Robust optimization in a queueing context

van der Heide, W.

Award date:
2016

Robust Optimization in a Queueing Context

W. van der Heide
September 2015 - February 2016
Publishing date: April 18, 2016

Eindhoven University of Technology
Department of Mathematics & Computer Science
Industrial and Applied Mathematics
Stochastics, Probability and Operations Research

Supervisor:
prof. dr. J. S. H. van Leeuwaarden

TU/e Technische Universiteit Eindhoven University of Technology

Abstract

In this thesis we consider the technique of Robust Optimization and apply it in a queueing context. We show how Robust Optimization accounts for uncertain model parameters with a so-called uncertainty set. We apply robust methods to the Capacity Assignment problem by Kleinrock and consider different forms of uncertainty. We characterize the corresponding optimization problems and compute the robust capacity assignment. Several examples and numerical simulations illustrate that a robust method can significantly reduce the probability of infeasibility and the probability of high average delay, without performing worse on average. We also consider multiserver $G/G/s$ queues under Halfin-Whitt scaling and take a worst case perspective. By characterizing interarrival and service times in uncertainty sets we derive an upper bound for the worst case queue length in this scaling regime.

List of Figures

List of Tables

Contents

1 Introduction

In this introductory chapter we explain the technique of Robust Optimization through several examples. How does it work and what are the strengths of this technique? In the following chapters we will apply Robust Optimization to various models in a queueing context. Chapter 2 and 3 discuss the capacity assignment problem. Chapter 4 considers multiserver queues under Halfin-Whitt scaling.

1.1 What is Robust Optimization?

One way of solving real-world problems is through mathematical modeling. The idea is to develop a model that is simple enough to be mathematically tractable, but realistic enough to have practical use. Often the focus is on optimization, finding the 'best' solution for a problem, like achieving the lowest possible costs or providing the highest service standards to customers. The hope is then that the model is accurate enough to be relevant in practice.

Unfortunately, models are not always satisfactory or sufficient descriptions of reality. They use simplifications, predictions and estimations. The model parameters are often based on historical data or expert opinions and those are inherently uncertain. A method to handle parameter uncertainty is Robust Optimization (RO). For an overview, we refer to Ben-Tal et al. (2009). RO tries to find a solution that is resilient (robust) against errors or uncertainty in the data. Instead of using point estimates, this method tries a whole range of possible parameter values, captured in an uncertainty set. This uncertainty set expresses how sure we are that we estimated the parameters correctly.

Instead of optimizing our model for the estimated value of the parameters, we optimize it for the worst possible value in the uncertainty set. Instead of hoping for the best, we prepare for the worst. The idea is that the robust solution performs well even if the model parameters are different from what was expected. It protects the user against risk by taking a more conservative approach. The following introductory example shows how this works.

1.1.1 Newsvendor problem

As an example of Robust Optimization we will consider the newsvendor problem, a well-known model for determining the optimal inventory of a perishable good. The newsvendor buys items and then sells them for a profit. The problem is that he does not know how much demand there will be each day. If he buys too many items, the amount he does not sell perishes and he has to throw it away. But if he buys too few items, he has to disappoint customers and misses out on a profit. The optimal number of items to buy is the amount that maximizes his profit.

Say the newsvendor buys items for €3 and sells them for €5. Then ordering a quantity Q items costs $3Q$ euro. He can never sell more items then he bought and no more then the demand, so with a demand D he sells exactly $\min\{Q, D\}$ items. The profit P is his income from selling minus the costs from buying:

$$P = 5\min\{Q, D\} - 3Q$$

However, the newsvendor does not know the demand D in advance. On a busy day he might sell 120 items and on another day he might only have a demand for 80 items. We model the demand with a uniform distribution: $D \sim \text{Unif}(80, 120)$. This means that each demand between 80 and 120 is equally likely. Now we can look at the expected profit, the average profit he would get per day in the long run if he ordered Q items per day.

$$\mathbb{E}[P] = \mathbb{E}[5\min\{Q, D\}] - 3Q$$

The optimal order quantity Q^* is the quantity that maximizes the expected profit. We skip the details and just state that

$$Q^* = 80 + (120 - 80) \cdot \frac{5-3}{5} = 80 + 16 = 96.$$

There is always a demand for 80 items, so the newsvendor should always buy at least 80 items. He wants to buy a little more, because if the demand is higher he misses €2 profit on each item. On the other hand, every unsold item is a loss of €3. The optimal order quantity of 96 is slightly lower than the average demand of 100.

Until now we have done nothing with Robust Optimization. We just found the optimal solution for the model. We had to deal with uncertainty in the demand, but we had the distribution and could just take the expectation. Robust Optimization deals with a different kind of uncertainty, namely uncertainty in the model parameters. What if the true minimum demand is not 80, but 70? Then ordering 96 items would be too much; we would have to throw away more of them than expected.

We can formalize the uncertainty in the parameters in a so-called uncertainty set. This set contains all possible demand distributions we are worried about. In Section 1.3 we discuss the construction of the uncertainty set in more depth. For now, let us allow a deviation of 10 for the minimum and maximum demand. Instead of knowing for sure that the minimum is 80, it could be anything between 70 and 90. The maximum demand is somewhere between 110 and 130. The set $\mathcal{U}_D$ of all possible demand distributions then becomes

$$\mathcal{U}_D = \left\{ D \sim \text{Unif}(a, b) \;\middle|\; a, b \in \mathbb{N}, 70 \leq a \leq 90, 110 \leq b \leq 130 \right\}$$

Now we need to pick an order quantity Q. After that, we assume the worst possible scenario in the uncertainty set happens. We want to pick Q such that the profit in this worst case is as high as possible. In other words, we want the best outcome for the most pessimistic scenario. The order is important here. We first make a decision for the order quantity and then we get to see the result. Anticipating on the actual demand distribution is not possible.

In this example it is straightforward to see that lower demand is always worse for the newsvendor. No matter what quantity he decides to order, the worst thing that could happen to him is being left with unsold items because the demand was too low. The lowest demand distribution in the uncertainty set is Unif$(70, 110)$, so the newsvendor should optimize his order quantity for that pessimistic scenario. That is,

$$Q^* = 70 + (110 - 70) \cdot \frac{5-3}{5} = 70 + 16 = 86,$$

so the robust solution to the newsvendor problem would be ordering 86 items. Robust Optimization considered the possibility that we overestimated the demand and that the actual demand could be 10 items lower. If that is the case, it is better to order 10 items less. The robust approach is conservative and protects the newsvendor from the risk of buying too many items if we overestimated the demand. In Table 1.1.1 we compare the classic and the robust approach for the predicted demand and the two most extreme demand distributions in the uncertainty set.

	classic	**robust**	**absolute difference**	**% difference**
lower demand	149.195	155.415	-6.220	-4.17%
predicted demand	175.415	169.439	5.976	3.41%
higher demand	189.439	172.000	17.439	9.21%

Table 1.1: Comparison of the profit for the classic versus robust newsvendor problem

The results are very typical for Robust Optimization. The robust solution is better in the pessimistic scenario with low demand. It provides a relatively stable expected profit of at least €155 per day. In the classic approach there is more variation in the outcomes. If the demand turns out to be what was expected in the model or even higher, the classic approach can take full advantage of it with a high expected profit. The robust approach is for conservative, risk averse newsvendors who want a stable income. The classic approach is for newsvendors who want maximum profits and can afford the risk of being wrong.

Finally, note that the uncertainty set plays a major role in Robust Optimization. Taking a larger uncertainty set would mean allowing even more pessimistic scenarios and being more conservative. On the other hand, taking a smaller uncertainty set would bring us closer to the classic approach. If we are extremely confident about the model parameters, the uncertainty set would contain just one value and the robust approach would be identical to the classic approach.

1.1.2 Relationship with other techniques

Robust Optimization is not the only technique that deals with uncertainty; the reader might be familiar with some of them. What makes Robust Optimization special?

- **Sensitivity analysis** is a way to calculate the sensitivity of a solution to uncertainty in the parameters. It tells you how big the effect of deviations is and which parameters have the biggest

effect. Sensitivity analysis is done after obtaining a solution and can signal that uncertainty needs to be taken into account. RO offers a way to deal with uncertainty by incorporating it into the model (through the parameters). Sensitivity analysis does the evaluation afterwards, whereas RO offers an integrated solution.

- **Stochastic programming** models uncertainty with probability distributions and then solves the corresponding linear optimization problem. It requires difficult to estimate stochastic distributions for the parameters, while RO needs less information about the uncertainty. Furthermore Stochastic programming is only computationally tractable for specific distributions. RO uses a more general setting and does not need as many assumptions about the uncertainty to remain computationally tractable.
- **Robust control** is a concept from engineering where the goal is to build a controller that works well under uncertain disturbances. An example is keeping your bike straight when the wind is blowing against it. The designer has little control over the disturbances, but he wants to keep them in check. That can be done by feedback loops (if the bike is tilting to the left, lean to the right). The goal is similar to RO, but the context is very different. Note that there also exists Adjustable Robust Optimization which reacts to changes in the parameters as well.
- **Bayesian statistics** is a field of statistics that looks at variables in terms of 'belief' and assumes variables have no 'true' value, but come from some distribution. We can use data to update our belief about that distribution. Uncertainty in parameters is the core of the Bayesian framework and it offers techniques to update that uncertainty when presented with data. Therefore Bayesian statistics may be a good candidate to construct uncertainty sets for RO.

Modeling uncertainty is not new, but Robust Optimization gives a unique perspective that takes uncertainty into account at the modeling stage while remaining computationally tractable.

1.2 When is Robust Optimization valuable?

In the introductory example we saw a few of the strengths and weaknesses of a robust approach compared with a classic approach. A fundamental difference is the philosophy and the starting assumptions. Robust Optimization makes sense if:

- Model parameters are uncertain
- This uncertainty can be quantified
- Small changes in model parameters would drastically affect the results

Queueing models already incorporate several sources of uncertainty. Arrival and service times are often uncertain and they require special restrictions or assumptions on the associated distributions to analyze them. Poisson arrival processes and exponential service times make life a lot easier, and there are several other 'easy' or 'tractable' distributions. These distributions are often chosen for mathematical convenience, even when it is obvious that it is not a realistic assumption. Empirical distribution functions on the other hand would be a lot more realistic, but they tend to be difficult to analyze and often require cumbersome numerical algorithms or stochastic simulation.

For tractability, the parameters like the mean arrival rate or mean service time are often assumed known. You could argue that the model already contains a lot of uncertainty, so we cannot handle uncertainty in the parameters as well. But in reality it is highly unlikely that we can accurately predict mean arrival rates. The reason to ignore uncertainty in the parameters is a lack of mathematical tools and not a lack of importance of uncertainty. However, uncertainty in the parameters should not be ignored, since queueing models show huge swings in performance for slightly different occupation rates. The next example illustrates that.

Example: sensitivity to arrival rate
Consider a single server queue with Poisson arrivals and exponential processing times (M/M/1 queue). The mean processing time is 1 minute and on average 0.95 customers arrive per minute. That means that the server is working 95% of the time. In this model the mean waiting time is $\mathbb{E}[W] = 0.95/(1 - 0.95) = 19$ minutes. But what if the mean arrival rate is actually 0.96? That would give a mean waiting time of $\mathbb{E}[W] = 0.96/(1 - 0.96) = 24$ minutes. An increase of around 1% in the arrival rate, from 0.95 to 0.96, leads to an increase in the mean waiting time from 19 to 24 minutes. That is more than a 26% increase! In this example, even a small error in predicting the arrival rate could have huge consequences. This is very typical behavior in many queueing models.

Robust Optimization can mitigate this model sensitivity to errors. It reduces the variance of performance measures like waiting time and decreases the probability of extreme outcomes. In queueing models extreme outcomes can have a drastic effect, because the system can become infeasible. If there are more arrivals than a server can handle, the queue will grow to infinity. Infeasibility represents the worst performance a queueing system can have. Due to the sensitivity this can happen even for very small errors in the parameters. Robust Optimization can be a valuable technique to reduce the probability of infeasibility.

Robust Optimization could also be used in adaptive systems that learn over time. When we learn more about the parameters, we become more certain about what their true values are. That can be captured in terms of a shrinking uncertainty region. As the size of the uncertainty set diminishes with time, the robust solution starts to resemble the classic solution more. In that robust methods could improve systems early in the learning process, while later on its effect becomes smaller.

Besides improving the performance of queueing systems, Robust Optimization can also be used to analyze them. Uncertainty in queueing theory is usually modeled with stochastic distributions, but a perspective with a worst case in an uncertainty set is also worth considering. The deterministic nature can make problems easier to analyze and it requires less knowledge about the underlying distributions. Bertsimas et al. (2011) discuss one example of performance analysis of queueing networks via Robust Optimization, in Chapter 4 of this thesis we show another example.

1.3 Uncertainty sets

A key concept of Robust Optimization is the uncertainty set. This set contains all scenarios the user wants to consider. A larger uncertainty set means considering more deviations from the expected results and therefore a more conservative approach. A common misconception is that the uncertainty set should contain every possible event that could happen. That is not true, it should only contain events you consider realistic and worth considering. When designing a chair, it should po-

tentially support a heavy person, so that event is in your uncertainty set. However, the event that an elephant steps on your chair is not something you care about, so those too extreme weights should not be in your uncertainty set.

Let us look at an example with claims arriving at an insurance company. One of the model parameters is the number of claims per month. The company kept record of the number of claims over the last four months. The amounts were 127, 90, 145 and 102. There are several ways to construct an uncertainty set with this data, each with its own advantages and disadvantages.

Average: Simply estimating the number of claims by the average. Here we get $(127+90+145+102)/4 = 116$ claims per month. We do not consider uncertainty and just take this average as the model parameter. This method is quick and widely used, but it does not represent reality very well. Over the past four months we saw significant deviations from the average of 116. This would be a naive classic approach.

Weighted scenarios: Take each month as a different scenario. Every scenario happened once in the past, so we could assign probability $1/4$ to each scenario. Denote this as a probability vector $p = (1/4, 1/4, 1/4, 1/4)$. However, we are not certain that every scenario is equally likely. Maybe 127 claims happens more often than 90 claims. Then we should give a higher weight to the first scenario, for example $p = (1/3, 1/6, 1/4, 1/4)$. If we are uncertain about the likelihood of each scenario, we could take every probability vector 'close to' p in the uncertainty set. We can specify 'close to' with some measure of distance between two probability vectors; one example is the χ^2-distance. This is called distributional robust optimization. The corresponding uncertainty set is

$$\mathcal{U} = \left\{ q \in \mathbb{R}^4 : \quad \sum_{i=1}^{4} q_i = 1, \quad \sum_{i=1}^{4} \frac{(p_i - q_i)^2}{p_i} \leq \alpha \right\}$$

where α determines the size of the uncertainty set. A larger α corresponds to more uncertainty in p and more conservativeness. The disadvantage here is in the modeling assumptions. Are there really only four scenarios that could happen? No, probably any number of claims between 90 and 145 could occur. Those scenarios were only discarded because they never happened in the past, but that means discarding a lot of realistic scenarios.

Confidence interval: We can see the data from the past four months as four realizations of some probability distribution. Under the assumption that this underlying probability distribution is normal, we can construct a confidence interval for the expectation. The formula for the 95%-confidence interval with 4 data points is

$$\bar{x} - t_{3;0.025} \cdot \frac{s}{\sqrt{4}} < \mu < \bar{x} + t_{3;0.025} \cdot \frac{s}{\sqrt{4}}$$

where $\bar{x}$ is the average of 116. $t_{3;0.025}$ is a quantile of the Student t-distribution and if we look it up in a table we find 3.182. Finally, s is an estimator for the standard deviation, which turns out to be 24.73 when we calculate it. Together this gives the 95%-confidence interval $76 < \mu < 156$. The interpretation is that when we make these intervals, 95% of the time the true expectation is within it. The corresponding uncertainty set for the number of claims n would be

$$\mathcal{U} = \left\{ n \in \mathbb{Z} : 76 < n < 156 \right\}$$

and the size of the uncertainty set can be adapted by taking a different confidence level than 95%. Using statistics to quantify uncertainty looks solid, but it should not be seen as a definitive answer. The underlying probability distribution for the number of claims is unknown, so the assumption that it is normal is hard to justify. Furthermore, the confidence interval can be very wide if there is not much data. The whole point of Robust Optimization is that it is used when the parameters are unknown and not enough data is available, but confidence intervals require assumptions on the underlying distribution and a lot of data.

Credible interval: In Bayesian statistics, credible intervals are a concept analogous to confidence intervals in frequentist statistics. In a Bayesian view of the world, one would already have beliefs about the number of claims even before seeing the data. Maybe it is information from an expert or ideas such as "there are probably a few hundred claims at this company each month". This prior belief is expressed in a prior probability distribution. Then one reviews the data and uses Bayesian techniques to update their beliefs in a posterior probability distribution. A 95% credible interval contains 95% of the probability mass in the posterior distribution. You believe that the model parameter is in the credible interval 95% of the time.

The advantage of this approach is that it is close to the Robust Optimization view. It sees model parameters as a random variable and we can use data to adjust our beliefs about that random variable. Searching for the one true value of a parameter is pointless if that parameter could take a range of values. The disadvantage is that Bayesian statistics is not widely known and accepted. Furthermore, the problem shifts from "How to formulate an uncertainty set?" to "How to determine my prior distribution?". The prior is often based on opinions and feelings, which can be seen as an advantage (incorporating all information) or a disadvantage (subjectivity).

