Robust Appointment Scheduling with General Convex Uncertainty Sets

Judith Brugman[*], Dick den Hertog[†], Johan S.H. van Leeuwaarden[‡]

Abstract

The Appointment Scheduling Problem (ASP) involves scheduling a finite number of customers with uncertain service times, served consecutively by a single server, aiming to minimize the weighted costs of waiting time, idle time, and overtime. Previous studies using stochastic programming were limited to small instances. We introduce a Robust Optimization (RO) approach that considers service times within a specified uncertainty set and aims to minimize the worst-case costs, which requires maximizing a convex function. Combining advanced methods from Robust Convex Optimization, such as the Reformulation-Perspectification Technique (RPT) and the cutting-set approach, leads to an exact solution procedure for determining optimal schedules. Our robust framework for ASP is designed to manage large instances and accommodates general convex uncertainty sets. Based on extensive numerical experiments for polyhedral and ellipsoidal uncertainty sets, and problem instances with over 50 customers, we reveal intricate interactions between the uncertainty of service times, the cost function, and the structural characteristics of optimal schedules.

Keywords: Appointment Scheduling, Robust Convex Optimization, Convex Uncertainty Sets, Reformulation-Perspectification Technique (RPT), Computational Tractability.

1 Introduction

The efficient management of appointments is increasingly crucial in a world where online booking systems are becoming more popular and customer expectations for superior service remain high. Efficient appointment scheduling is essential not only for optimizing resource utilization but also for minimizing delays in the system, thereby enhancing overall operational efficiency and service quality across various sections, such as healthcare, transportation and customer service. These sectors rely on allocating appointments to available resources – such as time slots, personnel or equipment – while considering sector-specific constraints and preferences. The complexity of scheduling is heightened by uncertainties such as the unknown duration of appointments. In healthcare, effective appointment scheduling is vital for ensuring timely access to medical services, reducing patient waiting times, and optimizing the use of expensive equipment and highly-skilled personnel. Given the rising healthcare costs and growing demand for services (Hulshof et al. 2012), improving scheduling has been a key focus worldwide. The unpredictable nature of appointment durations,

[*]Department of Econometrics and Operations Research, Tilburg University, j.m.brugman@uvt.nl
[†]Amsterdam Business School, University of Amsterdam, d.denhertog@uva.nl
[‡]Department of Econometrics and Operations Research, Tilburg University, j.s.h.vanleeuwaarden@uvt.nl

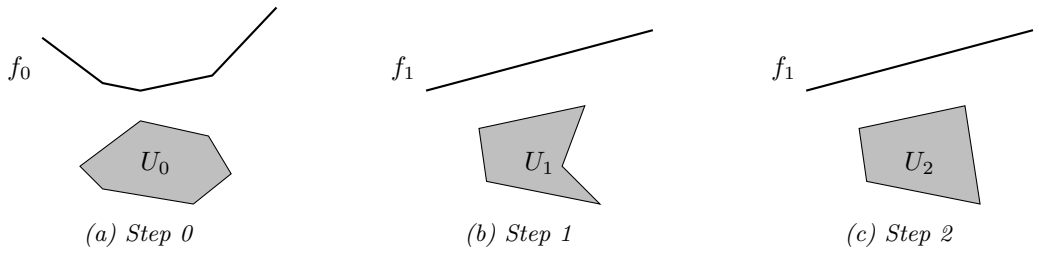
from medical consultations to surgical procedures, can lead to extended waits and idle times if not properly managed. A comprehensive discussion of these challenges is provided in Gupta and Denton (2008).

Classic setting. The literature on optimal appointment scheduling for systems with uncertain services is deeply rooted in queueing theory (Bailey 1952, Mercer 1960, Wang 1993, Kaandorp and Koole 2007, Green and Savin 2008, Hassin and Mendel 2008, Zacharias and Pinedo 2014, Liu 2016, Kuiper et al. 2017, Zacharias and Yunes 2020, Zhou et al. 2021). In particular, the Appointment Scheduling Problem (ASP), introduced in the work of Bailey (1952), is a well-studied problem that has received considerable attention from researchers and practitioners. The classic ASP considers a finite number of customers with uncertain service times, which are served consecutively by a single server during a fixed time horizon. The primary objective is to devise a schedule that assigns a time slot to each customer in such a way that it minimizes the total weighted costs of customers' waiting time and server's idle and overtime. These scheduling decisions must be made well ahead of time, despite incomplete knowledge of service durations, which adds high complexity to the problem. Allocating overly short time slots can cause delays in subsequent appointments and increased customer wait times, reducing service quality. Conversely, overly long slots may lead to periods of underutilization and potentially unnecessary overtime, indicating system inefficiency. The classic ASP is thus a stochastic optimization problem balancing these two opposing goals. It typically requires distributions for a finite number of service times, a linear cost function penalizing both expected wait and idle time, and queue dynamics describing waiting and idle times as functions of the appointment and service times.

Limitations of stochastic optimization. The classic ASP thus searches for optimal ways of scheduling a finite number of customers by solving a stochastic optimization problem. Extensions of the ASP include considerations for no-shows, multiple servers, and customer preferences. For comprehensive literature reviews, see Cayirli and Veral (2003), Gupta and Denton (2008), Ahmadi-Javid et al. (2017), and Ala and Chen (2022), which focus primarily on healthcare services. Various methods, including sample average approximation (SAA), quasi-gradient methods, and sequential bounding approaches, have been employed to solve the ASP (Robinson and Chen 2003, Denton and Gupta 2003b, Shapiro et al. 2009, Birge and Louveaux 2011, Begen et al. 2012). SAA, for instance, reformulates the problem into a linear program for a given scenario of service times and considers average costs over a large number of sampled scenarios. These traditional methods typically operate under the assumption of precise input data, including a full specification of the service time distribution. As an alternative, Distributionally Robust Optimization (DRO) only requires partial information of the distributions, including metrics like the mean, moments, and range. DRO then finds schedules that minimize the worst-case expected costs among all possible distributions that comply with this partial information. Several papers solve some version of the distributionally robust ASP (Kong et al. 2013, Mak et al. 2015, Qi 2016, Bertsimas et al. 2018, Padmanabhan et al. 2021, van Eekelen et al. 2024). Despite the effectiveness of SAA and related DRO techniques in developing schedules while facing uncertainties, they are not without limitations. Both approaches encounter what is commonly referred to as the "curse of dimensionality", since the complexity of the problem increases exponentially with the number of customers. Consequently, existing literature typically restricts the application of these methods to problems with at most 20 customers, highlighting a significant challenge in extending these models to larger-scale problems. These severe computational restrictions motivated us to develop new methods based on robust optimization, ensuring efficiency and scalability for larger problem instances.

