lemma A.1 *$U(x)$ is strictly concave for $x \geq e^e$.*

Proof.

$$\begin{aligned}
\frac{\partial U(x)}{\partial x} &= -a + b\sqrt{\frac{\ln\ln x}{x}} + \frac{b}{\ln x}\frac{1}{\sqrt{x\ln\ln x}} \\
\frac{\partial^2 U(x)}{\partial x^2} &= b\Big(x^{-\frac{1}{2}}\frac{1}{2}(\ln\ln x)^{-\frac{1}{2}}\frac{1}{\ln x}\frac{1}{x} + (\ln\ln x)^{\frac{1}{2}}(-\frac{1}{2}x^{-\frac{3}{2}})\Big) \\
&\quad + b\Big(-(\ln x)^{-2}\frac{1}{x}(x\ln\ln x)^{-\frac{1}{2}} + (\ln x)^{-1}(-\frac{1}{2})(x\ln\ln x)^{-\frac{3}{2}}(\frac{1}{\ln x} + \ln\ln x)\Big) \\
&= bx^{-\frac{3}{2}}(\frac{1}{2})(\ln\ln x)^{-\frac{1}{2}}\Big(\frac{1}{\ln x} - (\ln\ln x)\Big) \\
&\quad + b\Big(-(\ln x)^{-2}\frac{1}{x}(x\ln\ln x)^{-\frac{1}{2}}\Big) + b\Big((\ln x)^{-1}(-\frac{1}{2})(x\ln\ln x)^{-\frac{3}{2}}(\frac{1}{\ln x} + \ln\ln x)\Big) \\
&< 0 \text{ since all three terms on RHS above are negative for } x \geq e^e
\end{aligned}$$

□

Lemma A.2 *Assuming (A.1) and* $e^e > (c/b)^2$,

$$U(x) < 0 \qquad \forall\, x > (18b^2/a^2)\ln\ln(3b/a).$$

Proof. Since $(18b^2/a^2)\ln\ln(3b/a) > e^e$, throughout the proof we restrict ourselves to the domain $x \geq e^e$. Since in addition $x > (c/b)^2$, we have $b\phi(x) \geq b\sqrt{x} > c$. In this range $-ax+2b\phi(x)+c \leq -ax + 3b\phi(x) = -ax + 3b\sqrt{x \ln\ln x}$. This quantity is less than zero provided

$$\left(\frac{x}{\ln\ln x}\right)^{\frac{1}{2}} > \frac{3b}{a} \triangleq \alpha.$$

It is easy to check that $x/\ln\ln x$ is a strictly increasing function with $\lim_{x\to\infty}(x/\ln\ln x) = \infty$. Let x_0 be the unique solution of $x/\ln\ln x = \alpha^2$ on $x \geq e^e$. We claim that $x_0 \leq 2\alpha^2 \ln\ln\alpha$. The assertion of the lemma follows from this bound. Let $A = 2\alpha^2 \ln\ln\alpha$. Then

$$\begin{aligned}
\frac{A}{\ln\ln A} &= \frac{2\alpha^2 \ln\ln\alpha}{\ln(2\ln\alpha + \ln^{(3)}\alpha + \ln 2)} \\
&\geq \frac{2\alpha^2 \ln\ln\alpha}{\ln(4\ln\alpha)} \quad \text{since } \ln\alpha \geq \ln^{(3)}\alpha \text{ and } \ln\alpha > \ln 2 \\
&\geq \frac{2\alpha^2 \ln\ln\alpha}{2\ln(\ln\alpha)} \quad \text{since } \ln\alpha > \ln(b/a) \geq 2e > 4. \\
&= \alpha^2.
\end{aligned}$$

This implies $x_0 \leq A$ and the proof is complete. □

Proposition A.3 *Under the assumption (A.1)*

$$\sup_{x\geq 0} U(x) \leq 7(b^2/a)\ln\ln(b/a) + c.$$

Proof. Since $a > 0$, then the supremum in $\sup_{x\geq 0} U(x)$ is achieved. Let x^* be any value achieving $\max_{x\geq 0} U(x)$. First suppose $0 \leq x^* < e^e$. It follows from the definition of ϕ in (3.2) that $\phi(x^*) = 1$ and thus $U(x^*) = -ax^* + 2b + c$. Using $0 \leq x^* < e^e$ and assumption (A.1), it is straightforward to check that $U(x^*)$ is indeed upper bounded from above by $7(b^2/a)\ln\ln(b/a) + c$. Next, we consider the case $x^* = e^e$, and using the fact that $a > 0$, we obtain $U(x^*) \leq 2b \cdot \sqrt{e^e \ln\ln(e^e)} + c$.