1.4 Reading guide

In this introductory chapter we introduced the method of Robust Optimization and key concepts like the uncertainty set. The following chapters apply these techniques to different contexts or examples from queueing. Chapter 2 and 3 consider a robust approach to the Capacity Assignment problem. Chapter 2 introduces the problem, shows how to make it robust and how to solve the corresponding robust optimization problem. Chapter 3 uses numerical results and examples to show the value of a robust approach to the Capacity Assignment problem. Chapter 4 can be read separately from the previous chapters. It considers multiserver queues under Halfin-Whitt scaling from a worst case perspective. Chapter 5 is the conclusion which summarizes the thesis and poses new questions about how to apply Robust Optimization in a queueing context.

2
Capacity Assignment Problem

In this chapter we consider the Capacity Assignment (CA) problem. This problem has been solved in 1976 by Kleinrock for given parameters. But what happens if we are uncertain about the value of the parameters? We introduce a robust counterpart to the CA problem and show how to derive the robust solution.

2.1 Model

Consider a computer-communication network with M channels to send messages over. These messages have lengths drawn independently from an exponential distribution with mean $1/\mu$ (bits). Each channel experiences Poisson distributed traffic with rate λ_i and has a capacity C_i (bits per second) to process them. Part of the traffic is internal (messages passed from other channels) and the rest is external (new messages). The total external traffic entering the network is denoted by γ. We are interested in the average message delay T.

This problem was introduced and solved by Kleinrock (1976). The problem was later extended by Wein (1989) to general interarrival and service distributions. Kleinrock states that, when the M channels can be considered independent, the average message delay is given by

$$T = \sum_{i=1}^{M} \frac{\lambda_i}{\gamma} \left[\frac{1}{\mu C_i - \lambda_i} \right] \tag{2.1}$$

We continue with Kleinrock's description and solution of the Capacity Assignment (CA) problem:

Given:	Flows λ_i and network topology
Minimize:	T
With respect to:	C_i
Under constraint:	$D = \sum_{i=1}^{M} d_i C_i$

The CA problem can be written as

$$\min_{C_i} \sum_{i=1}^{M} \frac{\lambda_i}{\gamma} \left[\frac{1}{\mu C_i - \lambda_i} \right]. \tag{2.2}$$

So we want to pick the capacities C_i in order to minimize T. However, each channel has a linear cost d_i for its capacity, and there is only a total budget D. How should the capacities be divided? The first step is forming the Lagrangian

$$G = T + \beta \left[\sum_{i=1}^{M} d_i C_i - D \right], \tag{2.3}$$

where the positive parameter β will be determined later. Note that minimizing G with respect to the capacities is the same as minimizing T, since the budget constraint makes the term in brackets identically equal to zero. To minimize G we must satisfy the following set of M equations:

$$\frac{\partial G}{\partial C_i} = 0, \qquad i = 1, 2, ..., M.$$

Using equation (2.1) gives

$$-\frac{\lambda_i \mu}{\gamma} \frac{1}{(\mu C_i - \lambda_i)^2} + \beta d_i = 0$$

which can be written as

$$(\mu C_i - \lambda_i)^2 = \frac{\lambda_i \mu}{\beta \gamma d_i}$$

or

$$C_i = \frac{\lambda_i}{\mu} + \sqrt{\frac{\lambda_i \mu}{\beta \gamma d_i}} \cdot \frac{1}{\mu} = \frac{\lambda_i}{\mu} + \frac{1}{\sqrt{\beta \gamma \mu}} \sqrt{\frac{\lambda_i}{d_i}} \qquad i = 1, 2, ..., M. \tag{2.4}$$

We proceed by multiplying with d_i and summing over i:

$$\sum_{i=1}^{M} C_i d_i = \sum_{i=1}^{M} \frac{\lambda_i d_i}{\mu} + \frac{1}{\sqrt{\beta \gamma \mu}} \sum_{i=1}^{M} \sqrt{\lambda_i d_i}.$$

The left-hand side is simply D. Now we solve for the constant to eliminate β:

$$\frac{1}{\sqrt{\beta \gamma \mu}} = \frac{D - \sum_{i=1}^{M} \frac{\lambda_i d_i}{\mu}}{\sum_{i=1}^{M} \sqrt{\lambda_i d_i}} = \frac{D_e}{\sum_{i=1}^{M} \sqrt{\lambda_i d_i}}$$

where the "excess dollars" D_e are defined by

$$D_e \triangleq D - \sum_{i=1}^{M} \frac{\lambda_i d_i}{\mu}. \tag{2.5}$$

Finally, by substituting everything into equation (2.4) we arrive at the optimal solution for the CA problem

$$C_i = \frac{\lambda_i}{\mu} + \left(\frac{D_e}{d_i}\right)\frac{\sqrt{\lambda_i d_i}}{\sum_{j=1}^{M}\sqrt{\lambda_j d_j}} \qquad i = 1, 2, ..., M. \tag{2.6}$$

This solution can be understood as follows. The assigned capacities are at least λ_i/μ, which is the minimum required capacity for handling all messages. The remaining budget, given by the excess dollars D_e, is used to improve the performance of the network. After scaling by the cost rate d_i, each channel gets additional capacity in proportion to the square root of the cost-weighted load $\lambda_i d_i$. Low cost, high load channels receive the lion share of the additional capacity.

2.2 Robust approach

Let us now consider the robust counterpart of the CA problem (RCA). We no longer assume that the λ_i are known. This uncertainty is captured in an uncertainty set $\mathcal{U}$. The modified CA problem becomes:

Given:	Flows $\lambda_i \in \mathcal{U}$ and network topology
Minimize:	T for the worst $\lambda_i \in \mathcal{U}$
With respect to:	C_i
Under constraint:	$D = \sum_{i=1}^{M} d_i C_i$

In view of equation (2.2) the objective function thus becomes

$$\min_{C_i} \max_{\lambda_i \in \mathcal{U}} \sum_{i=1}^{M} \frac{\lambda_i}{\gamma}\left[\frac{1}{\mu C_i - \lambda_i}\right]. \tag{2.7}$$

In other words, we want to find the capacities that minimize the average delay for the worst choice of traffic that is still in the uncertainty set. The order is important here: first we pick capacities and then the worst traffic is determined. In general, this is a difficult optimization problem, since the worst traffic could be different depending on the capacities chosen. There are potentially infinitely many scenarios in the uncertainty set $\mathcal{U}$ to optimize over. In the literature, this is often circumvented by using the dual optimization problem, see for example Bertsimas & Sim (2004). However, we will show in a different way that the RCA problem is numerically tractable, because there are only a finite number of relevant scenarios to consider.

Now we need to define uncertainty sets. We consider two types of uncertainty: uncertainty about the total volume of traffic in the network (overall higher or lower traffic) and uncertainty about the distribution of the arrivals over the different channels (uncertain routing). Together they cover all forms of uncertainty in the arrival parameters, but we separate them because they have opposite effects on the robust solution. More uncertainty about the total volume of arrivals leads to assigning more capacity to the most used channels, while more uncertainty about the routing of arrivals leads to taking away capacity from the most used channels. It is not always straightforward which form of 'uncertainty' needs to be considered and what the corresponding uncertainty set should look like.

2.2.1 Overall higher or lower traffic

Let us consider the case where the traffic at all channels could be higher or lower than expected. We ignore the fact that the channels are connected and look at them independently. The channel i could have a mean arrival rate $\lambda_i + \alpha_i$ instead of the expected mean λ_i. The corresponding uncertainty set is

$$\mathcal{U} = \left\{ \tilde{\lambda} \in \mathbb{R}^M : \tilde{\lambda_i} > 0, |\tilde{\lambda_i} - \lambda_i| \leq \alpha_i \text{ for } i = 1, ..., M \right\}. \tag{2.8}$$

For each channel there is a separate inequality for the maximum mean arrival rate. The worst case is independent of the capacities chosen, because a higher arrival rate always leads to a longer delay. Therefore the worst values of the parameters are $\lambda_i^* = \lambda_i + \alpha_i$. The objective function (2.7) then becomes

$$\min_{C_i} \sum_{i=1}^{M} \frac{\lambda_i}{\gamma} \left[\frac{1}{\mu C_i - \lambda_i^*} \right]. \tag{2.9}$$

which is just the original objective function (2.2) with λ_i replaced by λ_i^*. That means that the original derivation can be followed to get the robust solution

$$C_i = \frac{\lambda_i^*}{\mu} + \left(\frac{D_e^*}{d_i} \right) \frac{\sqrt{\lambda_i^* d_i}}{\sum_{j=1}^{M} \sqrt{\lambda_j^* d_j}} \qquad i = 1, 2, ..., M. \tag{2.10}$$

This solution is only valid if the amount of excess dollars D_e^* is positive. $D_e^* \leq 0$ means that there is not enough budget to process all messages. Then the queue lengths would grow to infinity and we would call the model infeasible. Therefore the available budget limits how much uncertainty can be handled. This restricts the values of the α_i. That brings us to a key difficulty when applying Robust Optimization to queueing models: "*The size of the uncertainty set must be chosen such that the system is still feasible, even in the worst case scenario.*" An infeasible robust solution can be taken as a warning that the available resources are not sufficient to cover all the desired risks. Picking a smaller uncertainty set is one solution, but in practice raising the budget could be more desirable.

Provided that the budget is sufficient to handle the worst case scenario, what can we learn from the robust solution? Compared to the classic solution, every channel gets a higher base capacity of λ_i/μ. Channels with high uncertainty α_i profit the most from this. Since overall traffic is higher in the robust approach, there is less excess capacity to divide. The additional capacity is still assigned based on the square root of the cost-weighted load $\lambda_i^* d_i$. We can illustrate the difference between robust and classic with two competing effects.

If there is one channel with a relatively high α_i (a lot of uncertainty about the mean arrival rate), the robust approach assigns more capacity to that channel 'just to be sure'. If your main concern is that one particular channel gets way more traffic than expected, the robust solution tells you how to compensate for that. Another concern is that the overall load on the entire system is higher than expected. For example, $\alpha_i = \lambda_i \cdot 0.05$ indicates 5% more traffic everywhere. This gives $\lambda_i^* = 1.05\lambda_i$ and robust capacities

$$C_i = \frac{1.05\lambda_i}{\mu} + \left(\frac{D_e^*}{d_i} \right) \frac{\sqrt{1.05\lambda_i d_i}}{\sum_{j=1}^{M} \sqrt{1.05\lambda_j d_j}} = \frac{1.05\lambda_i}{\mu} + \left(\frac{D_e^*}{d_i} \right) \frac{\sqrt{\lambda_i d_i}}{\sum_{j=1}^{M} \sqrt{\lambda_j d_j}} \qquad i = 1, 2, ..., M.$$

(2.11)

We see that the additional capacity is divided over the channels by the same ratio as the classic approach in equation (2.6). The difference is in the first term, $1.05\lambda_i/\mu$. This term gives 5% more base capacity to every channel, so higher load channels end up with a bigger share of the total capacity than in the classic approach. This effect is stronger if there are less excess dollars D_e^*. We can summarize that the robust approach assigns more capacity to channels with a relatively high uncertainty and channels with a high mean arrival rate. This capacity is taken away from channels with less uncertain arrival rates and channels with a low mean arrival rate. In Section 3.2 numerical examples are given.

2.2.2 Uncertain routing

Another model is based on the connection between the channels in a network. The channels are used to send messages from A to B, but we are not certain what the routing from A to B is. Kleinrock assumes a unique path through the network for a given origin-destination pair, but on page 315 he notes that in some practical cases dropping the fixed routing procedure may introduce a useful degree of freedom. For this model we assume that the total amount of traffic can be estimated accurately, but the distribution over the channels is uncertain.

One uncertainty set that captures these modeling assumptions could be:

$$\mathcal{U} = \left\{ \tilde{\lambda} \in \mathbb{R}^M : \quad \tilde{\lambda} \geq 0, \quad \sum_{i=1}^{M} \tilde{\lambda}_i = \sum_{i=1}^{M} \lambda_i, \quad \sum_{i=1}^{M} |\tilde{\lambda}_i - \lambda_i| \leq \alpha \right\}. \tag{2.12}$$

The condition $\sum_{i=1}^{M} \tilde{\lambda}_i = \sum_{i=1}^{M} \lambda_i$ fixes the total arrival rate. The parameter α determines the size of the uncertainty set, it can be seen as the rate of messages with a different routing than expected. The model behind this assumes that there can be small errors in the estimated arrival rates. At a small rate, say ϵ, messages predicted to go to one channel use a different channel. There are N of these errors. Here are some reasons for these errors and ways to determine the corresponding ϵ and N:

- **Prediction errors.** Routing is based on message types. A survey is used to estimate the proportion of each message type. ϵ is the error margin on these estimates and can be calculated for example with a 95%-confidence interval. Each of the channels has an error margin, so there are a total of $N = M$ possible errors.
- **Hashing collision.** Routing uses hash functions to map messages to destinations (see for example L. Saino et al. (2013)). Emphasis for the mapping is on speed, so some 'wrong' mappings are allowed. $\epsilon =$ [the probability that an arrival gets a specific wrong mapping] $\cdot$ [the total arrival rate] and N is the number of wrong mappings that can occur.
- **Future changes.** Even if the current estimate is correct, the routing algorithm or the mix of messages may change in the future. We want a capacity assignment that is robust against these changes. ϵ is the rate of affected arrivals and N is the number of future changes expected.

There are other applications with their own uncertainty, like the road network on page 36. Each error shifts a part of the arrival rate from a random channel to another random channel. Some of these errors might cancel each other out. A shift from channel j to channel k cancels out a shift from channel k to channel j. That means that the total deviation from the estimated arrivals $\sum_{i=1}^{M} |\tilde{\lambda}_i - \lambda_i|$ is probably less than N times ϵ. The parameter α is chosen such that 95% of the time we have $\sum_{i=1}^{M} |\tilde{\lambda}_i - \lambda_i| \leq \alpha$.

If we have an estimate for the size of errors ϵ and the number of errors N, we can simulate hypothetically true arrival rates with N random shifts of size ϵ. Then we pick α such that 95% of the time these hypothetically true arrival rates are in the uncertainty set. In Section 3.1.2 the algorithm is explained in more detail. The important part is that the total traffic stays the same. If one channel gets more messages than expected, other channels get fewer messages to compensate. If one channel experiences a pessimistic scenario with more arrivals than expected, other channels have an optimistic scenario with less messages. This dependence between channels makes it difficult to pick one overall worst case scenario. In fact, the worst case depends on the capacities chosen. On page 15 we see an example of this dependence.

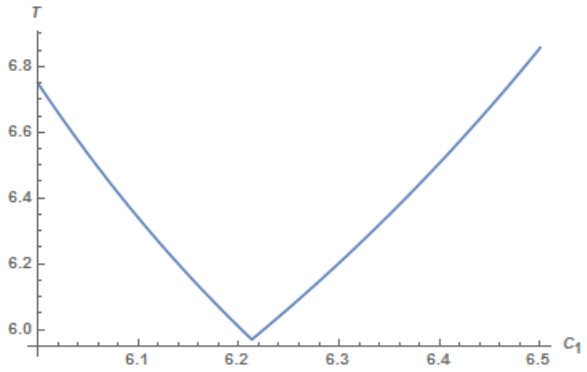

Figure 2.1: Worst average delay T for various capacity assignments $C = (C_1, 17 - C_1)$.

Example: worst case depends on capacities chosen
Consider the RCA problem with two channels. The mean arrival rates are estimated to be 4 and 8 messages per minute. Both channels have the same cost and the budget allows capacity for 17 messages per minute. We are uncertain about the routing of $\alpha = 1$ message per minute. That means that the first channel gets between 3 and 5 messages per minute and the second channel between 7 and 9 messages per minute. The total arrival rate of messages should always add up to 12, so $\lambda_1^* + \lambda_2^* = 12$. Also, the capacities should add up to the budget of 17, so $C_2 = 17 - C_1$. The objective function can be written as

$$\min_{C_1,C_2} \left[\frac{\lambda_1^*}{C_1 - \lambda_1^*} + \frac{\lambda_2^*}{C_2 - \lambda_2^*} \right] = \min_{C_1} \left[\frac{\lambda_1^*}{C_1 - \lambda_1^*} + \frac{12 - \lambda_1^*}{(17 - C_1) - (12 - \lambda_1^*)} \right].$$

under the constraints

$$3 \leq \lambda_1^* \leq 5 \quad \text{and} \quad 5 < C_1 < 8,$$

because C_1 needs to be at least 5 for the first channel and at most 8 so there is 9 capacity left for the second channel.