Robust optimization. In recent years, Robust Optimization (RO) has emerged as a promising paradigm for addressing complex optimization problems, notable for its computational tractability and high modeling power. The concept of RO was first mentioned in the early 1970s in Soyster (1973), and has gained significant attention from researchers particularly since the 2000s. It has been used in many applications, such as energy, logistics, finance and health care; for literature reviews we refer to Beyer and Sendhoff (2007) and Gabrel et al. (2014). Furthermore, the foundational principles for different categories of RO problems are comprehensively detailed in the books Ben-Tal et al. (2009) and Bertsimas and den Hertog (2022). Unlike traditional stochastic methods, RO addresses uncertainty by defining an uncertainty set in which the uncertain parameters are assumed to reside. Choosing the type of uncertainty, often based on a box, ellipsoidal or polyhedral set, is delicate, therefore a practical guide is given in Gorissen et al. (2015). Note that RO does not rely on probability distributions, but the uncertainty sets can be designed building upon asymptotic results of probability theory, such as the central limit theorem (Bandi and Bertsimas 2012). The objective of RO is to identify solutions that are resilient, performing optimally in the worst-case scenarios within this uncertainty set. The essence of RO lies in its ability to transform problems with robust constraints, which must hold for every possible value within the set, into tractable deterministic equivalents known as robust counterparts. This transformation allows RO to efficiently handle larger-scale problem instances and potentially address the challenges of high dimensionality.

Robust optimization for ASP. Although there have been notable attempts in applying RO to the ASP, these have often necessitated restrictive choices regarding uncertainty sets to preserve tractability. Mittal et al. (2014) explore the stochastic linear ASP model with service times confined to specific intervals, known as box uncertainty. By recognizing that the worst-case scenario consistently selects one of the extreme points of these intervals, they manage to derive closed-form optimal schedules that effectively balance the maximum underage and overage cost. Expanding on this idea, Schulz and Udwani (2019) allow for heterogeneous underage costs, yielding approximate solutions instead. Further developments by Bauerhenne et al. (2024) also embrace box uncertainty, including no-show probabilities, to minimize idle and overtime costs while ensuring waiting time guarantees. They tackle this by analytically determining worst-case scenarios for the service times and no-shows, and subsequently solving a mixed-integer linear program. Bandi and Gupta (2020) examine surgical scheduling within hospitals using a polyhedral uncertainty set. Their model manages staffing and sequencing constraints to optimize surgery schedules, but a critical analysis reveals two significant misconceptions in their approach, suggesting they inadvertently simplify the problem to box uncertainty. Note that initially, uncertainty was only present in the objective function, affecting the waiting times, idle times, and overtime. However, their implementation considers two sets of robust constraints involving the durations, meaning it spreads the uncertainty across multiple inequalities. In general, even though two nominal problems are equivalent, their resulting robust problems, as explained in Chapter 12 of Bertsimas and den Hertog (2022), could be different. Indeed, that is also the case for the robust ASP, due to the non-disjoint nature of the uncertainty. This modification becomes apparent when recognizing that each uncertain duration is now involved in several constraints, so each time the worst-case can be selected differently, and being the only uncertain parameter in the constraint, it can fluctuate between its minimum and maximum value in the polyhedron (independent of the other parameters). This contradicts the principles of polyhedral uncertainty, where the possible scenarios for each parameter are interdependent. Further complicating the issue, when their work introduces the dual variables to obtain the robust counterpart, these are uniformly chosen over all constraints, imposing an unnecessary restriction. These critical points underline that the approach taken by Bandi and Gupta (2020) aligns more with the



(a) Step 0 *(b) Step 1* *(c) Step 2*

Figure 1: Visualization of the steps in the method for robust convex optimization

principles of box uncertainty, as previously explored by Mittal et al. (2014), instead of polyhedral uncertainty. This clearly highlights the challenges in applying RO to the ASP when dealing with more complex uncertainty sets. We will address these challenges by implementing and extending state-of-the-art RO methods that are suitable for general convex uncertainty sets.

Robust convex optimization. We conclude that the primary challenge in developing a computationally tractable RO framework for ASP stems from the convex nature of the cost objective, with respect to the uncertain parameters. In contrast to robust concave optimization, for which tractable equivalent reformulations can be derived as demonstrated by Ben-Tal et al. (2015), robust convex optimization presents greater complexity, making such reformulations generally out of reach. Note that with robust convex optimization, we imply both convexity in the uncertain parameters and variables (contrary to Ben-Tal et al. (1998), that can only deal with convexity in the decision variables), as well as a convex uncertainty set. The complexity largely comes from the initial step that involves identifying the worst-case scenario within the uncertainty set, essentially maximizing a convex function. Finding a global maximum is known to be hard in this case. For example, for polyhedral uncertainty we could exhaustively explore all corner points of the uncertainty set, with the understanding that the function reaches its maximum at one of these points, but this quickly becomes computationally too expensive. Overcoming this challenge involves not only advanced mathematical techniques but also innovative reformulations to ensure scalability and general applicability. Nevertheless, recent advancements by Bertsimas et al. (2023) give rise to new opportunities, unifying and extending the existing approaches from the literature for robust convex optimization. This novel RO approach consists of four steps:

(0) The starting point is a robust convex optimization problem as in Figure 1a.

(1) Transform this robust convex problem into an equivalent bilinear problem with a nonconvex uncertainty set, see Figure 1b, making the function computationally tractable but complicating the uncertainty set.

(2) Use the Reformulation-Perspectification Technique (RPT) from Zhen et al. (2021) to generate a convex outer approximation of this nonconvex set; see Figure 1c, simplifying it into a robust concave optimization problem.

(3) Solve this robust concave problem using either the support function or via the dual problem, as demonstrated in the work of Ben-Tal et al. (2015), yielding a conservative approximation for the optimal objective value and thereby establishing an upper bound in the context of minimization problems.

Additionally, the work of Bertsimas et al. (2023) proposes a procedure to derive a lower bound, by intelligently selecting a finite scenario set from the uncertainty set and optimizing the corresponding linear program exclusively over this subset. While promising, several challenges persist when applying this framework to obtain lower and upper bounds. Given its generality, each step must be carefully tailored to the specific objective and uncertainty set in question. This task will prove to be very complex, as it requires e.g. fitting the problem setting in the notation of the framework, applying RPT to convex uncertainty sets, and deriving support functions. Moreover, obtaining exact robust solutions remains critical in many applications, which is where we extend their work by adopting a cutting-set approach.