It is again straightforward to check that the aforementioned bound is upper bounded from above by $7(b^2/a)\ln\ln(b/a) + c$.

We now consider the case $x^* > e^e$. By Lemma A.1, x^* is the unique point satisfying $\frac{\partial U(x^*)}{\partial x^*} = 0$, if it exists. The remainder of the proof is devoted to the final case where we obtain

$$0 = \frac{\partial U(x^*)}{\partial x^*} = -a + \frac{b(\frac{1}{\ln x^*} + \ln\ln x^*)}{\sqrt{x^* \ln\ln x^*}} \tag{A.2}$$

Continuing further, (A.2) implies

$$\frac{\sqrt{x^* \ln\ln x^*}}{\ln\ln x^* + \frac{1}{\ln x^*}} = \frac{b}{a} \triangleq \alpha. \tag{A.3}$$

Note

$$\frac{x^*}{\ln\ln x^*} > \alpha^2$$
$$\frac{x^*}{2\ln\ln x^*} < \alpha^2 \quad \text{since } \ln\ln x^* > \frac{1}{\ln x^*} \text{ for } x \geq e^e$$

It is easy to check that $x/\ln\ln x$ is a strictly increasing function for $x \geq e^e$ and $\lim_{x\to\infty}(x/\ln\ln x) = \infty$. (A.1) implies that there exist unique $x_{\min}$ and $x_{\max}$ satisfying

$$\frac{x_{\min}}{\ln\ln x_{\min}} = \alpha^2 \qquad \frac{x_{\max}}{2\ln\ln x_{\max}} = \alpha^2$$

The monotonicity of $x/\ln\ln x$ implies $x_{\min} \leq x^* \leq x_{\max}$. In order to complete the proof of the proposition, we will first state and prove Lemmas A.4 and A.5.

Lemma A.4 *$x_{\min} \geq \alpha^2 \ln\ln\alpha$ and $x_{\max} \leq 4\alpha^2 \ln\ln\alpha$.*

Proof. Let $B_1 = \alpha^2 \ln\ln\alpha$. Then

$$\begin{aligned} \frac{B_1}{\ln\ln B_1} &= \frac{\alpha^2 \ln\ln\alpha}{\ln\ln(\alpha^2 \ln\ln\alpha)} \\ &< \frac{\alpha^2 \ln\ln\alpha}{\ln\ln\alpha} \quad \text{since } \ln\ln\alpha \geq 1 \text{ for } \alpha \geq e^{2e} \end{aligned}$$

$$= \alpha^2.$$

Thus since $\frac{x}{\ln\ln x}$ is increasing for $x \geq e^e$, we have $x_{\min} \geq B_1$ and the first assertion is established.

Let $B_2 = 4\alpha^2 \ln\ln\alpha$. Then

$$\begin{aligned}
\frac{B_2}{2\ln\ln B_2} &= \frac{4\alpha^2 \ln\ln\alpha}{2\ln\ln(4\alpha^2 \ln\ln\alpha)} \\
&= \frac{4\alpha^2 \ln\ln\alpha}{2\ln(2\ln\alpha + \ln^{(3)}\alpha + \ln 4)} \\
&\geq \frac{4\alpha^2 \ln\ln\alpha}{2\ln(4\ln\alpha)} \quad \text{since } \ln\alpha \geq \ln^{(3)}\alpha \text{ and } \ln\alpha > \ln 4 \\
&\geq \frac{4\alpha^2 \ln\ln\alpha}{4\ln(\ln\alpha)} \quad \text{since } \ln\alpha \geq 2e > 4. \\
&= \alpha^2.
\end{aligned}$$

Thus, again since $x/\ln\ln x$ is increasing for $x \geq e^e$, then the second assertion follows. □

Lemma A.4 and $x_{\min} \leq x^* \leq x_{\max}$ imply

$$\alpha^2 \ln\ln\alpha \leq x^* \leq 4\alpha^2 \ln\ln\alpha. \tag{A.4}$$

Lemma A.5 $\sqrt{x_{\max} \ln\ln x_{\max}} \leq 4\alpha \ln\ln\alpha$.