If we pick the capacities $C = (6, 11)$, the maximum average delay $T^* = 6.75$ is realized for arrival rates $\lambda_1^* = (5, 7)$. Apparently in the worst case the first channel is the bottleneck with an arrival rate close to the capacity. Maybe we should increase the capacity for the first channel? If we pick the capacities $C = (7, 10)$, the maximum average delay $T^* = 9.75$ is realized for arrival rates $\lambda_1^* = (3, 9)$. Now the worst case has changed: the second channel became the bottleneck. Apparently the worst case depends on the capacities chosen. Since capacity can only be assigned once, giving more capacity to one channel takes away capacity from another channel, creating new problems there.
In Figure 2.1 we see the worst case average delay for various values of C_1. The minimum is at $C_1 \approx 6.21$. For values below 6.21 the first channel is the bottleneck and the worst case is $\lambda_1^* = (5, 7)$. For values above 6.21 the second channel is the bottleneck and the worst case is $\lambda_1^* = (3, 9)$. The minimum average delay is reached when both extreme worst cases are equally bad. The robust solution tells us we should pick capacities $C = (6.21, 10.79)$. Then the average delay is below 6 minutes for every scenario in the uncertainty set.

2.3 Structure of the worst case

We have seen that with uncertain routing each choice of the capacities can have a different worst case. That means that we potentially have an infinite number of worst cases to optimize over. However, we will show that the optimization problem is tractable, because there is only a finite number of distinct worst cases that cover all possible capacity choices. In this section we start with fixed capacity choices C_i and we will characterize the corresponding worst case arrival rates λ^*. Remember that the worst case is the set of arrival rates that maximizes the average delay for our choice of capacities.

First we need the assumption that the capacities C_i can handle every arrival rate in the uncertainty set: For any $\tilde{\lambda} \in \mathcal{U}$ we have $\mu C_i > \tilde{\lambda}_i$ for all $i = 1, ..., M$. Capacities without this property are not worth analyzing, because they would allow infeasibility within the uncertainty set. Infeasibility leads to infinite delay, so any capacities that do not cause infeasibility are always better solutions

to the robust optimization problem. If all capacity assignments cause infeasibility within the uncertainty set, there is too much uncertainty in the system to make it feasible reliably. In that case one should reduce the uncertainty in the model parameters before trying any kind of optimization.

Now we introduce a lemma with the average delay T as a function of the arrival rate at channel j, say x. Since we want the total arrival rate to remain constant, select one other channel $k \neq j$ with arrival rate $\Lambda - x$, where Λ is the combined arrival rate for channel j and k. Fix the arrival rates at all other channels, say λ_i^{FIX}. Using equation (2.1) we get the average delay function

$$T(x) = \frac{x}{\gamma(\mu C_j - x)} + \frac{\Lambda - x}{\gamma\big(\mu C_k - (\Lambda - x)\big)} + \sum_{\substack{i=1 \\ i \neq j,k}}^{M} \frac{\lambda_i^{\text{FIX}}}{\gamma}\left[\frac{1}{\mu C_i - \lambda_i^{\text{FIX}}}\right] \tag{2.13}$$

Lemma 2.3.1. *The function $T(x)$ is strictly convex for $x \in (\Lambda - \mu C_k, \mu C_j)$.*

Proof. Since the capacity needs to exceed the total arrival rate, we have

$$C_j + C_k > \frac{x}{\mu} + \frac{\Lambda - x}{\mu} = \frac{\Lambda}{\mu},$$

so the interval $(\Lambda - \mu C_k, \mu C_j)$ is non-empty. The first derivative is

$$\begin{aligned} T'(x) &= \frac{1}{\gamma}\left(\frac{1}{\mu C_j - x} + \frac{x}{(\mu C_j - x)^2} - \frac{1}{\mu C_k - (\Lambda - x)} - \frac{\Lambda - x}{(\mu C_k - (\Lambda - x))^2}\right) \\ &= \frac{1}{\gamma}\left(\frac{\mu C_j}{(\mu C_j - x)^2} - \frac{\mu C_k}{(\mu C_k - (\Lambda - x))^2}\right) \end{aligned}$$

and the second derivative is

$$T''(x) = \frac{1}{\gamma}\left(\frac{2\mu C_j}{(\mu C_j - x)^3} + \frac{2\mu C_k}{(\mu C_k - (\Lambda - x))^3}\right).$$

On the specified interval the denominators are both strictly positive. We have $T''(x) > 0$ so $T(x)$ is strictly convex. □

Now we consider the worst case arrival rates λ^* for the capacities C, so the arrival rates that maximize T. As a reference point we use the estimated/observed arrival rates λ. These are the values the classic, nonrobust method uses. It turns out that λ^* and λ only differ for two channels.

Lemma 2.3.2. *Among the worst case arrival rates λ^* there is at most one channel j such that $\lambda_j^* > \lambda_j$. There is also at most one channel l such that $\lambda_l^* < \lambda_l$.*

Proof. We use contradiction. Suppose there are multiple channels such that $\lambda_i^* > \lambda_i$. Pick two of these channels, say j and k with $j \neq k$. We will define two different sets of arrival rates in the uncertainty set, λ^L and λ^H. We show that at least one of these two sets gives a higher average delay than λ^*, which is a contradiction with the fact that λ^* is the worst case. Therefore there cannot be multiple channels such that $\lambda_i^* > \lambda_i$.

Define

$$\lambda_i^L = \begin{cases} \lambda_j, & \text{for } i = j, \\ \lambda_k^* + \lambda_j^* - \lambda_j, & \text{for } i = k, \\ \lambda_i^*, & \text{otherwise}, \end{cases}$$

This can be interpreted, starting from λ^*, as lowering the arrival rate at channel j to the predicted arrival rate λ_j. To compensate, the arrival rate at channel k is increased by the same amount. Now we show that $\lambda^L \in \mathcal{U}$ by showing that this vector satisfies all the requirements in expression (2.12). We will use the fact that $\lambda^* \in \mathcal{U}$, because the worst case has to be in the uncertainty set. The requirement

$$\lambda^L \geq 0$$

is satisfied, since $\lambda^* \geq 0$, $\lambda_j \geq 0$ and $\lambda_k^* + \lambda_j^* - \lambda_j \geq \lambda_k^* \geq 0$. Furthermore,

$$\sum_{i=1}^{M} \lambda_i^L = \lambda_j + \lambda_k^* + \lambda_j^* - \lambda_j + \sum_{\substack{i=1 \\ i \neq j,k}}^{M} \lambda_i^* = \sum_{i=1}^{M} \lambda_i^* = \sum_{i=1}^{M} \lambda_i.$$

Using our assumption $\lambda_j^* > \lambda_j$ and $\lambda_k^* > \lambda_k$ we get

$$|\lambda_j^L - \lambda_j| + |\lambda_k^L - \lambda_k| = |\lambda_j - \lambda_j| + |\lambda_k^* + \lambda_j^* - \lambda_j - \lambda_k| = |\lambda_j^* - \lambda_j| + |\lambda_k^* - \lambda_k|.$$

Therefore

$$\sum_{i=1}^{M} |\lambda_i^L - \lambda_i| = \sum_{i=1}^{M} |\lambda_i^* - \lambda_i| \leq \alpha,$$

so since λ^* satisfies the conditions of the uncertainty set, λ^L is in the uncertainty set as well.

Similarly, we define arrival rates

$$\lambda_i^H = \begin{cases} \lambda_j^* + \lambda_k^* - \lambda_k, & \text{for } i = j, \\ \lambda_k, & \text{for } i = k, \\ \lambda_i^*, & \text{otherwise}, \end{cases}$$

Now the arrival rate at channel k is lowered to λ_k and the arrival rate at channel j is higher to compensate. By symmetry we can conclude $\lambda^L \in \mathcal{U} \Rightarrow \lambda^H \in \mathcal{U}$.

This leaves us with three different arrival rates for channel j: a low arrival rate λ_j^L, a middle arrival rate λ_j^* and a high arrival rate λ_j^H. All three of these arrival rate are within the interval $(\Lambda - \mu C_k, \mu C_j)$, since

$$\Lambda - \mu C_k = \lambda_j^* + \lambda_k^* - \mu C_k < \lambda_j^* + \lambda_k^* - (\lambda_k^* + \lambda_j^* - \lambda_j) = \lambda_j = \lambda_j^L < \lambda_j^* < \lambda_j^H < \mu C_j,$$

where we used $\Lambda = \lambda_j^* + \lambda_k^*$ and $\mu C_k > \lambda_k^L = \lambda_k^* + \lambda_j^* - \lambda_j$.

According to Lemma 2.3.1 the function $T(x)$ is strictly convex, which gives

$$T(\lambda_j^*) < \max\left\{T(\lambda_j^L), T(\lambda_j^H)\right\}.$$

There exist arrival rates in the uncertainty set with a bigger average delay than the worst case. This is a contradiction with the definition of the worst case, so we can conclude that there can be at most one channel j such that $\lambda_j^* > \lambda_j$.

The proof for $\lambda_l^* < \lambda_l$ is analogous. Suppose there are multiple channels with $\lambda_i^* < \lambda_i$, pick two of them, say l and m. Define

$$\lambda_i^H = \begin{cases} \lambda_l, & \text{for } i = l, \\ \lambda_m^* + \lambda_l^* - \lambda_l, & \text{for } i = m, \\ \lambda_i^*, & \text{otherwise}, \end{cases}$$

and

$$\lambda_i^L = \begin{cases} \lambda_l^* + \lambda_m^* - \lambda_m, & \text{for } i = l, \\ \lambda_m, & \text{for } i = m, \\ \lambda_i^*, & \text{otherwise}, \end{cases}$$

We have $\lambda^H \in \mathcal{U}$ and $\lambda^L \in \mathcal{U}$, because the total deviation from the estimated parameters λ stays the same as for λ^* (similar to the proof above). The arrival rates for channel l are ordered $\lambda_l^L < \lambda_l^* < \lambda_l^H$ and they fall in the interval $(\Lambda - \mu C_m, \mu C_j)$, since

$$\Lambda - \mu C_m = \lambda_l^* + \lambda_m^* - \mu C_m < \lambda_l^* + \lambda_m^* - \lambda_m = \lambda_l^L < \lambda_l^* < \lambda_l^H = \lambda_l < \mu C_l,$$

According to Lemma 2.3.1 the function $T(x)$ is strictly convex, which gives

$$T(\lambda_l^*) < \max\left\{T(\lambda_l^L), T(\lambda_l^H)\right\}.$$

There exist arrival rates in the uncertainty set with a bigger average delay than the worst case. This is a contradiction with the definition of the worst case, so we can conclude that there can be at most one channel l such that $\lambda_l^* < \lambda_l$. □

Now we proceed to the structure of the worst case for given capacities.

Proposition 2.3.3. *The worst case λ^* is of the form*

$$\lambda_i^* = \begin{cases} \lambda_i + \alpha/2, & \textit{for } i = j, \\ \lambda_i - \alpha/2, & \textit{for } i = k, \\ \lambda_i, & \textit{otherwise}, \end{cases}$$

where $j \neq k$.

Proof. By Lemma 2.3.2 the worst case has at most one channel with an arrival rate larger than expected and at most one channel with an arrival rate smaller than expected, say channels j and k. Since the total arrival rate $\sum_{i=1}^{M} \lambda_i$ needs to stay the same, they deviate by the same amount. Call this deviation ϵ. Then the worst case is of the form

$$\lambda_i^* = \begin{cases} \lambda_i + \epsilon, & \text{for } i = j, \\ \lambda_i - \epsilon, & \text{for } i = k, \\ \lambda_i, & \text{otherwise}, \end{cases}$$

Because the worst case has to be in the uncertainty set, see (2.12), the total deviation in arrival rates should be less than α. It follows that $-\alpha/2 \leq \epsilon \leq \alpha/2$. Now we show that $\epsilon = \pm\alpha/2$.

By the assumption of feasibility the capacities should be large enough for any vector of arrival rates in the uncertainty set ($\mu C_i > \tilde{\lambda}_i, \ \forall \tilde{\lambda} \in \mathcal{U}$). Therefore

$$\left[\lambda_j - \alpha/2, \lambda_j + \alpha/2\right] \subset \left(\Lambda - \mu C_k, \mu C_j\right)$$

By Lemma 2.3.1 the average delay is strictly convex in the arrival rate for channel j on the interval $\left(\Lambda - \mu C_k, \mu C_j\right)$, so it is also strictly convex on the subinterval $\left[\lambda_j - \alpha/2, \lambda_j + \alpha/2\right]$. The maximum of a convex function on a compact set is attained at the boundary, so $\lambda_j^* = \lambda_j - \alpha/2$ or $\lambda_j^* = \lambda_j + \alpha/2$. Therefore $\epsilon = \pm\alpha/2$ and the main result follows. □

Proposition 2.3.3 shows that we do not have to optimize over all possible values in the uncertainty set. Instead, there is a finite number of characteristic worst cases in the uncertainty set. It is sufficient to only optimize over the worst cases stated in the proposition. There are $M(M-1)$ arrival rates of that form (M choices for a channel with a higher arrival rate, remaining $M-1$ choices for a channel with a lower arrival rate). This allows us to calculate the robust solution numerically. We illustrate this with the example on page 20.

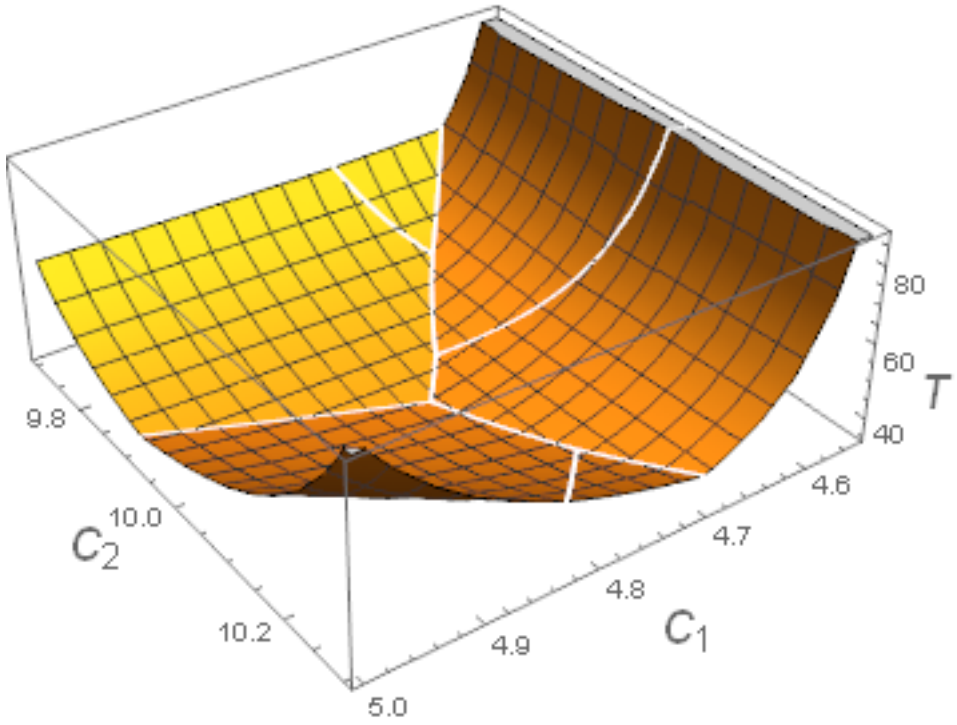

Figure 2.2: Worst average delay T for various capacity assignments $C = (C_1, C_2, 32 - C_1 - C_2)$.

The example shows that the worst average delay function is of a special form: it is convex in the capacities and consists of $M(M-1)$ pieces. Each piece corresponds to a different worst case. A heuristic way to find the minimum is a form of steepest descent. One starts at a point, for example

Example: finding the robust solution for uncertain routing
Consider the uncertain routing model with $M = 3$ channels. For simplicity we set $\mu = \gamma = 1$ and equal costs for capacity $d = (1, 1, 1)$. The arrival rates are $\lambda = (4, 9, 16)$. The total budget is $D = 32$ and hence $D_e = 32 - 4 - 9 - 16 = 3$. We set the uncertainty parameter $\alpha = 1.0$. This allows a maximal increase/decrease in arrival rate of 0.5 per channel. Following Proposition 2.3.3 we get 6 possible worst cases. We call the reduced uncertainty set that only contains these 6 worst cases $\hat{\mathcal{U}}$.

$$\hat{\mathcal{U}} := \\ \big\{(3.5, 9.5, 16.0), (3.5, 9.0, 16.5), (4.0, 9.5, 15.5), (4.0, 8.5, 16.5), (4.5, 9.0, 15.5), (4.5, 8.5, 16.0)\big\}$$

Now the optimization problem (2.7) simplifies to

$$\min_{C_i} \max_{\lambda_i \in \mathcal{U}} \sum_{i=1}^{3} \frac{\lambda_i}{C_i - \lambda_i} = \min_{C_i} \max_{\lambda_i \in \hat{\mathcal{U}}} \sum_{i=1}^{3} \frac{\lambda_i}{C_i - \lambda_i}.$$

For given capacities, we can calculate the worst average delay by calculating the average delay for each of the 6 vectors in $\hat{\mathcal{U}}$ and taking the maximum. In Figure 2.2 the worst average delay is plotted on the vertical axis with the capacities C_1 and C_2 on the horizontal axes. Note that once we pick C_1 and C_2, the rest of the budget fixes C_3. So for three channels we get two degrees of freedom. In the figure we see 6 regions corresponding to the 6 worst cases. In each region a different worst case governs the average delay. The robust solution is the minimum in the figure.
In this example the minimum is attained for capacities $C = (4.782, 10.003, 17.215)$, where three regions intersect. These three regions corresponds to the worst cases $(3.5, 9.5, 16.0)$, $(3.5, 9.0, 16.5)$ and $(4.5, 8.5, 16.0)$. Since the worst average delay is a convex function of the capacities, changing the robust capacities would mean increasing the average delay for one of these three worst cases. The robust solution finds a balance where all three worst cases are equally bad, they all give the same average delay of 34.7825. To compare, the nonrobust capacities would be $C = (4.667, 10.000, 17.333)$. Those capacities have worst case $(4.5, 8.5, 16.0)$ with an average delay of 44.6667. The robust solution takes away capacity from the busiest channel and gives it to the least busy channel for a reduction in average delay from 44.6667 to 34.7825.
Remarkable is that the capacity for the second channel does not change as much between the nonrobust and robust approach, changing from 10 to 10.003. That can be seen in Figure 2.2 as well, since the gradient at the classic solution has a relatively large component for C_1 and a relatively small component for C_2. Therefore changes in C_1 have a bigger impact on the average delay than changes in C_2. The smallest channel benefits the most from extra capacity, the busiest channel is the least impacted by losing capacity, so the optimal strategy is concentrating the capacity changes on the outer channels and leaving the middle channel almost the same.

at the classic solution. Calculate the gradient at that point and move in the direction of the negative gradient. This is the direction where the average delay decreases the most. In the new point we can calculate the gradient again, and so on. At some point we will hit a piece of the function that corresponds to a different worst case, which means we need to go back a little. If we take enough steps like this, we approach the minimum of the function. There are a few caveats like the function not being differentiable at the intersection of two pieces, but this is the general idea.