Structural properties of optimal schedules. As mentioned, a key methodological motivation for this study is to design a computationally efficient robust optimization framework, capable of handling general characterizations of uncertainty and large problem instances. Equally important is our goal to assess the interplay between uncertainty and optimal schedules, providing valuable guidance for practitioners looking to implement these robust scheduling solutions. Earlier studies revealed several structural features of optimal schedules. For i.i.d. service times with known distributions, studies by Wang (1993), Denton and Gupta (2003a), Robinson and Chen (2003), and Kuiper and Lee (2022), using various stochastic programs and optimization methods observed a typical 'dome-shaped' pattern. In these schedules, time slots first increase and then decrease, allocating more time for customers in the middle of the day and less at the beginning and end. This can be explained by non-stationary queueing theory: early delays cause further delays, necessitating longer appointment slots early on. However, longer slots risk idleness, so shorter slots at the start and end can prevent idling and overtime. Consequently, longer slots are best placed in the middle, balancing idling and waiting times. This dome-shaped pattern is a common feature in ASP literature assuming independent service times. In contrast, literature on distributionally robust ASP with partially known service time distributions often finds 'decreasing' schedules, where slot lengths shorten over time. This is due to worst-case scenarios grouping long service times early on, requiring more time allocation for initial jobs to mitigate delays. This decreasing schedule contrasts with Bailey's scheduling rule, which uses increasing schedules to balance idle and waiting times effectively. Recent research by van Eekelen et al. (2024) using DRO methods that assume independent service times found structures resembling Bailey's rule, with minimal early slots. These optimal schedules shift from dome-shaped to increasing when overtime costs are high. Additionally, Benjaafar et al. (2023) found optimal increasing schedules when incorporating service level constraints.

This paper's contributions. In this paper, we develop a general comprehensive framework for appointment scheduling to determine robust schedules under service time uncertainty. Our key contributions can be divided into two sets:

- *Methodological innovations:* The first set of contributions revolves around introducing a new framework for studying the robust ASP, by applying the method of Bertsimas et al. (2023) for robust convex optimization to this particular problem to obtain lower and upper bounds, and enhancing it with a cutting-set approach to obtain exact solutions. Our framework is designed to determine schedules that minimize the worst-case weighted sum of waiting time, idle time, and overtime, while accounting for uncertainty in service durations. It is capable of handling larger-scale instances and can provide optimal robust schedules for problems involving more than 50 customers. The framework is applicable for general convex uncertainty sets, and is illustrated through implementation for both polyhedral and ellipsoidal uncertainty.

- *Practical implications and insights:* The second set of contributions is of a more practical nature and explores the implications of uncertainty within the ASP and its impact on scheduling decisions. Our comprehensive analysis of the ASP highlights the intricate interactions between the type and size of uncertainty of service times, the cost function, and the structural characteristics of optimal schedules and worst-case scenarios. We demonstrate for budget and ellipsoidal uncertainty in particular, that a diverse array of optimal robust schedules can emerge and provide insights into the conditions under which schedules are decreasing, dome-shaped or increasing.

Outline. The remainder of this paper is structured as follows. Section 2 formally outlines the robust Appointment Scheduling Problem. In Section 3, we introduce the general framework developed to solve the robust ASP, by discussing subsequently the generation of the upper bound, lower bound and exact solution. The application of this framework to polyhedral and ellipsoidal uncertainty sets is demonstrated in Sections 4.1 and 4.2, respectively. Section 5 analyzes the optimal robust schedules and worst-case scenarios that emerge from specific settings for budget uncertainty and ellipsoidal uncertainty, together with some computational results, including cost bounds and running times. Finally, Section 6 concludes the paper with a summary of our findings and an outlook on promising directions for future research.

2 Robust appointment scheduling

Consider a service system with a single server that needs to serve N customers within a fixed planning horizon T. Customers arrive punctually and are served in order of arrival. Let x_n denote the service time of customer n, collectively represented by the vector $\mathbf{x} = (x_1, \ldots, x_N)$. Prior to service, $\mathbf{x}$ is unknown and is constrained within a predefined convex uncertainty set $\mathcal{U}$, i.e. $\mathbf{x} \in \mathcal{U}$. This set is defined by inequalities $g_i(\mathbf{x}) \leq 0$ for $i = 1, \ldots, K$, where each g_i is convex. The goal is to determine the optimal schedule $\mathbf{s} = (s_1, \ldots, s_N)$, with s_n denoting the allocated time slot for customer n. The appointment time for the n-th customer is thus $a_n = \sum_{i=1}^{n-1} s_i$. The schedule must ensure that each s_n is nonnegative and the total time does not exceed T (equivalently, we can condition that $\sum_{n=1}^{N} s_n = T$, as w.l.o.g. we can assign the remaining time to the last customer). This specifies the feasible schedule set as $\mathcal{S} = \{\mathbf{s} : \mathbf{s} \geq 0, \sum_{n=1}^{N} s_n = T\}$.

The cost function $f(\mathbf{s}, \mathbf{x})$, aimed at minimization, depends on both the schedule $\mathbf{s}$ and the service durations $\mathbf{x}$. It includes the weighted sum of total waiting time W, server's idle time I, and overtime O, with respective weights c_W, c_I and c_O. Unlike stochastic models where typically only idle time or overtime is included as their expectations differ only by a constant (Proposition 1 in Denton and Gupta (2003b)), our robust setting necessitates considering all three components explicitly.

Let w_n denote the waiting time of customer n, so that the total waiting time is $W = \sum_{n=1}^{N} w_n$. These waiting times can be calculated recursively starting with $w_1 = 0$ and for $n = 2, \ldots, N$ using $w_n = (w_{n-1} + x_{n-1} - s_{n-1})^+$, leading to $w_n = \max_{1 \leq l \leq n-1} \left\{0, \sum_{j=n-l}^{n-1} (x_j - s_j)\right\}$. The total waiting time W is the sum over all customer waiting times. The total idle time I reflects the gaps between consecutive services and can be calculated as the difference between the starting time of the last appointment $(w_N + \sum_{j=1}^{N-1} s_j)$ and the cumulative service time of all customers before that $(\sum_{j=1}^{N-1} x_j)$. Lastly, the overtime O is calculated as the excess working time beyond T. This can be conceptualized as the waiting time of a last hypothetical customer $N + 1$ scheduled at time T.