Proof.

$$\begin{aligned}
\sqrt{x_{\max} \ln\ln x_{\max}} &\leq \sqrt{\left(4\alpha^2 \ln\ln\alpha\right) \ln\ln\left(4\alpha^2 \ln\ln\alpha\right)} \text{ by Lemma A.4} \\
&= \alpha\sqrt{4\ln\ln\alpha}\sqrt{\ln\left(2\ln\alpha + \ln^{(3)}\alpha + \ln 4\right)} \\
&\leq \alpha\sqrt{4\ln\ln\alpha}\sqrt{\ln\left(4\ln\alpha\right)} \text{ since } \ln\alpha \geq \ln^{(3)}\alpha \text{ and } \ln\alpha \geq \ln(e^{2e}) > \ln 4 \\
&\leq \alpha\sqrt{4\ln\ln\alpha}\sqrt{2\ln\ln\alpha} \text{ since } \ln\alpha > 4
\end{aligned}$$

and the lemma follows from the last step. □

We now complete the proof of Proposition A.3. We have

$$\begin{aligned}
U(x^*) &\leq -ax^* + 2b\sqrt{x^* \ln\ln x^*} + c \\
&\leq -ax_{\min} + 2b\sqrt{x_{\max} \ln\ln x_{\max}} + c \quad \text{since } x_{\min} \leq x^* \leq x_{\max} \\
&\leq -ax_{\min} + 8b\alpha \ln\ln\alpha + c \quad \text{by Lemma A.5} \\
&\leq -a\alpha^2 \ln\ln\alpha + 8b\alpha \ln\ln\alpha + c \quad \text{by Lemma A.4} \\
&= 7(b^2/a) \ln\ln(b/a) + c.
\end{aligned}$$

□

Bibliography

Aksin, Z., M. Armony, and V. Mehrotra (2007). The modern call-center: A multi-disciplinary perspective on operations management research. *Production and Operations Management, Special Issue on Service Operations in honor of John Buzacott (ed. G. Shanthikumar and D. Yao) 16*(6), 655–688.

Andrews, M., B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu (1996). Universal stability results for greedy contention-resolution protocols. *Proc. 27th IEEE Symposium on Foundations of Computer Science*, 380–389.

Bellman, R. (1957). *Dynamic Programming.* Princeton, University Press.

Ben-Tal, A., L. E. Ghaoui, and A. Nemirovski (2009). *Robust Optimization.* Princeton University Press.

Ben-Tal, A., B. Golany, A. Nemirovski, and J.-P. Vial (2005). Retailer-supplier flexible commitments contracts: A robust optimization approach. *MSOM 7*, 248–271.

Ben-Tal, A. and A. Nemirovski (1998). Robust convex optimization. *Math. Oper. Res. 23*, 769–805.

Ben-Tal, A. and A. Nemirovski (1999). Robust solutions of uncertain linear programs. *Oper. Res. Lett. 25*, 1–13.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control, Vols. I and II.* Athena Scientific.

Bertsimas, D. (1990). An analytic approach to a general class of G/G/s queueing systems. *Operations Research 38*(1), 139–55.

Bertsimas, D. and D. Brown (2009). Constructing uncertainty sets for robust linear optimization. *Operations Research 57*(6), 1483–95.

Bertsimas, D., D. Brown, and C. Caramanis (2011). Theory and applications of robust optimization. *to appear in SIAM Review*.

Bertsimas, D., D. Gamarnik, and J. Tsitsiklis (1996). Stability conditions for multiclass fluid queueing networks. *IEEE Trans. Automat. Control 41*, 1618–1631.

Bertsimas, D., D. Gamarnik, and J. Tsitsiklis (2001). Performance of multiclass Markovian queueing networks via piecewise linear Lyapunov functions. *Ann. of Appl. Prob. 11*(4), 1384–1428.

Bertsimas, D. and J. Nino-Mora (1999). Optimization of multiclass queueing networks with changeover times via the achievable region method: Part II, the multi-station case. *Math. Oper. Res. 24*, 331–361.

Bertsimas, D., I. Paschalidis, and J. Tsitsiklis (1994). Optimization of multiclass queueing networks: Polyhedral and nonlinear characterization of achievable performance. *The Annals of Applied Probability 4*, 43–75.

Bertsimas, D. and M. Sim (2004). The price of robustness. *Oper. Res. 52*, 35–53.

Bertsimas, D. and A. Thiele (2006). A robust optimization approach to inventory theory. *Oper. Res. 54*(1), 150–168.

Bertsimas, D. and J. Tsitsiklis (1993). Simulated annealing. *Statistical Science 8*(1), 10–15.

Bienstock, D. and N. Özbay (2008). Computing robust basestock levels. *Discrete Optimization 5*(2), 389–414.

Borodin, A., J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson (2001). Adversarial queueing theory. *Journal of ACM 48*, 13–38.

Chen, H. and D. Yao (2001). *Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization*. Springer-Verlag.