In this chapter we explained what the CA problem is and what its robust counterpart RCA looks like. We introduced two different forms of uncertainty and showed how to solve for the robust solution numerically. In the next chapter we will use these methods to compare RCA with CA and to show what the effect of Robust Optimization is in this setting. What makes it valuable?

3

Numerical results for the CA problem

In this chapter we use numerical simulation results to compare RCA with CA and to answer the question whether or not Robust Optimization is valuable in this setting.

3.1 Sampling from the uncertainty set

To compare the performance of robust and nonrobust capacities, we need to sample hypothetically true values for the parameters. For each sampled set of parameters we calculate the value of the objective function for the robust and nonrobust optimal solutions. For the CA problem the objective function is the average delay, see (2.1). It is a common practice in the Robust Optimization literature to take a sample such that approximately 95% of the sample is within the uncertainty set. For example, Ben-Tal et al. (2013) sample probability vectors in a newsvendor problem using normal distributions. The mean of these normal distributions is the estimated probability from data and the standard deviations are chosen such that approximately 95% of the sample is within the uncertainty region. We will modify this method to make it applicable to the CA problem.

It is noteworthy that if approximately 95% of the sample is within the uncertainty set, approximately 5% of the samples is outside of it. This is in line with the technique of RO, since the uncertainty set should capture most scenarios, but not all of them. See also Section 1.3 on the construction of uncertainty sets. For queueing models this freedom in sampling can have drastic consequences. In the CA problem for example, a channel could get more arrivals than its capacity. Then the queue at that channel would grow indefinitely and the average delay would be infinite. In that case we call the system infeasible.

If a capacity assignment leads to infeasibility, the entire system stops functioning. Therefore, we will use the probability of infeasibility as one of the performance indicators for a capacity assignment. Both the robust and nonrobust method can experience infeasibility, but the one with the least infeasibility is the most desirable. When we exclude cases where the system is infeasible, we can also compare other performance characteristics like the absolute and relative difference in average delay and the range of values the average delay can take. How close are the minimum and

maximum average delay?

Ben-Tal et al. (2013) use a method centered around the estimated parameters. There is one set of estimated parameters and all samples are generated from a normal distribution around those estimations. One only has to find the robust and nonrobust solutions once, and then apply that solution to the sampled hypothetically true values. Only solving the optimization problem once is an advantage if optimizing requires considerable computational power. That is the case for our uncertain routing, so we also use a method centered around the estimated parameters for that model.

However, one could argue that it is more realistic to take fixed true values for the parameters. The part that changes are the estimated parameters. The estimations are subject to errors and following the Central Limit Theorem it is natural to assume the estimated parameters are normally distributed. Then we can sample estimated parameters and calculate a robust and a nonrobust solution for each of them. This is only possible if calculating the robust solution for given estimated parameters can be done relatively quickly. With overall higher traffic this is the case. For that model we compare the robust and nonrobust capacities for each sampled set of parameters with the true value of the parameters.

In the uncertain routing model we cannot use a normal distribution to sample arrival rates, since the total arrival rate needs to stay the same. Sampling $\lambda_1, ...\lambda_{M-1}$ would fix λ_M. Instead we use an algorithm which is symmetric in all channels and keeps the total arrival rate constant. It also achieves that approximately 95% of the sample is in the uncertainty set.

We use only two sampling methods, but there are many variations possible. For example, if the errors have a known distribution we can adjust our sample for that. But the power of Robust Optimization is that it does not need a distribution for the uncertainty set. We are not even sure that our uncertainty set is correct, so having approximately 95% of the sample within it is somewhat arbitrary. We are testing the performance of Robust Optimization when all its assumptions hold, but we want to know the performance of Robust Optimization in reality. So instead of showing many sampling methods we will draw conclusions that are valid regardless of the sampling method used. What is the influence of the robust method on the capacities? How does this affect the average delay? What is the effect on infeasibility?

3.1.1 Sampling for "overall higher traffic"

The first simulation setup is for the "overall higher traffic" model. Take a fixed situation with flows through the network with rate λ_i. We do not observe the real rates, but get data with some error. We assume the data is generated from a normal distribution with mean λ_i. We assume that the uncertainty set has a reasonable size, so we can choose the standard deviation such that 95% of the probability mass would fall within the uncertainty set if the real λ_i were observed. If the uncertainty set contains deviations up to 5%, the standard deviation is $\sigma_i = 0.05\lambda_i/1.96$, where 1.96 is the 95% quantile of the normal distribution. Then we draw 10^6 samples for the traffic rates from this normal distribution. For each sample, we calculate classic and robust capacities. Then we can calculate the average delay for these capacities for the true arrival rates. This gives 10^6 results for both approaches, so they can be compared.

There are a few implementation issues with this approach. Since the 'observed' rates are drawn from a normal distribution, they could take negative values. This can be solved by redrawing all negative arrival rates. But even a very low observed rate is problematic. Both the classic and the robust approach could assign a capacity that is too low for the actual arrival rate. The result is an

infeasible system. An extremely high observed arrival rate could give similar problems where our methods report that the budget is insufficient even though it was just an error in the observed data. We report the probability of infeasibility separately for both the robust and the nonrobust approach. The mean and standard deviation are computed for only the feasible part of the sample.

Besides drawing from the distribution $N(\lambda_i, 0.05\lambda_i/1.96)$ we construct a second sample while drawing from the distribution $N(0.95\lambda_i, 0.05\lambda_i/1.96)$. This second sample represents an average underestimation of the arrival rates of 5%. This should be an ideal scenario for the robust solution, because it has as a worst case that all arrival rates are 5% higher than predicted.

3.1.2 Sampling for "uncertain routing"

The second simulation setup is for the "uncertain routing" model. Now we take our data as fixed, and the real values of the parameters as the changing factor. We draw possible values from the parameters with an algorithm based on the nature of errors in real-world applications. We consider small errors in the routing of messages over different channels. Instead of using channel j, the message uses channel k. That would mean subtracting ϵ from the arrival rate at channel j and adding that arrival rate to channel k, where ϵ is a small part of the total arrival rate. If we experience N of these small errors, we would get the following algorithm.

1. Start with the data $(\lambda_1, \lambda_2, ..., \lambda_M)$
2. Pick a random channel from 1 to M, say channel j.
3. The new rate for channel j is ϵ higher, $\lambda_j \leftarrow \lambda_j + \epsilon$.
4. Pick a random channel from 1 to M, say channel k.
5. The new rate for channel k is ϵ lower, $\lambda_k \leftarrow \lambda_k - \epsilon$.
6. Repeat step 2-5 for a total of N times.
7. Output $(\lambda_1, \lambda_2, ..., \lambda_M)$

The uncertainty set for uncertain routing (see (2.12)) was based on this scheme. During this algorithm the total arrival rate stays the same - what we add in step 3 is subtracted in step 5 - so we see that condition back in the uncertainty set. The parameter α can be chosen such that approximately 95% of the sample falls within the uncertainty set $\mathcal{U}_\alpha$. Consider the uncertain routing model from the example on page 20 with $M = 3$ channels. For simplicity we had $\mu = \gamma = 1$ and equal costs for capacity $d = (1, 1, 1)$. Furthermore, the observed arrival rates are $\lambda = (4, 9, 16)$. The total budget is $D = 32$ and we can calculate $D_e = 32 - 4 - 9 - 16 = 3$.

We pick the parameters for our algorithm $\epsilon = 0.05$ and $N = 45$. That means we have 45 errors with rate 0.05 on a total arrival rate of $4 + 9 + 16 = 29$. After generating 10^6 sample values we found that approximately 95% of the sample was within $\mathcal{U}_{1.0}$, so we get $\alpha = 1.0$. Solving the optimization problem with the method by Kleinrock gives the nonrobust solution for the capacities $C \approx (4.667, 10.000, 17.333)$. Solving the robust problem (2.7) gives $C \approx (4.782, 10.003, 17.215)$. We will compare those two capacity assignments with each other. We are also interested in the performance of Robust Optimization when the uncertainty set is too large or too small. Does the method fail if the uncertainty set poorly describes the real situation? To test that, we generate two

extra samples, one with $\epsilon = 0.025$ and $N = 45$ ($\mathcal{U}$ too large) and another with $\epsilon = 0.075$ and $N = 45$ ($\mathcal{U}$ too small). As the size for our samples we choose 10^6, since that gives small enough confidence intervals without taking excessive simulation time.

3.2 Overall higher traffic

Consider a network with $M = 4$ channels and arrival rates $\lambda = (4, 6, 8, 10)$. However, due to prediction errors we observe faulty arrival rates from a normal distribution $N(\lambda_i, 0.05\lambda_i/1.96)$ for each channel. The classic approach uses the predicted arrival rates and the robust approach uses 5% higher arrival rates. For simplicity we set $\mu = \gamma = 1$. The costs for capacity are $d = (1, 1.5, 1, 1.5)$, so we can test the influence of different costs per channel. The total budget is $D = 39$. That means that we have excess dollars $D_e = 39 - 4 \cdot 1.0 - 6 \cdot 1.5 - 8 \cdot 1.0 - 10 \cdot 1.5 = 3$ for additional capacity.

Now we simulate 10^6 data sets as described in Section 3.1.1 and calculate the classic and the robust capacities. Then we see what average delay these capacities cause for the real arrival rates. The first question is what the difference between both methods is. On average, the robust approach overestimates the arrival rates. Therefore it assigns more capacity to busy channels and expensive channels, because those are the first bottlenecks if the system becomes busier. The busiest channels get the largest absolute increase in arrival rates and the expensive channels tend to have relatively low capacities in the classic approach, to save money. That does mean that the robust solution assigns less capacity to channels with low arrival rates and cheap channels. This can be seen if we look at the average capacities assigned over all samples in Table 3.1.

True arrival rate	4	6	8	10
True optimal arrival rate	4.51	6.51	8.73	10.66
Nonrobust average capacities	4.51	6.51	8.72	10.66
Robust average capacities	4.41	6.51	8.69	10.77

Table 3.1: Average robust and nonrobust capacities for each of the four channels, sample size 10^6.

The robust approach takes away capacity at the cheap first and third channel to increase the capacity at the big expensive fourth channel. That comes with a price: the robust solution sometimes gives a capacity of less than 4 to the first channel, causing infeasibility. This severe underestimation happened in 0.3% of the simulated data sets. By contrast, the classic approach had a 0.1% probability of infeasibility. On average, the nonrobust capacities are very close to the true optimal capacities. In the nonrobust case infeasibility is usually caused by understaffing the busiest fourth channel. In this example we see that Robust Optimization causes infeasibility more often. But how do both approaches perform when the system is feasible? Table 3.2 shows the probability of infeasibility and the mean and standard deviation of the feasible part of the sample.

We can conclude that Robust Optimization gives worse results here. It causes infeasibility more often and gives longer average delays than the classic, nonrobust approach. It also heavily increases the standard deviation, so there is a higher risk of extreme outcomes. The opposite of what we want. Even if our initial predictions of the arrival rate tend to underestimate the true arrival rate (rightmost column), robust performs worse. In this example Robust Optimization is nothing more than adapting your prediction and optimizing the system for higher arrival rates. That works great if there are actually higher arrival rates, but otherwise it backfires.

Observed mean arrival rates	λ accurate predictions	0.95λ underestimating predictions
Probability of infeasibility in percentage (%)		
nonrobust	[0.105, 0.118]	[0.083, 0.094]
robust	[0.295, 0.317]	[0.173, 0.190]
best method	nonrobust	nonrobust
Mean average delay		
nonrobust	[46.434, 46.469]	[45.520, 45.704]
robust	[52.066, 57.127]	[48.680, 50.235]
best method	nonrobust	nonrobust
Standard deviation of average delay		
nonrobust	[8.787, 8.811]	[46.924, 47.054]
robust	[1287, 1291]	[395.8, 396.9]
best method	nonrobust	nonrobust

Table 3.2: Comparison of infeasibility, mean and standard deviation for overall higher or lower traffic.

We know that the classic capacities are optimal if we predicted the true arrival rates. The robust capacities are optimal if we underestimated all arrival rates by 5%. On the scale from severely underestimating to severely overestimating the classic capacities perform best in the middle, while the robust capacities need an edge case. Since the prediction errors are normally distributed, the errors are symmetric. Overestimating occurs as often as underestimating. That means that in the majority of the time the classic solution is better. The only hope for robust is that the edge case of underestimating causes extreme delays.

A key observation is that extreme delays only occur when the assigned capacities are very close to the true arrival rate. In the formula for average delay (2.1) there is a division by $\mu C_i - \lambda_i$, which gives a huge delay if it is close to zero. High utilization at one channel outweighs medium utilization at all channels. One bottleneck dominates the whole system. That means that the robust solution is protecting us against the wrong risk. The real risk is not underestimating the arrival rate everywhere, it is underestimating the arrival rate at one channel. Table 3.1 shows that robust capacity amplify this problem the first and third channel, only the fourth channel gets a higher capacity. Extreme delays happen more often than with classic capacities, since there are two candidates for bottlenecks instead of one. That is the reason robust has such a high standard deviation and also a higher mean average delay.

All in all Robust Optimization did exactly what we asked it to do. It protects us against the worst case in the uncertainty set. Unfortunately a higher arrival rate at all channels, like the worst case in the uncertainty set, is still better than an extremely high arrival rate at one channel. Because we worried about the wrong risks, the approach did not work as intended. The lesson learned is that not every uncertainty set magically works, it has to capture the relevant risk for the problem. In the CA problem the relevant risk is a concentration of all arrivals on one channel. That is exactly what Proposition 2.3.3 states for the "uncertain routing" model. In the next section we move on to numerical results for that model.

3.3 Uncertain routing

Now we use the three samples of size 10^6 as described in Section 3.1.2. For each sample we calculate performance characteristics for robust and nonrobust capacities. One performance characteristic is the probability of infeasibility, since a feasible system is always more desirable than an infeasible one. For the rest of the sample, where both robust and nonrobust capacities yield a feasible system, we compare the mean and the standard deviation of the average delay. To see if robust capacities prevent extreme outcomes, we compare the probability that the average delay is more than 35. The value of 35 was chosen because it is close to the worst average delay in the uncertainty set. Finally, we show the absolute difference in average delay for 10^4 hypothetically true arrival rates, so we can see the distribution of the difference.

ϵ α	0.025 0.5	0.050 1.0	0.075 1.5
Probability of infeasibility in percentage (%)			
nonrobust	[0., 0.]	[0.111, 0.125]	[2.910, 2.977]
robust	[0., 0.]	[0.021, 0.028]	[1.021, 1.061]
best method	tie	robust	robust
Mean average delay			
nonrobust	[27.439, 27.441]	[29.556, 29.596]	[33.344, 33.412]
robust	[27.645, 27.645]	[29.066, 29.076]	[32.528, 32.610]
best method	nonrobust	robust	robust
Standard deviation of average delay			
nonrobust	[0.524, 0.524]	[10.212, 10.221]	[17.109, 17.125]
robust	[0.538, 0.538]	[2.572, 2.575]	[20.589, 20.607]
best method	nonrobust	robust	nonrobust
Probability of average delay > 35 in percentage (%)			
nonrobust	[0.007, 0.010]	[3.966, 4.043]	[17.740, 17.892]
robust	[0.000, 0.001]	[2.769, 2.833]	[17.179, 17.329]
best method	robust	robust	robust

Table 3.3: Comparison of infeasibility, mean, standard deviation and extreme outcomes for uncertain routing.

Table 3.3 shows our performance characteristics for the three samples. Remember that $\epsilon = 0.050$ is the amount of uncertainty we assumed in the uncertainty set the robust capacities are based on. We also want to know what happens if we are wrong about the magnitude of the uncertainty, so there is a scenario $\epsilon = 0.025$ with much less variation in the arrival rates and a scenario $\epsilon = 0.075$ with a lot more variation than anticipated.