Hence,

$$
\begin{aligned}
W = &\sum_{n=1}^{N} \max_{1\le l\le n-1} \Big\{0, \sum_{j=n-l}^{n-1} (x_j - s_j)\Big\} \\
I = &\max_{1\le l\le N-1} \Big\{0, \sum_{j=1}^{l} (s_j - x_j)\Big\} \\
O = &\max_{1\le l\le N} \Big\{0, \sum_{j=(N+1)-l}^{N} (x_j - s_j)\Big\}
\end{aligned}
$$

Combining these expressions, we define the cost objective of the robust ASP as

$$f(\mathbf{s}, \mathbf{x}) = c_W W + c_I I + c_O O. \tag{1}$$

Note that this cost function is the sum of the maximum of linear functions, and is therefore convex with respect to the uncertain parameters x_n and the decision variables s_n.

Thus, the goal of the robust ASP is to identify a feasible schedule that minimizes the worst-case costs across all potential outcomes of the service times within the uncertainty set, which can be restated using epigraph notation:

$$\min_{\mathbf{s}\in S} \max_{\mathbf{x}\in\mathcal{U}} f(\mathbf{s}, \mathbf{x}) \iff \begin{array}{ll} \min\limits_{\mathbf{s}\in\mathcal{S},\, \tau} & \tau \\ \text{s.t.} & f(\mathbf{s}, \mathbf{x}) \le \tau \ \ \forall \mathbf{x} \in \mathcal{U} \end{array} \tag{2}$$

As extensively discussed in Section 1, this robust ASP, including a robust constraint that is convex in the uncertain parameters, is very difficult to solve.

3 General robust framework

This section proposes a general robust framework for the robust ASP as described in (2). This framework provides upper and lower bounds on worst-case costs, and ultimately identifies an optimal robust schedule.

3.1 Upper bound

To derive an upper bound, we reformulate (2) into a tractable approximate problem. This involves modifying the robust constraint by enlarging the uncertainty set, following the methodology of Bertsimas et al. (2023) for handling robust convex constraints. We outline the three primary steps of their framework (visualized in Figure 1), and tailor them to the robust ASP constraint.

Step 1: Exact bilinear reformulation

We begin by reformatting the robust ASP constraint in (2) to the robust convex optimization framework. We identify the sum-of-max of linear (SML) structure in the cost function in (1), where each of the $N + 1$ terms represents the maximum of a set of linear functions with indices denoted by the set I_n. The function can be represented as:

$$f(\mathbf{s}, \mathbf{x}) = h(\mathbf{y}) = \sum_{n=1}^{N+1} \max_{l\in I_n} y_{nl}. \tag{3}$$

In this formulation, the terms $y_{nl} = \mathbf{a}_{nl}^T\mathbf{x} + b_{nl}(\mathbf{s})$ are affine, defined by constant vectors $\mathbf{a}_{nl}$ and functions $b_{nl}(\mathbf{s})$ depending linearly on decision variables $\mathbf{s}$. This structural information is encapsulated in the matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ and avector function $\mathbf{b}(\mathbf{s}) \in \mathbb{R}^M$, where $M = \mathcal{O}(N^2)$ reflects the total number of linear terms. Detailed constructions of $\mathbf{A}$ and $\mathbf{b}(\mathbf{s})$, including the relationship $\mathbf{b}(\mathbf{s}) = -\mathbf{A}\mathbf{s}$, are further elaborated in Appendix B.1, with an illustrative example for clarity.

The reformatted ASP constraint now complies with the general robust convex constraint in Bertsimas et al. (2023):

$$h(\mathbf{A}(\mathbf{s})\mathbf{x} + \mathbf{b}(\mathbf{s})) \leq 0 \ \forall \mathbf{x} \in \mathcal{U}, \tag{4}$$

where h (the SML function) is a proper, closed and convex function, the functions $\mathbf{A}(\mathbf{s})(= \mathbf{A})$ and $\mathbf{b}(\mathbf{s})(= -\mathbf{A}\mathbf{s})$ are affine in $\mathbf{s}$, and $\mathcal{U}$ is a convex set.

Now, let h^* and h^{**} denote the first and second convex conjugate of function h, respectively. Utilizing the property that $h = h^{**}$ for closed and convex functions, we can take the supremum of $h^{**}(\mathbf{A}(\mathbf{s})\mathbf{x} + \mathbf{b}(\mathbf{s}))$ over $\mathbf{x} \in \mathcal{U}$. By rearranging and substituting the resulting terms, Proposition 5 in Bertsimas et al. (2023) demonstrates that the general constraint in (4) is equivalent to

$$\sup_{(\mathbf{w},w_0,\mathbf{x},\mathbf{V},\mathbf{v}_0)\in\Theta} \left\{\mathrm{Tr}(\mathbf{A}(\mathbf{s})^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} - w_0\right\} \leq 0, \tag{5}$$

where Tr denotes the trace of the matrix. The corresponding uncertainty set is defined as

$$\Theta = \{(\mathbf{w}, w_0, \mathbf{x}, \mathbf{V}, \mathbf{v}_0) \mid \mathbf{x} \in \mathcal{U},\ \mathbf{w} \in \mathrm{dom}\ h^*,\ h^*(\mathbf{w}) \leq w_0,\ \mathbf{V} = \mathbf{w}\mathbf{x}^T,\ \mathbf{v}_0 = w_0\mathbf{x}\},$$

which introduces new uncertain parameters $\mathbf{w} \in \mathbb{R}^M$ and $w_0 \in \mathbb{R}$, along with their product terms $\mathbf{V} \in \mathbb{R}^{M\times N}$ and $\mathbf{v}_0 \in \mathbb{R}^N$ multiplying by $\mathbf{x}$.

For the robust ASP where $h(\mathbf{y})$ represents the SML function, the convex conjugate is $h^*(\mathbf{w}) = 0$ with domain $\mathrm{dom}\ h^* = \left\{\mathbf{w} \mid \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\right\}$ (proof in Appendix A.1). Here, matrix $\mathbf{H} \in \{0,1\}^{M\times(N+1)}$ is used, where each entry h_{mn} corresponds to whether linear function m belongs to the maximum operator n, with $h_{mn} = 1$ for $m \in I_n$ and 0 otherwise. Considering w_0 appears negatively in the supremum and the constraint is simplified to $w_0 \geq 0$, the optimal solution is $w_0 = 0$. Consequently, also $\mathbf{v}_0 = w_0\mathbf{x} = \mathbf{0}$, allowing us to omit these two variables.

So, the reformulated constraint in (5) for the ASP is

$$\sup_{(\mathbf{w},\mathbf{x},\mathbf{V})\in\Theta} \left\{\mathrm{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w}\right\} \leq \tau, \tag{6}$$

and the uncertainty set is simplified to

$$\Theta = \left\{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \mathbf{x} \in \mathcal{U},\ \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1},\ \mathbf{V} = \mathbf{w}\mathbf{x}^T\right\}.$$

Hence, the resulting robust constraint is linear in terms of both $\mathbf{x}$ and $\mathbf{s}$, yet now the uncertainty set is nonconvex due to the bilinear equality constraints.