Chen, X., M. Sim, and P. Sun (2007). A robust optimization perspective on stochastic programming. *Oper. Res. 55*(6), 1058–1071.

Chung, K. (2001). *A Course in Probability Theory* (Third ed.). Academic Press.

Clark, A. and H. Scarf (1960, November). Optimal policies for a multi-echelon inventory problem. *Management Sci. 6*, 475–490.

Cruz, R. (1991a). A calculus for network delay, part I: network elements in isolation. *IEEE Trans. Information Theory 37*(1), 114–131.

Cruz, R. (1991b). A calculus for network delay, part II: network analysis. *IEEE Trans. Information Theory 37*(1), 132–141.

Dai, J. G. (1995). On the positive Harris recurrence for multiclass queueing networks: A unified approach via fluid models. *Ann. Appl. Probab. 5*, 49–77.

Dai, J. G. and T. G. Kurtz (1995). A multiclass station with Markovian feedback in heavy traffic. *Math. Oper. Res. 20*(3), 721–742.

Dai, J. G. and S. P. Meyn (1995). Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transcation on Automatic Controls 40*, 1889–1904.

de Smit, J. (1983a). A numerical solution for the multi-server queue with hyper-exponential service times. *Operations Research Letters 2*, 217–224.

de Smit, J. (1983b). The queue GI/M/s with customers of different types or the queue of GI/H/s. *Advances in Applied Probability 15*, 392–419.

Erlang, A. (1948). *On the rational determination of the number of cicuits.* In E. Brockmeyer and H.L. Halstron and A. Jensen, editors, The Life and Works of A.K. Erlang. The Copenhagen Telephone Company, Copenhagen.

Federgruen, A. and P. Zipkin (1984). Computational issues in an infinite-horizon multiechelon inventory model. *Oper. Res. 32*(4), 818–836.

Fu, M. (1994). Sample path derivatives for (s,S) inventory systems. *Oper. Res. 42*, 351–364.

Gallager, G. and A. Parekh (1993). A generalized processor sharing approach to flow control in integrated services networks: the single node case. *IEEE/ACM Transactions on Networking 1*(3), 344–357.

Gallager, G. and A. Parekh (1994). A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking 2*(2), 137–150.

Gallego, G. and I. Moon (1993). The distribution free newsboy problem: Review and extensions. *J. Oper. Res. Soc. 44*, 825–834.

Gallego, G. and I. Moon (1994). Distribution free procedures for some inventory models. *J. Oper. Res. Soc. 45*, 651–658.

Gamarnik, D. (2000). Using fluid models to prove stability of adversarial queueing networks. *IEEE Transactions on Automatic Control. (*Conference version in *FOCS98.) 4*, 741–747.

Gamarnik, D. (2003). Stability of adaptive and non-adaptive packet routing policies in adversarial queueing networks. *SIAM Journal on Computing. (*Conference version in *STOC99.)*, 371–385.

Gamarnik, D. and A. Zeevi (2006). Validity of heavy traffic steady-state approximations in open queueing networks. *Ann. Appl. Prob. 16*(1), 56–90.

Ganesh, A., N. O'Connell, and D. Wischik (2004). *Big Queues.* Springer-Verlag, Lecture Notes in Mathematics, Vol. 1838.

Gans, N., G. Koole, and A. Mandelbaum (2003). Telephone call centers: Tutorial, review, and research prospects,. *Manufacturing & Service Operations Management 5*, 79–141.

Glasserman, P. and S. Tayur (1994). The stability of a capacitated, multiechelon production-inventory system under a base-stock policy. *Oper. Res. 42*(5), 913–925.

Glasserman, P. and S. Tayur (1995). Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. *Management Sci. 41*(2), 263–281.

Goel, A. (1999). Stability of networks and protocols in the adversarial queueing model for packet routing. *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms.*

Graves, S. and S. Willems (2000). Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management 2*(1), 68–83.

Halfin, S. and W. Whitt (1981). Heavy-traffic limits for queues with many exponential servers. *Oper. Res. 29*(3), 567–588.

Harrison, J. M. (1990). *Brownian Motion and Stochastic Flow Systems.* Krieger Publishing Company.

Harrison, J. M. (2000). Stochastic networks and activity analysis. *Ann. Appl. Probab. 10*, 75–103.

Huh, W. and G. Janakiraman (2008). A sample-path approach to the optimality of echelon order-up-to policies in serial inventory systems. *Oper. Res. Lett. 36*(5), 547–550.

Ingber, L. (1993). Simulated annealing: Practice versus theory. *Mathl. Comput. Modelling 18*(11), 29–57.