First of all, we see that the robust solution has a lower probability of infeasibility for every sample. Moreover, for the cases $\epsilon = 0.025$ and $\epsilon = 0.050$ the nonrobust capacities were always infeasible when the robust capacities were infeasible. In the case $\epsilon = 0.075$ this happened almost always in our sample of size 10^6. Out of almost 30000 infeasible arrival rates, only 64 times the robust capacities were infeasible while the nonrobust capacities were feasible. If your goal is preventing infeasibility, Robust Optimization is clearly the way to go. However, when there is a small

amount of uncertainty ($\epsilon = 0.025$), infeasibility almost never occurs with the classic capacities as well. More uncertainty makes infeasibility occur more often, but Robust Optimization can reduce the probability significantly.

In the mean average delay we see small deviations. When there is as much uncertainty as assumed ($\epsilon = 0.050$) or even more uncertainty ($\epsilon = 0.075$), the robust solution has a slightly lower mean. Apparently preventing extreme cases has a small positive effect on the mean as well. When there is less uncertainty ($\epsilon = 0.025$), the robust mean is 0.205 higher, which is an increase of about 0.75%. At best the robust mean is a few percent lower than the classic mean, at worst it is about 0.75% higher. The effect on the mean is fairly small for all samples.

The standard deviation of the average delay is a different story. We see that the standard deviation heavily increases as the uncertainty increases. For $\epsilon = 0.050$, which was the assumption the uncertainty set was based on, the robust capacities drastically lowered standard deviation. The robust average delay is more reliable and shows smaller swings throughout the sample. This is characteristic for the robust method. For $\epsilon = 0.025$ and $\epsilon = 0.075$ we found a small increase in standard deviation between the robust and the nonrobust capacities. For $\epsilon = 0.025$ that is understandable, since robust is preparing for scenarios that never happen and the punishment is a small increase in mean and standard deviation.

The increase for $\epsilon = 0.075$ is more remarkable. For the probability of infeasibility and the mean we saw the trend that robust outperforms nonrobust more as the uncertainty increases. Why is that not the case for the standard deviation? The reason is that infeasibility is not taken into account in our calculation of the standard deviation. In the table we can see that nonrobust gets to discard the 3% highest delays due to infeasibility, while robust only gets to discard their highest 1%. The result is a (slightly) higher standard deviation for robust.

The probability that the average delay is higher than 35 is always in favor of the robust capacities. Extreme outcomes happen less often with robust capacities, regardless of the magnitude of the uncertainty. In an absolute sense robust capacities prevent more extreme cases as there is more uncertainty, but the relative reduction is the largest for low uncertainty. For $\epsilon = 0.025$ the absolute decrease is about 0.008% of the cases, which is a relative reduction around 95%! On the other hand, for $\epsilon = 0.075$, around 0.5% fewer scenarios had an average delay of more than 35, but that is a minor relative reduction of around 0.03%. The expected uncertainty with $\epsilon = 0.050$ gives a decrease in probability from 4% to 2.8%, a relative decrease of around 30%. Decreasing the probability of extreme outcomes by 30% is a good reason to use robust capacities.

Now we know the effect of Robust Optimization on the average delay in terms of the mean and the standard deviation, but we can also plot the distribution. In Figures (3.1), (3.2) and (3.3) the absolute differences in average delay are plotted for each sample. We used a sample size of 10^6. Infeasible values were discarded, so the actual sample size is 10^4 minus all infeasible values.

We see that in each of the three graphs more than half of the sample values gave a positive value. That means that in a majority of the cases the classic, nonrobust solution performed better than the robust solution. The strength of RO is in the magnitude of the improvements. For $\epsilon = 0.050$ and $\epsilon = 0.075$ we see that the left side of the graph takes big negative values, while the right side shows smaller positive values. That means that robust sometimes works way better than classic, but robust is (almost) never a lot worse than classic. There is a huge potential upside, and almost no downside.

With the results from sampling we can draw the following conclusions:

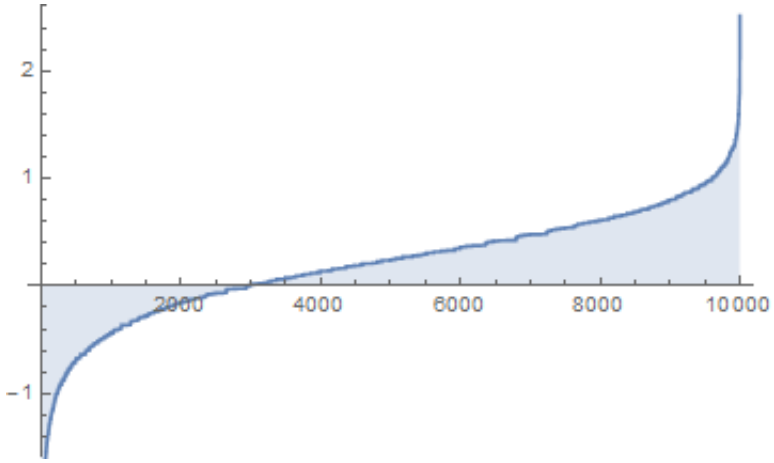

Figure 3.1: Absolute difference in delay for a sample with $\epsilon = 0.025$ of size 10^4.

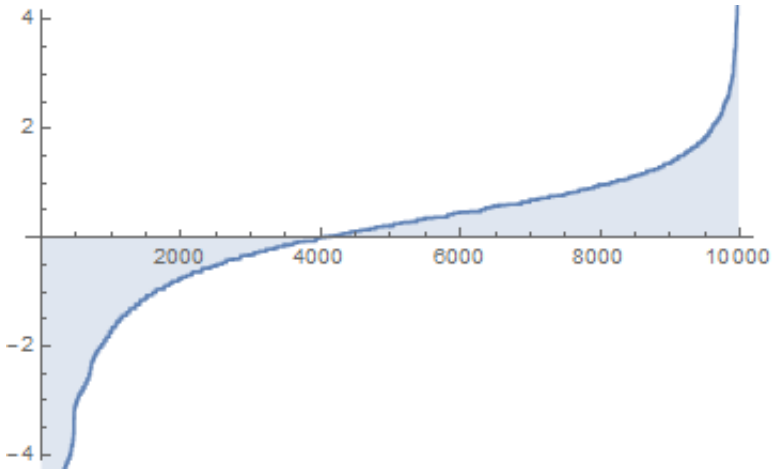

Figure 3.2: Absolute difference in delay for a sample with $\epsilon = 0.050$ of size 10^4.

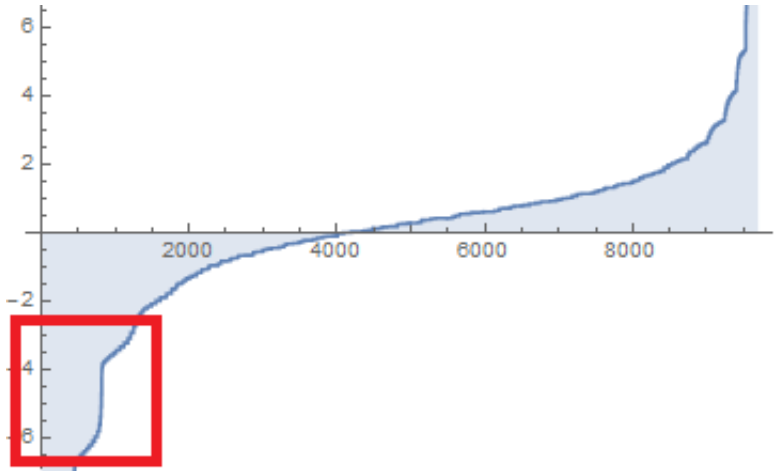

Figure 3.3: Absolute difference in delay for a sample with $\epsilon = 0.075$ of size 10^4.

- **Robust capacities prevent infeasibility.** Infeasibility occurs significantly less often with robust capacities. We saw a decrease of about 75% in the probability of infeasibility. Moreover, the robust method did not introduce new cases where the nonrobust capacities would have been feasible, but the robust capacities are not.
- **Robust capacities prevent very high average delay.** Robust capacities significantly reduce the probability of having an average delay longer than 35. With the right uncertainty set, they also reduce the standard deviation of the average delay significantly (with a factor 4).
- **Robust capacities have little impact on the mean average delay.** The mean average delay for both methods was almost the same, but robust performed a little better when the uncertainty was estimated correctly or underestimated.
- **Robust performance is insensitive to the magnitude of uncertainty.** Even if the uncertainty is severely under- or overestimated, robust capacities do not perform much worse than nonrobust capacities. The conclusions above still hold and at worst the mean and standard deviation of the average delay are slightly higher. That means that the potential downside of robust capacities is small in terms of performance.

3.4 Generalizations

3.4.1 Uncertain routing without sampling

In the previous sections we used a specific sampling method to evaluate the performance of robust capacities. A sampling method is a way to give a weight to various hypothetically true arrival rates. It gives quantitative results like "robust capacities cause 75% less infeasibility". But even without sampling we can calculate the average delay for various hypothetically true arrival rates. That way we get qualitative results about the structure of the robust solution. Those results are more general, since they do not depend on a specific sampling method.

We still consider the uncertain routing model with estimated arrival rates $\lambda = (4, 9, 16)$. In Table 3.4 we see the average delay for the classic, nonrobust approach. In the columns λ_1 varies from 3.3 to 4.7 and in the rows λ_3 varies from 15.3 to 16.7. Once we know λ_1 and λ_3, we can find $\lambda_2 = 29 - \lambda_1 - \lambda_2$. Values in the uncertainty region are shown in red, these are all scenarios that satisfy

$$|\lambda_1 - 4| + |\lambda_2 - 9| + |\lambda_3 - 16| \leq 1.0.$$

We recognize the six corners of the uncertainty region as the six worst cases from Proposition 2.3.3, namely $(3.5, 9.5, 16.0), (3.5, 9.0, 16.5), (4.0, 9.5, 15.5), (4.0, 8.5, 16.5), (4.5, 9.0, 15.5)$ and $(4.5, 8.5, 16.0)$. The table also shows values outside of the uncertainty set, since the true parameter values might be outside the uncertainty set.

In Table 3.5 we see the same table, but now for the robust approach. Robust Optimization finds the capacity assignment that minimizes the largest value in the uncertainty region. This value of 34.7825 (rounded to 34.8 in the table) is attained in the three corners $(3.5, 9.5, 16.0), (3.5, 9.0, 16.5)$ and $(4.5, 8.5, 16.0)$. We see in both tables that the average delay is a convex function of the arrival rates. For the classic solution the minimum is at attained for arrival rates $(4.0, 9.0, 16.0)$. Those are the observed arrival rates. The more the true arrival rates deviate from the observed arrival rates, the higher the average delay grows. For very large deviations, like $(3.3, 10.4, 15.3)$, we even see an average delay of ∞. This means that the system is infeasible, the second channel cannot handle an arrival rate of 10.4.

With these tables we can verify that most of the conclusions in the previous section hold regardless of the distribution of the true uncertainty. Without a sampling method we cannot calculate the mean average delay, so the conclusion "Robust capacities have little impact on the mean average delay" cannot be checked with these tables. The rest of the conclusions do hold.

Robust capacities prevent infeasibility.
We see in the tables that when the robust capacities lead to an infeasible system, the nonrobust capacities produce an infeasible system as well. But infeasibility also occurs in the rightmost column for the nonrobust capacities, while the robust capacities show a feasible system there. The rightmost column corresponds to $\lambda_1 = 4.7$, so the robust capacity for channel 1 of 4.782 is enough while the nonrobust capacity of 4.667 leads to infeasibility. We see the same effect in the top left corner, nonrobust leads to more infeasibility of the second channel since it has a slightly lower capacity for the second channel. For the third channel infeasibility is no problem, the average delay at the bottom of the table did not increase dramatically. The reason is that the third channel has a lot of excess capacity with both methods, so it can handle substantial increases in arrival rates. Robust Optimization uses this fact and assigns a bit less capacity to the third channel without consequences in terms of infeasibility.

Robust capacities prevent very high average delay.
When we look at all possible values for the average delay in the uncertainty set, we see that the nonrobust capacities have a minimum average delay of 27.0 and a maximum average delay of 44.5. In other words, around 95% of the time the average delay is between 27.0 and 44.5. The robust capacities have a minimum average delay of 27.1 and a maximum average delay of 34.8. The minima are almost the same, but the robust solution has a way lower maximum. This shows that Robust Optimization drastically reduces the risk of having a high average delay. The difference is

$\lambda_3 \setminus \lambda_1$	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0	4.1	4.2	4.3	4.4	4.5	4.6	4.7
15.3	∞	∞	∞	∞	∞	111	61.6	45.9	38.8	35.5	34.9	37.3	46.	86.6	∞
15.4	∞	∞	∞	∞	111	61.4	45.4	38.	34.2	32.6	33.	36.	45.1	86.	∞
15.5	∞	∞	∞	111	61.3	45.2	37.5	33.5	31.4	30.7	31.7	35.1	44.5	85.5	∞
15.6	∞	∞	111	61.4	45.2	37.4	33.1	30.7	29.5	29.5	30.8	34.5	44.1	85.3	∞
15.7	∞	111	61.6	45.3	37.4	33.	30.4	28.9	28.3	28.7	30.3	34.2	43.9	85.3	∞
15.8	112	62.	45.6	37.7	33.1	30.4	28.7	27.8	27.7	28.3	30.1	34.1	44.	85.4	∞
15.9	62.5	46.1	38.1	33.5	30.6	28.8	27.7	27.2	27.3	28.2	30.2	34.3	44.2	85.8	∞
16.	46.7	38.7	34.	31.	29.1	27.9	27.2	27.	27.3	28.3	30.4	34.6	44.7	86.2	∞
16.1	39.5	34.7	31.7	29.7	28.4	27.5	27.1	27.1	27.6	28.7	30.9	35.2	45.3	86.9	∞
16.2	35.7	32.6	30.6	29.2	28.2	27.7	27.5	27.6	28.2	29.4	31.7	36.	46.2	87.8	∞
16.3	33.9	31.7	30.3	29.3	28.6	28.2	28.2	28.5	29.2	30.4	32.8	37.2	47.3	89.	∞
16.4	33.3	31.8	30.7	29.9	29.5	29.3	29.4	29.7	30.5	31.8	34.2	38.6	48.8	90.6	∞
16.5	33.7	32.6	31.8	31.3	31.	30.9	31.	31.5	32.3	33.7	36.1	40.6	50.8	92.6	∞
16.6	35.2	34.3	33.7	33.3	33.2	33.2	33.4	33.9	34.8	36.2	38.6	43.1	53.4	95.2	∞
16.7	37.8	37.1	36.7	36.4	36.3	36.4	36.7	37.3	38.2	39.6	42.1	46.6	56.9	98.7	∞

Table 3.4: Average delay for various arrival rates with nonrobust capacities (4.667, 10.0, 17.333).

$\lambda_3 \setminus \lambda_1$	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0	4.1	4.2	4.3	4.4	4.5	4.6	4.7
15.3	∞	∞	∞	∞	3227	108	60.7	45.1	37.8	34.1	32.5	32.7	35.4	43.3	74.3
15.4	∞	∞	∞	3227	108	60.6	44.9	37.4	33.4	31.3	30.6	31.5	34.5	42.7	73.9
15.5	∞	∞	3227	108	60.7	44.9	37.3	33.	30.6	29.5	29.4	30.6	34.	42.4	73.7
15.6	∞	3228	108	61.	45.1	37.3	33.	30.4	28.9	28.3	28.7	30.2	33.7	42.2	73.7
15.7	3228	109	61.3	45.4	37.6	33.1	30.4	28.7	27.8	27.7	28.3	30.	33.6	42.3	73.8
15.8	109	61.9	45.9	38.	33.5	30.6	28.8	27.7	27.3	27.4	28.2	30.	33.8	42.6	74.1
15.9	62.6	46.6	38.6	34.	31.1	29.2	28.	27.3	27.1	27.4	28.3	30.3	34.2	43.	74.7
16.	47.4	39.4	34.8	31.8	29.8	28.5	27.7	27.3	27.2	27.7	28.8	30.8	34.8	43.7	75.4
16.1	40.5	35.8	32.8	30.7	29.3	28.4	27.8	27.6	27.8	28.3	29.5	31.6	35.6	44.6	76.3
16.2	37.1	34.	31.9	30.5	29.5	28.8	28.5	28.4	28.7	29.3	30.5	32.7	36.8	45.8	77.5
16.3	35.6	33.5	32.	30.9	30.2	29.8	29.6	29.6	30.	30.7	32.	34.2	38.3	47.3	79.1
16.4	35.6	34.	32.9	32.1	31.6	31.3	31.2	31.4	31.8	32.6	33.9	36.2	40.3	49.4	81.2
16.5	36.8	35.6	34.8	34.2	33.8	33.6	33.6	33.9	34.3	35.2	36.5	38.9	43.	52.1	83.9
16.6	39.3	38.4	37.8	37.4	37.1	37.	37.1	37.4	37.9	38.8	40.2	42.5	46.7	55.8	87.7
16.7	43.6	43.	42.5	42.2	42.	42.	42.1	42.4	43.	43.9	45.3	47.7	51.9	61.1	92.9

Table 3.5: Average delay for various arrival rates with robust capacities (4.782, 10.003, 17.215).

big; the nonrobust capacities could experience an increase in delay of 64.6% due to uncertainty within the uncertainty set, while the robust capacities keep this increase below 28.8%.