Step 2: Convex outer approximation

In the second step, we create a safe approximation of this robust constraint by taking a convex outer approximation of the uncertainty set. Initially, we define a preliminary uncertainty set Θ_0, by excluding the nonconvex bilinear constraints $\mathbf{V} = \mathbf{w}\mathbf{x}^T$ from Θ:

$$\Theta_0 = \left\{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \mathbf{x} \in \mathcal{U},\ \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\right\}.$$

Using Θ_0 in (6) yields an upper bound by safely expanding the original set, as $\Theta \subset \Theta_0$. However, this approximation might not be tight due to the removal of the bilinear equality constraints, resulting in a significant loss of information.

To enhance this approximation while preserving convexity, we then apply the Reformulation-Perspectification Technique (RPT) from Zhen et al. (2021), introducing additional linear constraints that recapture some lost information. This technique involves multiplying each constraint in $\mathcal{U}$, $g_i(\mathbf{x}) \leq 0$ for $i = 1, \ldots, K$, with the non-negativity constraints $w_j \geq 0$ for $j = 1, \ldots, M$. From this nonconvex result, the RPT produces a convex constraint by considering the perspective function and replacing the terms $\mathbf{V} = \mathbf{w}\mathbf{x}^T$. The exact implementation of the RPT depends on the form of this function $g_i(\mathbf{x})$. Furthermore, we multiply the parameter $\mathbf{x}$ directly with the constraint $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ and perform the same substitution. Hence, this yields the following convex constraints:

$$\begin{aligned} (w_j \geq 0) \times \ (g_i(\mathbf{x}) \leq 0) \qquad &\rightarrow \quad w_j g_i(\mathbf{x}) \leq 0 \quad \forall i, j \\ (\mathbf{H}^T\mathbf{w} = \mathbf{1}) \ \times \ \mathbf{x} \qquad &\rightarrow \quad \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T. \end{aligned}$$

This leads to an improved convex uncertainty set Θ_1, integrating Θ_0 with the new constraints

$$\Theta_1 = \left\{ (\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid (\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_0, \ w_j g_i(\mathbf{x}) \leq 0 \, \forall i, j, \ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T \right\}.$$

Employing Θ_1 in (6) allows for a tighter safe approximation than using Θ_0, as $\Theta \subseteq \Theta_1 \subseteq \Theta_0$:

$$\sup_{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_1} \operatorname{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \leq \tau. \tag{7}$$

So, the second step results in an approximate linear constraint with a convex uncertainty set.

Step 3: Solving robust linear problem

Incorporating the robust constraint (7) into (2) leads to the following robust problem:

$$\begin{aligned} \min_{\mathbf{s} \in \mathcal{S}, \, \tau} \ & \tau \\ \text{s.t. } & \operatorname{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \leq \tau \quad \forall (\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_1. \end{aligned} \tag{8}$$

Note that this problem is linear and the uncertainty set is convex, therefore there are two simple non-iterative approaches: computing the support function of the uncertainty set (Ben-Tal et al. 2015) and deriving the dual formulation of the problem (Gorissen and den Hertog 2013).

Support function: To solve the problem using the support function, we first consolidate all decision variables into the vector $\mathbf{z} = (\mathbf{w}; \mathbf{x}; \mathbf{v}_1; \ldots; \mathbf{v}_N)$, where $\mathbf{v}_n$ is the n-th column of $\mathbf{V}$. We define vector $\mathbf{y}(\mathbf{s}) = \big[\mathbf{b}(\mathbf{s}); \ \mathbf{0}; \ \mathbf{a}_1; \ \cdots \ ; \ \mathbf{a}_N\big]$, allowing us to reformulate the robust constraint as

$$\sup_{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_1} \left\{ \operatorname{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \right\} \leq \tau \iff \sup_{\mathbf{z} \in \Theta_1} \mathbf{y}(\mathbf{s})^T\mathbf{z} \leq \tau \iff \delta^*(\mathbf{y}(\mathbf{s}) \mid \Theta_1) \leq \tau,$$

where δ^* is the support function of Θ_1 (the convex conjugate of the indicator function). The simplified version of the upper bound problem in (8) is therefore

$$\begin{aligned} \min_{\mathbf{s} \in \mathcal{S}, \, \tau} \ & \tau \\ \text{s.t. } & \delta^*(\mathbf{y}(\mathbf{s}) \mid \Theta_1) \leq \tau. \end{aligned} \tag{9}$$

This model can be easily solved using standard solvers, its complexity depending on the structure of δ^*, which is further detailed in Bertsimas and den Hertog (2022) for various uncertainty sets.

Dual problem: Alternatively, we can solve the robust problem using its dual formulation. We begin by ensuring that the optimization variables $\mathbf{s}$ and τ are explicitly defined in the problem, incorporating that $\mathbf{b}(\mathbf{s}) = -\mathbf{A}\mathbf{s}$. We also introduce a constant variable $\phi = 1$, so that the uncertainty is only in the left-hand side of the first constraint. The primal formulation is structured as

$$\begin{aligned} \max_{\mathbf{s},\tau,\rho} \quad & -\tau \\ \text{s.t.} \quad & -\mathbf{w}^T\mathbf{A}\mathbf{s} - \tau + \rho\,\mathrm{Tr}(\mathbf{A}^T\mathbf{V}) \leq 0 \;\forall (\mathbf{w},\mathbf{x},\mathbf{V}) \in \Theta_1 \\ & \mathbf{1}^T\mathbf{s} \leq T \\ & \phi = 1 \\ & \mathbf{s} \geq 0. \end{aligned} \tag{10}$$

Beck and Ben-Tal (2009) demonstrate that the dual of a robust problem can be derived by dualizing the deterministic problem and then substituting the $\forall$-quantifier in the primal with an $\exists$-quantifier in the dual. So, to obtain the dual problem, we introduce dual variables y_1, y_2 and y_3 corresponding to the primal constraints. Quickly identifying that $y_1 = 1$ and $y_3 = -\mathrm{Tr}(\mathbf{A}^T\mathbf{V})$, we can eliminate these variables and focus on $y_2 = y$. This leads to the alternative version of (8):

$$\begin{aligned} \min_{y,\mathbf{w},\mathbf{x},\mathbf{V}} \quad & Ty - \mathrm{Tr}(\mathbf{A}^T\mathbf{V}) \\ \text{s.t.} \quad & -\mathbf{A}^T\mathbf{w} + \mathbf{1}y \geq 0 \\ & y \geq 0 \\ & \mathbf{x} \in \mathcal{U} \\ & \mathbf{w} \geq 0 \\ & \mathbf{H}^T\mathbf{w} = \mathbf{1} \\ & w_j g_i(\mathbf{x}) \leq 0 \quad \forall i = 1,\ldots,K, j = 1,\ldots,M \\ & \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T. \end{aligned} \tag{11}$$

This problem is readily solved by standard optimization solvers. The complexity largely depends on the structure of the functions g_i', and the number of variables and constraints increased to $\mathcal{O}(N^3)$ and $\mathcal{O}(KN^2)$, respectively. This method is particularly useful when the support function is difficult to derive. Note that the problem in (9) is equivalent to the dual of this dual problem in (11).