Jin, H., J. Ou, and P. R. Kumar (1997). The throughput of irreducible closed Markovian queueing networks: functional bounds, asymptotic loss, efficiency, and the Harrison-Wein conjectures. *Math. Oper. Res. 22*, 886–920.

Kasugai, H. and T. Kasegai (1961). Note on minimax regret ordering policy — static and dynamic solutions and a comparison to maximin policy. *J. Oper. Res. Soc. Japan 3*, 155–169.

Kiefer, J. and J. Wolfowitz (1955). On the theory of queues with many servers. *Transactions of the American Mathematical Society 78*, 1–18.

Kingman, J. F. C. (1970). Inequalities in the theory of queues. *Journal of the Royal Statistical Society. Series B (Methodological) 32*, 102–110.

Kleinrock, L. (1975). *Queueing Systems.* John Wiley and Sons, Inc.

Kumar, P. R. and J. Morrison (2004). New linear program performance bounds for queueing networks. *Journal of Optimization Theory and Applications*, 575–597.

Kumar, S. and P. R. Kumar (1994). Performance bounds for queueing networks and scheduling policies. *IEEE Transactions on Automatic Control 8*, 1600–1611.

Langenhoff, L. and W. Zijm (1990). An analytical theory of multi-echelon production/distribution systems. *Statistica Neerlandica 44*(3), 149–174.

Latouche, G. and V. Ramaswami (1987). *Introduction to Matrix Analytic Methods in Stochastic Modeling.* Society for Industrial Mathematics.

Loynes, R. (1962). The stability of a queue with non-independent inter-arrival and service times. *Proc. Camb. Phil. Soc. 58*, 497–520.

Meyn, S. P. and R. L. Tweedie (1993). *Markov Chains and Stochastic Stability.* Springer-Verlag. London, UK.

Muharremoglu, A. and J. Tsitsiklis (2008). A single-unit decomposition approach to multi-echelon inventory systems. *Oper. Res. 56*, 1089–1103.

Natarajan, K., D. Pachamanova, and M. Sim (2009). Constructing risk measures from uncertainty sets. *Operations Research 57*(5), 1129–1141.

Özbay, N. (2006). *Ph.D. Thesis.* Columbia University.

Pollaczek, F. (1961). Théorie analytique des problémes stochastiques relatifs á un groupe de lignes téléphoniques avec dispositif d'attente. *Gauthier, Paris.*

Reiman, M. I. (1984). Open queueing networks in heavy traffic. *Math. Oper. Res. 9*, 441–458.

Rong, Y., Z. Bulut, and L. Snyder. A scalable heuristic for base-stock levels in multiechelon distribution networks. *Available at SSRN: http://ssrn.com/abstract=1475469.*

Rosling, K. (1989). Optimal inventory policies for assembly systems under random demands. *Oper. Res. 37*, 565–579.

Roundy, R. and J. Muckstadt (2000). Heuristic computation of periodic-review base stock inventory policies. *Management Sci. 46*(1), 104–109.

Scarf, H. (1958). A min-max solution of an inventory problem. In K. Arrow, S. Karlin, and H. Scarf (Eds.), *Studies in the Mathematical Theory of Inventory and Production*, pp. 201–209. Stanford University Press, Stanford, CA.

Scarf, H. (1960). The optimality of (s,S) policies in the dynamic inventory problem. In K. Arrow, S. Karlin, and P. Suppes (Eds.), *Mathematical Methods in the Social Sciences.* Stanford University Press, Stanford, CA.

Sethi, S. and F. Cheng (1997). Optimality of (s,S) policies in inventory models with markovian demand. *Operations Research 45*(6), 931–939.

Shapiro, A., D. Dentcheva, and A. Ruszczynski (2009). *Lectures on Stochastic Programming: Modeling and Theory.* Society of Industrial and Applied Mathematics. Philadelphia.

Shwartz, A. and A. Weiss (1995). *Large deviations for performance analysis.* Chapman and Hall.

Sigman, K. (1990). The stability of open queueing networks, stochastic processes and their applications. *Stochastic Processes and their Applications 35*, 11–25.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res. 21*, 1154–1157.

Turner, J. (1986). New directions in communications, or which way to the information age? *IEEE Commun. Mag. 24*, 8–15.

Whitt, W. (1972, Sep.). Embedded renewal processes in the GI/G/s queue. *Journal of Applied Probability. 9*(3), 650–658.

Whitt, W. (2002). *Stochastic-Process Limits.* Springer.

Zipkin, P. (2000). *Foundations of Inventory Management.* McGraw-Hill Higher Education, Boston, MA.