Robust performance is insensitive to the magnitude of uncertainty.
An extreme case is no uncertainty at all. That means that the predicted arrival rates $(4.0, 9.0, 16.0)$ were correct after all and the classic average delay of 27.0 would always be realized. Using robust capacities would lead to an average delay of 27.26 instead of 27.0. That is an increase of less than 1%. Even if all starting assumptions about uncertainty are wrong, we are still less than 1% above the optimal average delay. Robustness comes with a price, but for the CA problem that price is negligible. Robust Optimization has huge potential upsides and a very small downside.

3.4.2 Different forms of uncertainty

Until now we investigated two forms of uncertainty. One was uncertainty about the total volume of traffic, where a robust approach did not work. The other was uncertain routing, where the robust approach was a big improvement. We could also use a hybrid of these forms of uncertainty in a set like

$$\mathcal{U} = \left\{ \tilde{\lambda} \in \mathbb{R}^M : \quad \tilde{\lambda} \geq 0, \quad \left| \sum_{i=1}^{M} \tilde{\lambda}_i - \sum_{i=1}^{M} \lambda_i \right| \leq \beta, \quad \sum_{i=1}^{M} |\tilde{\lambda}_i - \lambda_i| \leq \alpha \right\}.$$

This set allows a deviation β in the total volume while the sum of the deviations at all channels can be at most α. Unfortunately an increase in the total volume suggests increasing the capacity for the channel with the highest arrival rate, while for uncertain routing it is better to take away capacity from that channel. The two forms of uncertainty work against each other. Instead of using a hybrid model we advise to use just the uncertain routing model, for the total volume a single point estimate works better.

There are also other model parameters where uncertainty plays a role. The service rate μ and the capacities C_i could be uncertain as well. One reason could be prediction errors, like not knowing the average service time. Another source of uncertainty could be implementation errors. Exactly implementing the capacities C_i can be difficult, especially if capacity consists of discrete units (extra employees for example). Luckily, models with uncertainty in the service times look very similar to uncertainty in the arrival rates. Remember that the average delay function is

$$\min_{C_i} \sum_{i=1}^{M} \frac{\lambda_i}{\gamma} \left[\frac{1}{\mu C_i - \lambda_i} \right].$$

Uncertainty in λ_i or μC_i has a similar effect on the denominator $\mu C_i - \lambda_i$. What matters is the difference between the capacity and the arrival rate. That is the dominating factor that determines the average delay. There is a minor difference between uncertainty in arrival or service times, because the average delay at each channel is weighted by the arrival rate λ_i. That means that arrival rate uncertainty is amplified a bit more, since a higher arrival rate than expected not only causes a longer delay at that channel, but also gives it a higher weight.

A model with constant capacity looks like the model for uncertain routing. For example, when you have to decide how to divide the total server capacity over different message types. A similar result to Proposition 2.3.3 and the supporting Lemmas holds. It is easily verified that the average

delay is a strictly convex function of the capacities, and the rest of the proofs do not even rely on the average delay function. A model where the capacity of each channel is uncertain independent of the other channels is similar to the overall higher or lower traffic model. In that case it is better to avoid the robust method we used.

Another extension is assigning different costs d_i to buying capacity for channel i. That makes the relationship between the capacities a little more complicated. The budget does not always buy you the same amount of capacity. This only changes the allowed capacity assignments, but the optimal classic capacities are still given by equation (2.6). In the robust optimization problem the budget constraint $D = \sum_{i=1}^{M} C_i$ is replaced by $D = \sum_{i=1}^{M} d_i C_i$, the solution methods are the same.

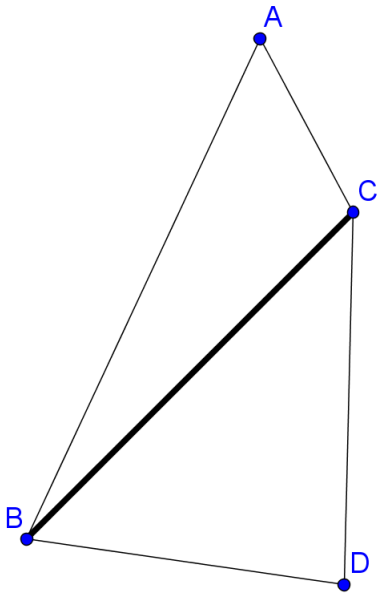

Figure 3.4: Road network between cities A, B, C and D.

Let us end the section with a model of a road network.

Example: road network

Consider the road network between the cities A, B, C and D as shown in Figure 3.4. There is a highway from city B to C and there are smaller roads from A to B, A to C, B to D and C to D. The highway has a high capacity and has a high maximum speed. That means that someone traveling from A to B has a choice; the shortest route via the road from A to B, or the fast route over the highway from A to C to B. Even if we know how many people travel from A to B, we do not know which route they will take.

The population in this area is rising, so more traffic is expected in the future. To prevent the road network from getting congested, there is a budget to upgrade the road network by increasing the capacity. There are two plans:

1. Spending most of the money on the highway. This is an efficient way to spend the budget if most of the extra traffic uses the highway.

2. Spending most of the money on the smaller roads. The smaller roads could become congested as the population rises, while the highway still has enough capacity.

We want to pick the plan with the lowest probability of congestion. This is a typical example of a CA problem with uncertain routing. We can estimate the amount of traffic with a rising population, but it is uncertain which roads will handle that extra traffic. In the classic approach we would predict how the traffic is divided and pick the optimal capacity assignment for that prediction. A robust approach would consider worst case scenarios and pick the capacity assignment that minimizes the worst delay.

As we have seen before, the robust method favors small servers. The small roads will get congested the fastest with an increase in arrival rate, so they get the lion share of the extra capacity (plan 2). The robust solution is better at preventing huge traffic jams due to increasing population, which was the goal of the upgrade. Giving more capacity to the highway might do better on average, but it also performs really bad in a pessimistic scenario.

Besides the uncertainty where the extra traffic will appear, it is also uncertain how much traffic will increase overall. At best, we can have an estimate of the population growth and the relationship with the increase in traffic. We could assume the most pessimistic scenario for our capacity, but we have seen that it is better to use a classic approach with an estimate for the rise in traffic.

Note that the CA problem as a model has some limitations in this situation. One assumption is independence between the traffic rates on different roads. In reality we expect that when two roads are connected, their traffic rates are dependent. The other assumption is that any continuous capacity assignment is possible. In reality, increasing the capacity of a road can be done in discrete amounts by adding a lane. This can be easily solved by adding extra constraints on the capacities. If there are only a finite number of capacity assignments allowed, we can simply try them all when searching for the optimal assignment.

4

Worst case perspective in multiserver queues

In this chapter we consider multiserver queues under Halfin-Whitt scaling. We are interested in the queue length in a worst case scenario and use that as a bound. We introduce the problem, prove several results and discuss open questions regarding this research area.

4.1 Problem description

In previous chapters we used Robust Optimization to deal with uncertainty in the model parameters. But queueing models also exhibit uncertainty in arrivals and departures. In this chapter we take it one step further and replace the probabilistic nature of arrivals and departures by arbitrary values restricted by constraints. Due to the deterministic nature of the model it is easier to compute performance measures, but because we use probabilistic laws in the starting assumptions, the results are still connected to the stochastic model. One of the first efforts in this area can be found in Bertsimas et al. (2011).

The analysis of multiserver queues tends to be a challenging area of queueing theory. Many results require Poisson arrival and/or service time distributions. In general we lack the tools to compute bounds on key performance measures, such as the queue length. The advantage of the Robust Optimization approach in a queueing context is that knowledge about the underlying distribution is not required. That makes it a promising method for models with general arrival and service distributions. There are some bounds for general models, for example Kingman (1970) derived bounds on the mean waiting time for the $G/G/s$ queue. Such bounds are not known for the Halfin-Whitt regime, so we consider a model with that scaling.

We are interested in a very pessimistic scenarios with many arrivals in a short time and long service times. We characterize the allowed pessimistic scenarios in the form of an uncertainty set. Under the assumption that the arrival times and service times satisfy the conditions of the uncertainty sets, we obtain upper bounds for the queue length. This gives us a worst case perspective on the $G/G/s$ queue. Just like in the other examples in this thesis this is not the worst scenario that could happen, but the worst scenario that we want to consider in the uncertainty set. Starting from

the situation with all servers busy, the queue length will grow in a pessimistic scenario.

In the long run there is more service capacity than arriving load, so the queue will eventually become empty again. But what happens in the meantime? How large can the queue length become and at what time is it maximal? There are a few parameters in the model, namely the size of the uncertainty sets for the arrival and service times and the number of servers in relation to the load. Our results directly express performance measures in these parameters. This can help decision makers in weighing risk versus reward. How many servers do you need to reach a given risk level? What is the effect on the worst case queue length if you reduce the number of servers? With a certain number of servers, how long can a busy period be?

4.1.1 Model

The model we consider is an FCFS multiserver $G/G/s$ queue in steady state. That means that there are s servers and we do not specify the arrival time and service time distributions, hence the G for 'General'. We use the scaling introduced by Halfin and Whitt (1981), which connects the number of servers and the arrival rate for large systems. Fix some excess parameter $\beta > 0$ and let $\lambda_s = s - \beta\sqrt{s}$. Furthermore let A and B denote some random variables with non-negative support, finite expectation $\mathbb{E}[A] = \mathbb{E}[B] = 1$ and finite variance. Now consider the FCFS $G/G/s$ queue with interarrival times drawn i.i.d. distributed as $A\lambda_s^{-1}$ and processing times drawn i.i.d. distributed as B.

Because we are interested in the queue length, we use a setting that was also used by Gamarnik and Goldberg (2013). The idea is keeping track of the number of jobs in the queue by counting the number of arrivals and departures. To that end we introduce the following renewal processes.

$$
\begin{aligned}
A(t) &= \text{number of arrivals at time } t \text{ with renewal distribution } A\lambda_s^{-1} \\
N_i(t) &= \text{number of departures from server } i \text{ at time } t \text{ with renewal distribution } B \\
X(t) &= \text{total number of departures from all servers combined at time } t = \sum_{i=1}^{s} N_i(t)
\end{aligned}
$$

At time 0 we start counting, so $A(0) = 0$ and $N_i(0) = 0$ for all $i = 1, ...s$. One concern could be that the arrival process should not grow faster than the departure process: for stability every arriving customer needs to leave the system eventually. On average an interarrival time is $\mathbb{E}[A\lambda_s^{-1}] = \lambda_s^{-1}$ while the average time between departures is at most $\mathbb{E}[B] = 1$ for each server, so s^{-1} at all servers. Under the Halfin-Whitt scaling $\lambda_s < s$ ensures stability. However, as s gets larger the load on the system increases as λ_s approaches s. Where relevant, we will make the dependence on the number of servers explicit with a subscript s, but sometimes we leave it out for ease of notation.

Now a few assumptions are needed to describe the situation we are interested in. We consider the system in steady state, which means the first renewals only have a residual interarrival or service time left. The corresponding processes are called an equilibrium renewal process. As the starting state take one job in every server and an empty queue. This represents the moment just before the queue starts growing. Our perspective is worst case, we want to find the longest possible queue length. To achieve that, the servers should be working all the time. Every time a server finishes a job, there is a nonempty queue to draw from. If a server ever becomes idle, the queue length must be zero.

In the long run a server will become idle, but what happens after that time is irrelevant for the worst case queue length. We might as well assume that there is no idle time when building up to

the longest queue length. This assumption is true if there are more arrivals than departures. When the servers are always working, the consequence is that the departure processes at each server are independent of departures at other servers and independent of the arrival process.

- $A(t) \geq \sum_{i=1}^{s} N_i(t)$ for any $t > 0$.
- $A(t)$ and $N_i(t)$ are mutually independent for all $i = 1, ..., s$.

The queue length $Q(t)$ at time t is the number of arrivals minus the number of departures at all queues, so $Q(t) = A(t) - \sum_{i=1}^{s} N_i(t)$. Note that this is only true under our assumption of more arrivals than departures. Also note that we took a queue length of 0 at time $t = 0$ to keep things simple, but we might as well start with a positive queue length $Q(0)$. Then $Q(t) = A(t) - \sum_{i=1}^{s} N_i(t) + Q(0)$ and in the following analysis $Q(t)$ would be a constant $Q(0)$ larger. Since we are after the scaling of the queue length in the number of servers s, this constant disappears in the final result.

4.1.2 Worst case queue length

Since $A(t)$ is an equilibrium renewal process we have $\mathbb{E}[A(t)] = t\lambda_s = t(s - \beta\sqrt{s}) = ts - \beta t\sqrt{s}$. Similarly, using the independent and identically distributed departure processes at the servers, $\mathbb{E}[X(t)] = \mathbb{E}[\sum_{i=1}^{s} N_i(t)] = s\mathbb{E}[N_1(t)] = ts$. For the standard deviation we have $\sigma(X(t)) = \sigma(\sum_{i=1}^{s} N_i(t)) = \sqrt{s} \cdot \sigma(N_1(t))$. By the Central Limit Theorem (CLT), as $t \to \infty$, the random variables

$$\frac{A(t) - (ts - \beta t\sqrt{s})}{\sigma\big(A(t)\big)} \tag{4.1}$$

and

$$\frac{X(t) - ts}{\sigma\big(X(t)\big)} \tag{4.2}$$

are asymptotically standard normal. Now the idea is bounding the number of arrivals and departures by discarding the tails of these standard normal random variables. Since a standard normal variable Z satisfies $\mathbb{P}(Z \leq 2) \approx 0.975$ and $\mathbb{P}(Z \leq 3) \approx 0.995$ we can take a value Γ_a around 2 or 3 such that random variable (4.1) stays below it with high probability. Similarly we can take a value Γ_s around 2 or 3 such that random variable (4.2) stays above it with high probability. Note that we want an upper bound on the arrivals, but a lower bound on the departures to keep the waiting time under control.

This motivates us to construct the following uncertainty set for the realized number of arrivals $A_{t,s}$ at time t :

$$\mathcal{U}^a = \big\{A_{t,s} \;\big|\; A_{t,s} \leq ts - \beta t\sqrt{s} + \Gamma_a \sigma\big(A(t)\big),\ \forall t \geq 0\big\}$$

and the uncertainty set for the realized total number of departures from all queues $X_{t,s}$ at time t:

$$\mathcal{U}^s = \big\{X_{t,s} \;\big|\; X_{t,s} \geq ts - \Gamma_s \sigma\big(X(t)\big),\ \forall t \geq 0\big\}$$

Note that similar uncertainty sets were used in Bandi et al. (2015). Each renewal process can take its expected value, but there is a margin of error based on the CLT. The parameters Γ_a and

Γ_s determine the allowed deviation from the expected values. They are chosen such that each individual equation is satisfied with high enough probability. Γ_a and Γ_s are used to reflect the accepted risk level.

There is a problem with this choice for the uncertainty sets. Even though each inequality has a high probability of being satisfied, the uncertainty sets contain a lot of inequalities. There is at least one inequality for every arrival/departure in the busy period we are interested in. The probability that a given arrival/departure process is in this uncertainty set is roughly

$$\mathbb{P}(\text{arrival/departure process is in } \mathcal{U}) \approx \mathbb{P}(\text{one inequality safisfied})^{[\text{number of inequalities}]},$$

which can get very small for large systems and when investigating over a long time horizon. Basically the assumptions are almost never satisfied in the stochastic system, so results for this model do not automatically hold in the stochastic system we are interested in. However, this uncertainty set does have a reasonable motivation through the CLT, so we believe it can still provide a valuable perspective. Constructing an uncertainty set is a trade-off between realism and tractability, and here we go for tractability. The strict nature of our uncertainty set enables us to bound the queue length for arbitrary arrival and service distributions.

The queue length for the system is the number of arrivals minus the number of departures, so the longest queue length for a given arrival sequence $A_{t,s}$ and departure sequence $X_{t,s}$ occurs at time t^*, where t^* maximizes

$$Q_s = \max_{t\geq 0} \big[A_{t,s} - X_{t,s}\big].$$

We want to look at the worst (longest) queue length Q_s^* for arrivals and departures within the uncertainty set, which gives

$$Q_s^* = \max_{A_{t,s}\in\mathcal{U}^a} \max_{X_{t,s}\in\mathcal{U}^s} \max_{t\geq 0} \big[A_{t,s} - X_{t,s}\big]. \tag{4.3}$$

Now we exchange the order of maximization to get

$$Q_s^* \leq \max_{t\geq 0} \left[\max_{A_{t,s}\in\mathcal{U}^a} A_{t,s} - \max_{X_{t,s}\in\mathcal{U}^s} X_{t,s} \right].$$

In words, we first pick the worst possible arrival and departure processes and then find the moment in time where the queue length is maximal. We will show that this bound is in fact tight by pointing out arrival times and departure times that achieve the worst case that do not depend on the time t.