To determine the optimal schedule $\mathbf{s}^*$, we reference the dual variables associated with the first constraint in the dual problem (see Theorem 2.3 in Bertsimas and den Hertog (2022)).

3.2 Lower bound

To establish a lower bound for the robust problem (2), we first select a finite subset of scenarios $\tilde{\mathcal{U}}$ from the full uncertainty set $\mathcal{U}$. The selection of $\tilde{\mathcal{U}}$ leverages information of dual problem's optimal solutions, specifically $\mathbf{V}^*$, $\mathbf{x}^*$ and $\mathbf{w}^*$. Note that $\mathbf{x}^*$ may not represent a worst-case scenario directly, due to its absence in the dual problem's objective. Instead, we extract potential scenarios by ensuring the optimal relationship $\mathbf{V}^* = \mathbf{w}^*\mathbf{x}^{*T}$ is maintained, even though this is not explicitly

enforced by the model. Each row of $\mathbf{V}^*$, $\mathbf{v}_m^{*T}$, should theoretically satisfy $\mathbf{v}_m^{*T} = w_m \mathbf{x}^{*T}$. From this, candidates for $\mathbf{x}^*$ are deduced by dividing each row of $\mathbf{v}_m^{*T}$ by its corresponding w_m, yielding multiple scenario candidates $\tilde{\mathbf{x}}_m$. Only those that lie within $\mathcal{U}$ and are unique are kept in $\tilde{\mathcal{U}}$.

Imposing a constraint for each scenario in $\tilde{\mathcal{U}}$, we solve the deterministic problem:

$$\begin{aligned} &\min_{\mathbf{s},\tau} \ \tau \\ &\text{s.t. } f(\mathbf{s},\mathbf{x}) \leq \tau \ \ \forall \mathbf{x} \in \tilde{\mathcal{U}} \\ &\quad \mathbf{s} \in S. \end{aligned} \tag{12}$$

This problem features up to M convex constraints and is straightforward to solve. Given $\tilde{\mathcal{U}} \subset \mathcal{U}$, the solution provides a conservative estimate, thus establishing a lower bound.

To derive a tighter lower bound, we employ an adversarial approach for scenario selection combined with a hill climbing method for scenario improvement. Starting with an empty set, $\tilde{\mathcal{U}}_0$, we initially select the highest-cost scenario from $\tilde{\mathcal{U}}$ given the upper bound schedule. For subsequent iterations, we recalculate the optimal schedule based on the updated $\tilde{\mathcal{U}}_0$ and continue to add scenarios that yield the highest costs under this new schedule. This iterative process continues until adding more scenarios does not increase the resulting costs.

Subsequently, we apply hill climbing on these selected scenarios in $\tilde{\mathcal{U}}_0$ to refine them into $\tilde{\mathcal{U}}_1$. Inspired by Zhen et al. (2021) and Bertsimas et al. (2023), we fix the schedule $\mathbf{s}'$ from the adversarial phase and address the following bilinear form of the problem (6):

$$\sup_{\mathbf{x}\in\mathcal{U},\mathbf{w}\in\mathrm{dom}h^*} \left\{\mathbf{w}^T\mathbf{A}\mathbf{x} + \mathbf{b}(\mathbf{s}')^T\mathbf{w}\right\}.$$

Initially, we fix scenario $\mathbf{x}'$ as one of the candidates in $\tilde{\mathcal{U}}_0$ and aim to find an optimal $\mathbf{w}$ within the domain dom h^*. Therefore, we solve the following linear problem:

$$\max_{\mathbf{w}\in\mathrm{dom}h^*} \mathbf{w}^T(\mathbf{A}\mathbf{x}' + \mathbf{b}(\mathbf{s}')) = \sum_{n=1}^{N} \max_{\{\mathbf{w}_n:\, \mathbf{w}_n\geq 0, \mathbf{1}^T\mathbf{w}_n = 1\}} \mathbf{w}_n^T(\mathbf{A}\mathbf{x}' + \mathbf{b}(\mathbf{s}'))_n.$$

For all n, optimal allocation is achieved by assigning $w_{ni} = 1$ to the index that maximizes the value of $(\mathbf{A}\mathbf{x}' + \mathbf{b}(\mathbf{s}'))_n$, and $w_{ni} = 0$ otherwise. Hence, when $\mathbf{e}_i$ denotes the unit vector, we set

$$\mathbf{w}'_n = \mathbf{e}_{i^*} \text{ with } i^* = \arg\max(\mathbf{A}\mathbf{x}' + \mathbf{b}(\mathbf{s}'))_n. \tag{13}$$

With this $\mathbf{w}'$ established, we next seek an improved $\mathbf{x}''$ by optimizing the linear problem over the convex set $\mathcal{U}$:

$$\mathbf{x}'' = \arg \max_{x\in\mathcal{U}} \mathbf{w}'^T\mathbf{A}\mathbf{x} + \mathbf{b}(\mathbf{s}')^T\mathbf{w}' = \arg \max_{\mathbf{x}\in\mathcal{U}} \mathbf{w}'^T\mathbf{A}\mathbf{x}. \tag{14}$$

This problem can be efficiently solved. However, whether this problem also has an explicit solution, depends on the choice of the uncertainty set.

This iterative process alternates between refining $\mathbf{w}$ and $\mathbf{x}$ until no further improvements are observed in the objective value. Each scenario in $\tilde{\mathcal{U}}_0$ is processed in this manner, producing an improved scenario to include in $\tilde{\mathcal{U}}_1$. This refined set is then utilized to solve the deterministic problem (12), providing a tighter lower bound.

3.3 Exact solution

To obtain the exact robust bound and corresponding optimal robust schedule for problem (2), we use a cutting-set approach to iteratively refine the lower bound by expanding the scenario set $\tilde{\mathcal{U}}_1$ one-by-one (Bienstock and Özbay (2008), Bertsimas et al. (2016)). The method starts with a fixed schedule $\mathbf{s}'$, derived from the lower bound calculations, and identifies the corresponding worst-case scenario $\mathbf{x}$ from the uncertainty set to add to the expanded set $\tilde{\mathcal{U}}_2$:

$$\max_{\mathbf{x}\in\mathcal{U}} f(\mathbf{s}', \mathbf{x}). \tag{15}$$

This maximization problem, due to containing max operators, requires transforming into a mixed integer linear optimization problem, introducing binary variables and additional constraints. Given the potential time intensity of this task, particularly for large customer sets, we may impose a limit on this optimization and incorporate the current, possibly sub-optimal, value of $\mathbf{x}$ into $\tilde{\mathcal{U}}_2$.

Next, we recompute the optimal schedule $\mathbf{s}$ and associated costs for the expanded set $\tilde{\mathcal{U}}_2$:

$$\min_{\mathbf{s}\in\mathcal{S}} \max_{\mathbf{x}\in\tilde{\mathcal{U}}_2} f(\mathbf{s}, \mathbf{x}), \tag{16}$$

which is then used as the basis for the subsequent iteration of (15).

For each iteration, (15) and (16) result in a new upper and lower bound, respectively. The lower bound progressively increases as scenarios are added to $\tilde{\mathcal{U}}_2$. Simultaneously, the cost of the worst-case scenario, derived from a feasible but potentially non-optimal schedule, establishes an upper bound, but on the contrary this does not necessarily decrease with each iteration. The process terminates once the gap between these two bounds narrows below a predefined precision level ϵ. Convergence is guaranteed by Section 5.2 in Mutapcic and Boyd (2009), given that:

- The master problem (16) is solved exactly.
- The subproblem (15) of finding the worst-case is solved exactly.
- The function $f(\mathbf{s}, \mathbf{x})$ is uniformly Lipschitz continuous in $\mathbf{s}$ (see proof in Appendix A.2).
- The feasible set $\mathcal{S}$ is bounded.

In summary, the method outlined above applies for any convex uncertainty set, though a particular uncertainty set requires several specific steps. For the upper bound, particularly, the RPT technique in Step 2 must be tailored to the convex constraints in the set. This adaptation directly influences Step 3, where both the support function derivation and the constraint formulation in the dual problem are affected. For the lower bound, the complexity of the hill-climbing method varies with the type of uncertainty, occasionally permitting an analytical solution. The duration to determine the exact solution also greatly depends on the complexity and dimensions of the uncertainty set. The next section will illustrate how these methods are specifically adapted for polyhedral and ellipsoidal uncertainty sets.

4 Application to uncertainty sets

In this section, we apply the general robust ASP framework to both the polyhedral and ellipsoidal uncertainty set, and detail the steps that need to be adapted to these specific constraints. Both sets are widely applied in RO; see Bertsimas and den Hertog (2022).

4.1 Polyhedral uncertainty

Here, we assume polyhedral uncertainty for $\mathbf{x}$, defined by a series of K linear inequalities: $\mathbf{x} \in \mathcal{U} = \{\mathbf{x} : \mathbf{D}\mathbf{x} \leq \mathbf{d}\}$ with $\mathbf{D} \in \mathbb{R}^{K \times N}$ and $\mathbf{d} \in \mathbb{R}^K$.

For the upper bound, we start with the reformulated constraint (6), and consider the preliminary uncertainty set: $\Theta_0^{\text{pol}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \mathbf{D}\mathbf{x} \leq \mathbf{d},\ \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\}$. To refine this, we apply RPT by multiplying the linear inequalities in $\mathbf{D}\mathbf{x} \leq \mathbf{d}$ with $\mathbf{w} \geq 0$, and the equality $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ with $\mathbf{x}$, and then substituting $\mathbf{V} = \mathbf{w}\mathbf{x}^T$:

$$\begin{aligned} &(\mathbf{w} \geq 0) \ \times\ (\mathbf{D}\mathbf{x} - \mathbf{d} \leq 0) && \rightarrow \quad \mathbf{V}\mathbf{D}^T - \mathbf{w}\mathbf{d}^T \leq 0 \\ &(\mathbf{H}^T\mathbf{w} = \mathbf{1}) \ \times\ \mathbf{x} && \rightarrow \quad \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T. \end{aligned}$$

This results in an improved convex uncertainty set:

$$\Theta_1^{\text{pol}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid (\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_0^{\text{pol}},\ \mathbf{V}\mathbf{D}^T - \mathbf{w}\mathbf{d}^T \leq 0,\ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T\}.$$

We then use this refined set to solve (8) by applying the two methods based on either the support function or the dual formulation.

Given the polyhedral nature of the set Θ_1^{pol}, that can be expressed as $\Theta_1^{\text{pol}} = \{\mathbf{z} \mid \tilde{\mathbf{D}}\mathbf{z} \leq \tilde{\mathbf{d}}\}$, where $\mathbf{z} = (\mathbf{w};\ \mathbf{x};\ \mathbf{v}_1;\ \ldots;\ \mathbf{v}_N)$ includes all decision variables, we derive its support function. The details of $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{d}}$ are outlined in Appendix B.2. The support function then simplifies to a linear system (Bertsimas and den Hertog 2022):

$$\delta^*(\mathbf{y}(\mathbf{s}) \mid \Theta_1^{\text{pol}}) = \max_{\mathbf{z}} \{\mathbf{y}(\mathbf{s})^T\mathbf{z} \mid \tilde{\mathbf{D}}\mathbf{z} \leq \tilde{\mathbf{d}}\} = \min_{\mathbf{u}} \{\tilde{\mathbf{d}}^T\mathbf{u} \mid \tilde{\mathbf{D}}^T\mathbf{u} = \mathbf{y}(\mathbf{s}),\ \mathbf{u} \geq 0\}.$$

Integrating this into our optimization problem (9) and using the structure of $\tilde{\mathbf{D}}$, $\tilde{\mathbf{d}}$ and $\mathbf{y}(\mathbf{s})$, Appendix B.2 demonstrates how we obtain the following result:

$$\begin{aligned} \min_{\mathbf{s}, \tau, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{W}_4, \mathbf{W}_5} \quad & \tau \\ \text{s.t.} \quad & \mathbf{d}^T\mathbf{w}_1 + \mathbf{1}^T\mathbf{w}_3 \leq \tau \\ & -\mathbf{w}_2 + \mathbf{H}\mathbf{w}_3 - \mathbf{W}_4\mathbf{d} = \mathbf{b}(\mathbf{s}) \\ & \mathbf{D}^T\mathbf{w}_1 - \mathbf{W}_5^T\mathbf{1} = \mathbf{0} \\ & \mathbf{W}_4\mathbf{D} + \mathbf{H}\mathbf{W}_5 = \mathbf{A} \\ & \mathbf{w}_1, \mathbf{w}_2, \mathbf{W}_5 \geq 0 \\ & \mathbf{1}^T\mathbf{s} = T \\ & \mathbf{s} \geq 0. \end{aligned} \tag{17}$$

This linear optimization problem is characterized by $\mathcal{O}(KN^2)$ decision variables and $\mathcal{O}(N^3)$ linear constraints, which are addressed directly through standard optimization techniques.