Lemma 4.1.1. *In a $G/G/s$ FCFS queue in the Halfin-Whitt regime, there exists an arrival sequence $A_{t,s}^* \in \mathcal{U}^a$ and a departure sequence $X_{t,s}^* \in \mathcal{U}_s^s$ achieving*

$$Q_s^* = \max_{t\geq 0} \left[\max_{A_{t,s}\in\mathcal{U}^a} A_{t,s} - \max_{X_{t,s}\in\mathcal{U}_s^s} X_{t,s} \right]. \tag{4.4}$$

Proof. Since $X_{k,s}$ and $A_{k,s}$ are independent, we can maximize them separately over their uncertainty sets. We take all arrivals as early as possible and all departures as late as possible.
Take the sequence of departure times such that

$$X_{t,s}^* = \big\lceil ts - \Gamma_s \sigma\big(X(t)\big)\big\rceil, \quad \forall t \geq 0.$$

Similarly, take the sequence of arrival times such that

$$A^*_{k,s} = \big\lfloor ts - \beta t\sqrt{s} + \Gamma_a \sigma\big(A(t)\big)\big\rfloor, \;\; \forall t \geq 0.$$

These are the maximum values still contained in their corresponding uncertainty set and they do not depend on the time t. Therefore we can take the maximization over t out of equation (4.3). Note that these arrival and departure times are not unique, as long as the order of arrivals and departures stays the same the queue length is also maximized. □

4.2 Bounds in the Halfin-Whitt regime

In the literature some bounds can be found on the queue length in a $G/G/s$ queue. The problem is that these bounds do not give useful information in the Halfin-Whitt regime, they grow too fast with s. Gamarnik and Goldberg (2013) show that the queue length scales like $\mathcal{O}(\sqrt{s})$. One of the relevant bounds is a result by Kingman (1970), who states for the mean waiting time in a $G/G/s$ queue (note that we used $\mu = 1$)

$$\mathbb{E}[W] \leq \frac{1}{2}\frac{s\sigma_a^2 + \sigma_s^2 + (1 - 1/s)}{\frac{s}{\lambda} - 1} = \frac{\rho\big(s\sigma_a^2 + \sigma_s^2 + (1 - 1/s)\big)}{2(1-\rho)} = \frac{\lambda\big(\sigma_a^2 + \sigma_s^2/s + (1/s - 1/s^2)\big)}{2(1-\rho)}.$$

Through Little's law $\mathbb{E}[Q_s] = \lambda\mathbb{E}[W]$ this also yields a bound for the mean queue length. When we 'translate' this bound to the Halfin-Whitt regime, using $(1-\rho) \approx \beta/\sqrt{s}$, we get

$$\mathbb{E}[Q_s] = \lambda\mathbb{E}[W] \leq \sqrt{s} \cdot \frac{s^2\big(\sigma_a^2 + \sigma_s^2/s + (1/s - 1/s^2)\big)}{2\beta}.$$

The bound by Kingman does not survive the Halfin-Whitt regime, since it grows too fast with s. Recently progress has been made with a robust derivation by Bandi et al. (2015), who showed for the mean system time S (waiting time plus service time)

$$\mathbb{E}[S] \leq \frac{\lambda}{4}\frac{(\Gamma_a + \Gamma_s/\sqrt{s})^2}{1-\rho} + \frac{s}{\lambda}.$$

Unfortunately this bound also grows too fast in s to be used in the Halfin-Whitt regime.

In this section we prove a bound on the queue length that does have the desired scaling of $\mathcal{O}(\sqrt{s})$. In particular, we are interested in the limit of the worst case queue length $\lim_{s\to\infty} Q^*_s/\sqrt{s}$. This is possible because we construct a bound on the worst case queue length instead of the mean queue length. The interpretation of 'worst case' depends on the uncertainty sets chosen, ideally they are both realistic and tractable. In Section we discuss the relevance, but first we show tractability of the uncertainty set. First we show the structure of the proof for the special $M/M/s$ case and then move on to the general case of the $G/G/s$ queue.

4.2.1 M/M/s queue

For the $M/M/s$ queue we can explicitly calculate the standard deviations of the renewal processes, so it is a good test case. The goal of this section is showing the proof structure without the complications of the general case. Since an equilibrium renewal process with an exponential renewal

distribution is just a Poisson distribution, the standard deviation is the square root of the expected number of renewals. We find

$$\sigma\big(A(t)\big) = \sqrt{t\lambda_s} = \sqrt{t\big(s - \beta\sqrt{s}\big)} = \sqrt{ts - \beta t\sqrt{s}} \tag{4.5}$$

and, using independence between servers,

$$\sigma\big(X(t)\big) = \sqrt{s} \cdot \sigma\big(N_1(t)\big) = \sqrt{ts}.$$

The worst case arrival process is

$$A^*_{t,s} = \left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sqrt{ts - \beta t\sqrt{s}} \right\rfloor$$

and the worst case departure process is

$$X^*_{t,s} = \left\lceil ts - \Gamma_s\sqrt{ts} \right\rceil.$$

Proposition 4.2.1. *In an $M/M/s$ queue in the Halfin-Whitt regime satisfying $A_{t,s} \in \mathcal{U}^a$ and $X_{t,s} \in \mathcal{U}^s$ the queue length is bounded by*

$$\lim_{s\to\infty} \frac{Q_s}{\sqrt{s}} \leq \frac{(\Gamma_a + \Gamma_s)^2}{2\beta}. \tag{4.6}$$

Proof. We will calculate an upper bound for the worst queue length, Q^*_s, and use that as the upper bound for all queue lengths in the uncertainty set. Starting from equation (4.4) and substituting the worst case equilibrium processes we get

$$\begin{aligned}
Q^*_s &= \max_{t\geq 0}\left[\max_{A_{t,s}\in\mathcal{U}^a} A_{t,s} - \max_{X_{t,s}\in\mathcal{U}^s} X_{t,s}\right] \\
&= \max_{t\geq 0}\left[\left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sqrt{ts - \beta t\sqrt{s}}\right\rfloor - \left\lceil ts - \Gamma_s\sqrt{ts}\right\rceil\right] \\
&= \max_{t\geq 0}\left[\left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sqrt{ts - \beta t\sqrt{s}}\right\rfloor + \left\lfloor -ts + \Gamma_s\sqrt{ts}\right\rfloor\right] \\
&\leq \max_{t\geq 0}\left[-\beta t\sqrt{s} + \Gamma_a\sqrt{ts - \beta t\sqrt{s}} + \Gamma_s\sqrt{ts}\right] \\
&= \sqrt{s}\cdot\max_{t\geq 0}\left[-\beta t + \Gamma_a\sqrt{t - \frac{\beta t}{\sqrt{s}}} + \Gamma_s\sqrt{t}\right]
\end{aligned}$$

At this point we see that the queue length scales like $\mathcal{O}(\sqrt{s})$, exactly what one would expect in this regime. Divide left and right by the desired scaling factor $\sqrt{s}$ and take the limit for $s \to \infty$.

$$\lim_{s\to\infty}\frac{Q^*_s}{\sqrt{s}} \leq \lim_{s\to\infty}\max_{t\geq 0}\left[-\beta t + \Gamma_a\sqrt{t - \frac{\beta t}{\sqrt{s}}} + \Gamma_s\sqrt{t}\right] = \max_{t\geq 0}\left[-\beta t + \big(\Gamma_a + \Gamma_s\big)\sqrt{t}\right]. \tag{4.7}$$

What remains is finding t^*, the time at which the worst case queue length is maximal. Denote the function to be maximized by $f(t) := -\beta t + \big(\Gamma_a + \Gamma_s\big)\sqrt{t}$. We calculate $f'(t) = -\beta + \frac{\Gamma_a+\Gamma_s}{2\sqrt{t}}$ and $f''(t) = -\frac{\Gamma_a+\Gamma_s}{4t\sqrt{t}}$. Since $f''(t) < 0$ for all $t > 0$, the function $f(t)$ is concave and we can simply maximize it by solving $f'(t) = 0$. The solution is $\frac{(\Gamma_a+\Gamma_s)^2}{4\beta^2}$. That means that the time where the queue length divided by the square root of s is maximized is $t^* = \frac{(\Gamma_a+\Gamma_s)^2}{4\beta^2}$. Finally, by substituting this value for t in equation (4.7) we get

$$\begin{aligned}
\lim_{s\to\infty} \frac{Q_s^*}{\sqrt{s}} &\leq \max_{t\geq 0} \left[-\beta t + \big(\Gamma_a + \Gamma_s\big)\sqrt{t} \right]. \\
&= -\beta \cdot \frac{(\Gamma_a+\Gamma_s)^2}{4\beta^2} + (\Gamma_a+\Gamma_s)\sqrt{\frac{(\Gamma_a+\Gamma_s)^2}{4\beta^2}} \\
&= -\frac{(\Gamma_a+\Gamma_s)^2}{4\beta} + \frac{(\Gamma_a+\Gamma_s)^2}{2\beta} \\
&= \frac{(\Gamma_a+\Gamma_s)^2}{2\beta}
\end{aligned}$$

Because the worst case queue length is the longest possible queue length under the conditions of the uncertainty set, we have $Q_s \leq Q_s^*$ for any arrival/departure process.

$$\lim_{s\to\infty} \frac{Q_s}{\sqrt{s}} \leq \lim_{s\to\infty} \frac{Q_s^*}{\sqrt{s}} \leq \frac{(\Gamma_a+\Gamma_s)^2}{2\beta}.$$

□

Proposition 4.2.1 gives some insight in the queue length of the $M/M/s$ queue in the Halfin-Whitt regime. One of the main insights is the dependence on the excess parameter β. When you double β, the worst case queue length is halved. This is very useful in determining how much excess capacity is necessary for an appropriate risk level. The dependence on the uncertainty in the arrival and departure process is also remarkable. The parameters Γ_a and Γ_s play a symmetric role in the $M/M/s$ queue. When aiming to reduce queue lengths, one should reduce the total variability $\Gamma_a + \Gamma_s$ as much as possible. There is no bias towards one process or the other.

Another interesting aspect is that the time at which the worst case queue length is achieved, $t^* = \frac{(\Gamma_a+\Gamma_s)^2}{4\beta^2}$, does not depend on s. As s increases there are more arrivals and departures in that time period. A lot more happens in this time interval, but time itself does not scale. This tells us how much time this system needs to go from an empty queue to an extremely long one. Now we extend the results to the more general $G/G/s$ model.

4.2.2 G/G/s queue

For general arrival and service distributions we need the standard deviation of the corresponding equilibrium renewal processes. The standard deviation of a renewal process $N(t)$ with renewal distribution X is (see Nebres, 2011)

$$\lim_{t\to\infty} \frac{\mathrm{Var}[N(t)]}{t} = \frac{\mathrm{Var}[X]}{(\mathbb{E}[X])^3}.$$

Lemma 4.2.2. *For the given arrival and departure processes with $\mathbb{E}[A] = \mathbb{E}[B] = 1$ and with $\sigma_a, \sigma_s < \infty$, we have*

$$\sigma\big(A(t)\big) = \sigma_a\sqrt{ts - \beta t\sqrt{s}}$$

and

$$\sigma\big(X(t)\big) = \sigma_s\sqrt{ts}.$$

Proof. Using the result for the variance of an equilibrium renewal process and the starting assumption $\mathbb{E}[A] = 1$ we find

$$\frac{\sigma^2\big(A(t)\big)}{t} = \frac{\text{Var}[A\lambda_s^{-1}]}{(\mathbb{E}[A\lambda_s^{-1}])^3} = \frac{\lambda_s^{-2}\text{Var}[A]}{\lambda_s^{-3}\mathbb{E}[A]^3} = \lambda_s\text{Var}[A]$$

and when we calculate the standard deviation

$$\sigma\big(A(t)\big) = \sqrt{t\lambda_s\text{Var}[A]} = \sigma_a\sqrt{t\lambda_s} = \sigma_a\sqrt{ts - \beta t\sqrt{s}}.$$

For the departure process we find a similar result when using $\mathbb{E}[B] = 1$,

$$\frac{\sigma^2\big(N_1(t)\big)}{t} = \frac{\text{Var}[B]}{(\mathbb{E}[S])^3} = \text{Var}[B]$$

and

$$\sigma\big(X(t)\big) = \sqrt{s}\sigma\big(N_1(t)\big) = \sqrt{s}\cdot\sqrt{t\text{Var}[B]} = \sigma_s\sqrt{ts}.$$

□

With this Lemma we can tackle the problem for the general $G/G/s$ model, provided that the standard deviation is finite.

Proposition 4.2.3. *In a $G/G/s$ queue in the Halfin-Whitt regime satisfying $A_{t,s} \in \mathcal{U}^a$ and $X_{t,s} \in \mathcal{U}^s$ and with $\sigma_a, \sigma_s < \infty$, the queue length is bounded by*

$$\lim_{s\to\infty}\frac{Q_s}{\sqrt{s}} \leq \frac{(\Gamma_a\sigma_a + \Gamma_s\sigma_s)^2}{2\beta}. \tag{4.8}$$

Proof. The idea of the proof is similar to the $M/M/s$ case. We calculate an upper bound for the worst queue length, Q_s^*, and use that as the upper bound for all queue lengths in the uncertainty set. Starting from equation (4.3), substituting the worst case equilibrium processes from Lemma

4.1.1 and then the standard deviations from Lemma 4.2.2, we get

$$\begin{aligned}
Q_s^* &= \max_{t\geq 0}\left[\max_{A_{t,s}\in\mathcal{U}^a} A_{t,s} - \max_{X_{t,s}\in\mathcal{U}^s} X_{t,s}\right] \\
&= \max_{t\geq 0}\left[\left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sigma\big(A(t)\big)\right\rfloor - \left\lceil ts - \Gamma_s\sigma\big(X(t)\big)\right\rceil\right] \\
&= \max_{t\geq 0}\left[\left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sigma_a\sqrt{ts-\beta t\sqrt{s}}\right\rfloor - \left\lceil ts - \Gamma_s\sigma_s\sqrt{ts}\right\rceil\right] \\
&= \max_{t\geq 0}\left[\left\lfloor ts - \beta t\sqrt{s} + \Gamma_a\sigma_a\sqrt{ts-\beta t\sqrt{s}}\right\rfloor + \left\lfloor -ts + \Gamma_s\sigma_s\sqrt{ts}\right\rfloor\right] \\
&\leq \max_{t\geq 0}\left[-\beta t\sqrt{s} + \Gamma_a\sigma_a\sqrt{ts-\beta t\sqrt{s}} + \Gamma_s\sigma_s\sqrt{ts}\right] \\
&= \sqrt{s}\cdot\max_{t\geq 0}\left[-\beta t + \Gamma_a\sigma_a\sqrt{t-\frac{\beta t}{\sqrt{s}}} + \Gamma_s\sigma_s\sqrt{t}\right]
\end{aligned}$$

Again we notice that the queue length scales like $\mathcal{O}(\sqrt{s})$, exactly what one would expect in the Halfin-Whitt regime. Divide left and right by the desired scaling factor $\sqrt{s}$ and take the limit for $s\to\infty$.

$$\lim_{s\to\infty}\frac{Q_s^*}{\sqrt{s}} \leq \max_{t\geq 0}\left[-\beta t + (\Gamma_a\sigma_a + \Gamma_s\sigma_s)\sqrt{t}\right]. \tag{4.9}$$

What remains is finding t^*, the time at which the worst case queue length is maximal. Denote the function to be maximized by $f(t) := -\beta t + \big(\Gamma_a\sigma_a + \Gamma_s\sigma_s\big)\sqrt{t}$. We calculate $f'(t) = -\beta + \frac{\Gamma_a\sigma_a+\Gamma_s\sigma_s}{2\sqrt{t}}$ and $f''(t) = -\frac{\Gamma_a\sigma_a+\Gamma_s\sigma_s}{4t\sqrt{t}}$. Since $f''(t) < 0$ for all $t > 0$, the function $f(t)$ is concave and we can simply maximize it by solving $f'(t) = 0$. The solution is $\frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{4\beta^2}$. That means that the time where the queue length divided by the square root of s is maximized is $t^* = \frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{4\beta^2}$. Finally, by substituting this value for t in equation (4.7) we get

$$\begin{aligned}
\lim_{s\to\infty}\frac{Q_s^*}{\sqrt{s}} &\leq \max_{t\geq 0}\left[-\beta c + (\Gamma_a\sigma_a + \Gamma_s\sigma_s)\sqrt{t}\right] \\
&= -\beta\cdot\frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{4\beta^2} + (\Gamma_a\sigma_a+\Gamma_s\sigma_s)\sqrt{\frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{4\beta^2}} \\
&= -\frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{4\beta} + \frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{2\beta} \\
&= \frac{(\Gamma_a\sigma_a+\Gamma_s\sigma_s)^2}{2\beta}
\end{aligned}$$

Because the worst case queue length is the longest possible queue length under the conditions of the uncertainty set, we have $Q_s \leq Q_s^*$ for any arrival/departure process.

$$\lim_{s\to\infty} \frac{Q_s}{\sqrt{s}} \leq \lim_{s\to\infty} \frac{Q_s^*}{\sqrt{s}} \leq \frac{(\Gamma_a\sigma_a + \Gamma_s\sigma_s)^2}{2\beta}.$$

□

Proposition 4.2.3 has some interesting implications. Let us compare it with the bounds by Kingman and Bandi, rewritten using Little's law $\mathbb{E}[Q_s] = \lambda\mathbb{E}[W]$ in terms of the mean queue length:

$$\lim_{s\to\infty} \frac{Q_s}{\sqrt{s}} \leq \frac{(\Gamma_a\sigma_a + \Gamma_s\sigma_s)^2}{2\beta} \qquad \text{(Proposition 4.2.3)}$$

$$\mathbb{E}[Q_s] \leq \frac{\lambda^2\big(\sigma_a^2 + \sigma_s^2/s + (1/s - 1/s^2)\big)}{2(1-\rho)} \qquad \text{(Kingman)}$$

$$\mathbb{E}[Q_s] \leq \frac{\lambda^2}{4}\frac{(\Gamma_a + \Gamma_s/\sqrt{s})^2}{1-\rho} + s - \lambda E[B] \qquad \text{(Bandi)}$$

A direct comparison is hard, because the bounds are for different performance measures. Our bound is for the worst case queue length and not for the mean queue length. Kingman' bound has a stochastic nature, while the other two bounds are based on a robust setting with uncertainty parameters Γ_a and Γ_s. Finally, our bound is for the Halfin-Whitt regime while the other two are not. In the Halfin-Whitt regime $1-\rho$ scales like $\beta/\sqrt{s}$, so the denominators of the three bounds have the same functional dependence on $1-\rho$, which is characteristic for many classical queueing models.

Also compare the roles of σ_a^2 with Γ_a^2 and σ_s^2 with Γ_s^2. All bounds share a functional dependence on the sum of the variances of the interarrival and processing times. However, the uncertainty parameters Γ_a and Γ_s from Bandi et al. (2015) cannot be directly compared with the parameters Γ_a and Γ_s in Proposition 4.2.3, since they are based on different uncertainty sets. The new bound does not depend on s or λ except for the order $\sqrt{s}$, by which ρ_s is normalized. Apparently this choice of uncertainty set makes the result tractable in the Halfin-Whitt regime. Unfortunately the assumptions for the worst case are usually not satisfied, so we cannot extend our conclusions to the stochastic system automatically.

The uncertainty set excludes extremely short interarrival times and extremely long processing times. To compensate, the worst case assumes that all interarrival times are relatively short and all processing times are relatively long. The strictness of the uncertainty set is determined by a parameter Γ. For most distributions there will be some value for Γ for which the excluded extreme values almost never occur. Then the corresponding uncertainty set is reasonable. However, if the distribution has a very high or infinite variance (in the case of heavy-tailed distributions), there is no reasonable value for Γ and the bound does not work. That can be seen in the assumptions as well; the result requires finite variances and assumes the CLT to hold.

4.3 Open questions

The main shortcoming of the uncertainty sets described in this chapter is that an arbitrary arrival or departure process has a very low probability of being in the uncertainty set. At best we can say that a large majority of the arrivals and departures satisfy the assumptions. As a consequence the derived bound on the queue length does not necessarily hold in the stochastic system that assumes

a parameterized family of distributions for the arrival and service processes. This could be solved by finding a broader uncertainty set which contains an arbitrary arrival or departure process almost surely. The difficulty is that this uncertainty set needs to give a tractable optimization problem as well. Instead of the CLT one could use the Law of the Iterated Logarithm as inspiration for the uncertainty set, since Bertsimas et al. (2011) had some success with that.

Another open question is how to interpret robust results in a queueing context. How does it compare to probabilistic methods? Robust Optimization circumvents many problems with stochastic models, since it specifies which uncertainty is important. By creating an uncertainty set, one chooses to ignore part of the problem. Robust Optimization needs to show that this part can be ignored without changing the core behavior of the stochastic model. If you do not believe in the uncertainty set, the robust results are of little value. But if you can make a strong argument for the uncertainty set, Robust Optimization is a powerful new tool to analyze queueing networks.

Work needs to be done in showing that implications of a probabilistic model are just as strong as having the correct probabilistic model itself. For example, Proposition 4.2.3 shows a new bound on the queue length in the Halfin-Whitt regime. But one could suspect that in creating the uncertainty set the problem was simplified too much, and the bound is not representative for the behavior of the stochastic model. What is missing is a strong argument why this choice is valid, beyond pointing out that the bound has a functional form we would expect for this model.

The role of variability parameters like Γ_a and Γ_s remains somewhat vague as well. Why is one parameter sufficient for the whole arrival process? One would expect a significant difference between the variability of the time of the first arrival and a much later arrival. The CLT argument only works asymptotically. The value chosen for Γ seems arbitrary; the literature focuses more on the existence of this constant than on the actual value. That suffices when looking for qualitative insights, but the constants do matter for quantitative results.

5 Conclusion

In this thesis we have seen several examples of Robust Optimization in a queueing environment. In this chapter we discuss the lessons we can learn from Robust Optimization in this context. What makes it meaningful and valuable? What opportunities for future research are there?

5.1 Conclusion

Since Robust Optimization reduces the effect of uncertainty in model parameters, it works best if uncertainty matters. Small changes need to have a big impact. In queueing system that is often the case, especially if the system is in heavy traffic. When the load on the system is close to the maximum capacity, a small change in either the load or the capacity has a huge effect on performance characteristics such as average waiting time, probability of waiting and queue length. This sensitivity makes queueing models a good candidate for Robust Optimization.

In optimization, using nominal values can often lead to (severe) infeasibilities (Den Hertog, 2015). It means a violation of constraints which Robust Optimization can prevent. In a queueing context infeasibility also is a problem. It occurs when the system gets more arrivals on average than it can handle, violating the constraint that capacity needs to exceed arrival rate. Infeasibility is the worst state a queueing system can be in. But even being close to infeasibility is a huge problem. The goal is not only preventing infeasibility, but also staying away from it as much as possible. Robust Optimization is a suitable technique to do that.

The success of Robust Optimization depends greatly on the uncertainty set. Choosing the right uncertainty set can make or break the method. On on hand the uncertainty set needs to be chosen such that the resulting optimization problem is tractable, but on the other hand it needs to describe the system well. A common way to show tractability is by considering the dual optimization problem. In this thesis we used our knowledge about queueing models to simplify the uncertainty set. Even though there are infinitely many scenarios to consider, only a few of them are candidates for the worst case. Proposition 2.3.3 and Lemma 4.1.1 are examples of that.

Identifying the proper risks and characterizing them in an uncertainty set is not straightforward.

Not every risk can be mitigated by a robust approach. We saw some examples in the CA problem. The risk of overall higher traffic could not be reduced with robust capacities. The robust capacities are optimal for one worst case, when every arrival rate is underestimated. But when the deviations are a mix of over- and underestimation, the classic capacities work better. Moreover, there were multiple 'extreme' cases in the uncertainty set, and the robust capacities covered only one of them. Robust performed worse in the majority of cases and when it did work better, it was not the only 'extreme' case.

A form of risk that works better with Robust Optimization is uncertain routing. The total arrival rate is known, but we do not know how the arrival rates are divided over all servers. We showed that the worst case has a specific form. There is one bottleneck channel that receives all problems in the form of extra arrivals. Which channel is the bottleneck depends on the assigned capacities. Robust Optimization identifies bottleneck channels and gives them more capacity, without creating new bottlenecks. It finds a balance in the average delay across multiple possible scenarios. Therefore the robust capacities outperform the classic capacities for a wide range of possible deviations. But more importantly, they perform way better for the most occurring scenarios with extremely long delay.

Our numerical results showed how potent Robust Optimization can be. Reducing the probability of infeasibility by 75%, the probability of a high delay by 30% and making the standard deviation of the possible outcomes a factor 4 smaller, while performing slightly better on average as well. This was just one numerical example, but in general robust capacities give more reliable results and drastically reduce the occurrence of extreme outcomes. When there is more uncertainty about the model parameters, Robust Optimization can make an even bigger difference.

Are there also downsides to using robust capacities in the CA problem? Calculating them is a little bit more work. Instead of using the formula by Kleinrock, one needs to solve a relatively simple optimization problem. The problem is minimizing the maximum of $M(M-1)$ convex functions, where M is the number of servers. Many software programs can do that, we used Mathematica for example. That makes the robust capacities computationally tractable. Another concern is that you might be wrong about the starting assumptions. What if we overestimated uncertainty and the first predictions were right all along? It turns out that in that case the penalty for using robust capacities is very small, they perform almost as well as the optimal nonrobust capacities.

Another way to use Robust Optimization is replacing the stochastic nature of queueing models. Instead of describing interarrival and service times with probability distributions, they can take any value in an uncertainty set. This means excluding extreme cases outside of the uncertainty set. Probability laws like the CLT can help form the uncertainty set such that extreme cases have a very small probability of happening. This method shifts a stochastic model to an optimization problem and permits new ways to analyze the model. One advantage is analyzing models with general distributions, since only information about the mean and standard deviation is needed. Another advantage is that it provides a new perspective or angle to analyze queueing models.

5.1.1 Further research

The goal of this thesis was exploring the possibilities of Robust Optimization in a queueing context. Perhaps more important than finding the answers is formulating the right questions. There are a few interesting directions for further research. Robust Optimization has proven itself as an effective way to deal with uncertainty, and we know that in queueing networks uncertainty matters. The CA problem is just one example, but what is this like for queueing theory as a whole? How sensitive

are queueing models to errors in the parameters? Do queueing models describe practical problems accurately enough? Is a more conservative approach desirable? How well would robust queueing decisions work out in practice?

The questions above are aimed at improving our capability of implementing accurate and effective models in practice. But Robust Optimization also has potential for more theoretic results. It offers a radically different view on uncertainty. This view does not require knowledge about complete distribution functions and shifts the analysis to a different domain, namely optimization. By excluding unlikely extreme cases, the analysis can become easier. What advantages does a robust view on uncertainty offer? Are there problems in stochastic models that a robust approach can circumvent? When is the robust version of a model tractable?

A robust view on uncertainty also calls for a better understanding of the connection between probability and Robust Optimization. That leads to some deep questions about the ideas behind Robust Optimization. When is an uncertainty set meaningful? What does it represent? What is the connection between stochastic models and robust models? What is needed is not just proof that Robust Optimization works in a queueing context, but also a justification why it is a good assumption to view the world in uncertainty sets. Then it could get a place next to other methods to analyze queues.

As long as uncertainty plays a major role in queueing systems, Robust Optimization can form a valuable tool in improving and analyzing them. The potential is only limited by the amount of uncertainty, and queueing models have an abundance of uncertainty to deal with. Hopefully both research areas will profit from learning from each other in the coming years.

Bibliography

[1] C. Bandi, D. Bertsimas and N. Youssef, *Robust Queueing Theory*, Operations Research 63 (2015), No. 3: 676-700.

[2] A. Ben-Tal, L. El Ghaoui and A. Nemirovski, *Robust Optimization*, Princeton University Press (2009).

[3] A. Ben-Tal et al. *Robust solutions of optimization problems affected by uncertain probabilities*, Management Science 59 (2013), No. 2: 341-357.

[4] D. Bertsimas and X. V. Doan, *Robust and data-driven approaches to call centers*, European Journal of Operational Research 207 (2010), No. 2: 1072-1085.

[5] D. Bertsimas, D. Gamarnik and A. Rikun, *Performance analysis of queueing networks via robust optimization*, Operations Research 59 (2011), No. 2: 455–466.

[6] D. Bertsimas and M. Sim, *The price of robustness*, Operations Research 52 (2004), No. 1: 35-53.

[7] S. Borst, A. Mandelbaum and M. I. Reiman, *Dimensioning large call centers*, Operations research 52 (2004), No. 1: 17-34.

[8] D. Gamarnik and D. Goldberg, *Steady-state GI/GI/n queue in the Halfin-Whitt regime*, Annals of Applied Probability 23 (2013), No. 6: 2382-2419.

[9] N. Gans, H. Shen, et al., *Parametric stochastic programming models for call-center workforce scheduling*, Technical report (2009).

[10] S. Halfin and W. Whitt, *Heavy-traffic limits for queues with many exponential servers*, Operations research 29 (1981), No. 3: 567-588.

[11] D. den Hertog, *Practical Robust Optimization*, course notes, Tilburg University (2015).

[12] P. Jelenkovic, A. Mandelbaum and P. Momcilovic, *Heavy traffic limits for queues with many deterministic servers*, Queueing Systems 47 (2004), No. 1-2: 53-69.

[13] G. Jongbloed and G. Koole, *Managing uncertainty in call centers using Poisson mixtures*, Applied Stochastic Models in Business and Industry 17 (2001): 307-318.

[14] J. F. C. Kingman, *Inequalities in the theory of queues*, Journal of the Royal Statistical Society, Series B (Methodological) 32 (1970), No. 1: 102-110

[15] L. Kleinrock, *Queueing Systems, volume 2: computer applications*, New York: John Wiley & Sons, Inc. (1976).

[16] S. Liao, C. Van Delft and J.-P. Vial, *Distributionally robust workforce scheduling in call centres with uncertain arrival rates*, Optimization Methods and Software 28.3 (2013): 501-522.

[17] S. Liao, G. Koole, C. Van Delft and O. Jouini, *Staffing a call center with uncertain non-stationary arrival rate and flexibility*, OR spectrum 34 (2012), No. 3: 691-721.

[18] P. Nebres, *Renewal theory and its applications*, (2011).

[19] K. Postek, A. Ben-Tal, A., D. den Hertog and B. Melenberg, *Exact robust counterparts of ambiguous stochastic constraints under mean and dispersion information*, preprint (2015).

[20] L. Saino, I. Psaras and G. Pavlou, *Hash-routing schemes for information centric networking*, Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, ACM (2013), pp. 27-32.

[21] L. M. Wein, *Capacity allocation in generalized Jackson networks*, Operations Research Letters 8 (1989), No. 3: 143-146.

[22] E. B. Wilson and M. M. Hilferty, *The distribution of chi-square*, Proceedings of the National Academy of Sciences of the United States of America 17 (1931), No. 12: 684-688.

[23] J. Zan, J. J. Hasenbein and D. P. Morton, *Staffing Large Service Systems Under Arrival-rate Uncertainty*, arXiv preprint arXiv (2013): 1304.6701.

A Appendix

A.1 Construction of confidence intervals

In this thesis we used several 95%-confidence intervals. Here follows a description of how we calculated them. We used a sample of size $N = 10^6$, because that gave accurate enough results within a reasonable simulation time. Part of the sample yields an infeasible system for either the nonrobust or the robust capacities. In that case we cannot compare the mean or the standard deviation, since the average delay is infinite. Therefore we discarded all cases of infeasibility and computed the mean and standard deviation for the slightly smaller sample of size n.

Our estimator for the mean average delay is the sample mean

$$\bar{T} = \frac{1}{n} \sum_{i=1}^{n} T_i.$$

The estimator for the standard deviation is the square root of the sample variance

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (T_i - \bar{T})^2}.$$

These estimators can be calculated recursively to avoid the storage of the entire sample in memory.

$$\bar{T}(i) = \frac{i-1}{i} \cdot \bar{T}(i-1) + \frac{1}{i} \cdot T_i$$

and

$$S^2(i) = \frac{n-2}{n-1} \cdot S^2(i-1) + \frac{1}{i} \cdot \left(T_i - \bar{T}(i-1)\right)^2$$

with $\bar{T}(1) = T_1$ and $S^2(1) = 0$.

By the Central Limit Theorem the mean average delay is distributed approximately normal, so the corresponding 95%-confidence interval is given by

$$\left[\bar{T} - 1.96 \cdot \frac{S}{\sqrt{n}},\ \ \bar{T} + 1.96 \cdot \frac{S}{\sqrt{n}}\right]$$

where 1.96 is the quantile $z_{0.025}$ of the normal distribution. Note that since n is large, the normal quantile and the t-distribution quantile are almost equal. If T_i approximately follow a normal distribution, $\sum_{i=1}^{n}(T_i - \bar{T})^2$ is approximately χ^2-distributed. The corresponding 95%-confidence interval for the variance is given by

$$\left[\frac{(n-1)S^2}{\chi^2_{n-1;0.025}},\ \ \frac{(n-1)S^2}{\chi^2_{n-1;0.975}}\right].$$

By taking the square root of these values we get the standard deviation. The quantiles $\chi^2_{n-1;0.025}$ and $\chi^2_{n-1;0.975}$ were calculated with the approximation by Wilson and Hilferty (1931),

$$\chi^2_{n-1;0.025} = (n-1) \cdot \left(1.96\sqrt{\frac{2}{9(n-1)}} + 1 - \frac{2}{9(n-1)}\right)^3$$
$$\chi^2_{n-1;0.975} = (n-1) \cdot \left(-1.96\sqrt{\frac{2}{9(n-1)}} + 1 - \frac{2}{9(n-1)}\right)^3$$

Finally we are interested in the proportion of the sample that is infeasible. The estimator for that is the fraction of the sample with infeasibility,

$$\hat{p} = \frac{N-n}{N}.$$

By the Central Limit Theorem this proportion is distributed approximately normal, with a variance of $\hat{p}(1-\hat{p})$. The corresponding 95%-confidence interval is given by

$$\left[\hat{p} - 1.96 \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{N}},\ \ \hat{p} + 1.96 \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{N}}\right].$$

The proportion of the sample with $T > 35$ follows the same approach with estimator

$$\hat{q} = \frac{\#\text{ times } T > 35}{n}$$

and 95%-confidence interval

$$\left[\hat{q} - 1.96 \cdot \sqrt{\frac{\hat{q}(1-\hat{q})}{n}},\ \ \hat{q} + 1.96 \cdot \sqrt{\frac{\hat{q}(1-\hat{q})}{n}}\right].$$