The dual approach incorporates the constraints from Θ_1^{pol} into (11), leading to the following linear

optimization problem:

$$\begin{aligned} \min_{y,\mathbf{w},\mathbf{x},\mathbf{V}} \; & Ty - (\mathrm{Tr}(\mathbf{A}^T\mathbf{V}) + \alpha(\mathbf{w})) \\ \text{s.t. } & \theta(\mathbf{w}) + \mathbf{1}y \geq 0 \\ & y_2 \geq 0 \\ & \mathbf{D}\mathbf{x} \leq \mathbf{d} \\ & \mathbf{w} \geq 0 \\ & \mathbf{H}^T\mathbf{w} = \mathbf{1} \\ & \mathbf{w}\mathbf{d}^T - \mathbf{V}\mathbf{D}^T \geq 0 \\ & \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T. \end{aligned} \tag{18}$$

The generation of a lower bound adheres to the general framework, with the scenario set $\tilde{\mathcal{U}}$ being refined to to ensure all candidates meet the criteria $\mathbf{D}\mathbf{x} \leq \mathbf{d}$. Transitioning from $\tilde{\mathcal{U}}_0$ to $\tilde{\mathcal{U}}_1$ through hill climbing involves iterative computation of (13) and (14). While this last optimization lacks an explicit solution for polyhedral uncertainty, finding a numerical solution remains easy due to the linearly of the problem over a polyhedron.

The exact solution is derived using the cutting set as described in the general framework, which converges without specific requirements on the set $\mathcal{U}$. Theoretically, an optimal solution could be obtained by evaluating all extreme points of the polyhedron $\{\mathbf{x} \; : \; \mathbf{D}\mathbf{x} \leq \mathbf{d}\}$ as the optimum is always attained at one of those points, and using these points as the scenario set in model (12). Nonetheless, the exponential growth in the number of extreme points with N renders this approach impractical, and therefore we use the iterative process described in the general framework.

4.2 Ellipsoidal uncertainty

Here, we define an ellipsoidal uncertainty set for the service times as $\mathbf{x} \in \mathcal{U} = \{\mathbf{x} : (\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{Q}(\mathbf{x}-\bar{\mathbf{x}}) \leq \rho^2\} = \{\mathbf{x} : ||\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})|| \leq \rho\}$, where $\bar{\mathbf{x}}$ represents the nominal service times, and $\mathbf{Q}$ is a positive definite symmetric matrix.

For the upper bound, we integrate the ellipsoidal constraint into (6), so we formulate the preliminary uncertainty set as $\Theta_0^{\text{ell}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x}-\bar{\mathbf{x}})\| \leq \rho, \mathbf{w} \geq 0, \; \mathbf{H}^T\mathbf{w} = \mathbf{1}\}$. Progressing to the RPT application, we multiply the ellipsoidal constraint $\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho$ with $\mathbf{w} \geq 0$, and $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ with $\mathbf{x}$, and substitute $\mathbf{V} = \mathbf{w}\mathbf{x}^T$, resulting in convex constraints:

$$\begin{aligned} & (\mathbf{w} \geq 0) \; \times \; \left(\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| - \rho \leq 0\right) && \rightarrow \; \left(\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{V}_m - w_m\bar{\mathbf{x}})\|\right)_{m=1}^{M} - \rho\mathbf{w} \leq 0 \\ & (\mathbf{H}^T\mathbf{w} = \mathbf{1}) \; \times \; \mathbf{x} && \rightarrow \; \mathbf{H}^T\mathbf{V} = \mathbf{1}^T\mathbf{x}, \end{aligned}$$

where $\mathbf{V}_m$ denotes the m-th row of $\mathbf{V}$. Consequently, the revised uncertainty set, Θ_1, is

$$\Theta_1^{\text{ell}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid (\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_0^{\text{ell}}, \; \left(\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{V}_m - w_m\bar{\mathbf{x}})\|\right)_{m=1}^{M} - \rho\mathbf{w} \geq 0, \; \mathbf{H}^T\mathbf{V} = \mathbf{1}^T\mathbf{u}\}.$$

We then solve (8) using this defined uncertainty set to identify the optimal robust schedule.

To address this simpler robust problem, we look into the two approaches suggested in Section 3.1. The first involves the computation of the support function of Θ_1, which in this case entails delving into the complexities of matrix-valued support functions (as discussed in Section 13.2 of Bertsimas and den Hertog (2022)), along with the associated composition rules. The second method utilizing the dual problem, offers a more straightforward implementation that directly incorporates Θ_1^{ell} into the model in (11), obtaining the following conic quadratic optimization problem:

$$\begin{aligned}
\min_{y,\mathbf{w},\mathbf{x},\mathbf{V}} \quad & Ty - \text{Tr}(\mathbf{A}^T\mathbf{V}) \\
\text{s.t.} \quad & -\mathbf{A}^T\mathbf{w} + \mathbf{1}y \geq 0 \\
& y \geq 0 \\
& \|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho \\
& \mathbf{w} \geq 0 \\
& \mathbf{H}^T\mathbf{w} = \mathbf{1} \\
& \rho\mathbf{w} - \left(\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{V}_m - w_m\bar{\mathbf{x}})\|\right)_{m=1}^{M} \geq 0 \\
& \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T.
\end{aligned} \tag{19}$$

This yields a tractable problem with variables and constraints scaling as $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$, respectively, making it feasible for conventional optimization solvers for a practical range of N.

For ellipsoidal uncertainty, deriving a lower bound mirrors the general framework outlined earlier. Notably, the hill climbing technique is enhanced by the known solution for maximizing a linear function over a norm-constrained sphere, as detailed in Section 2.3 in Bertsimas and den Hertog (2022) using the Karush-Kuhn-Tucker optimality conditions. This provides an explicit formula for updating $\mathbf{x}''$ under the ellipsoidal norm constraint:

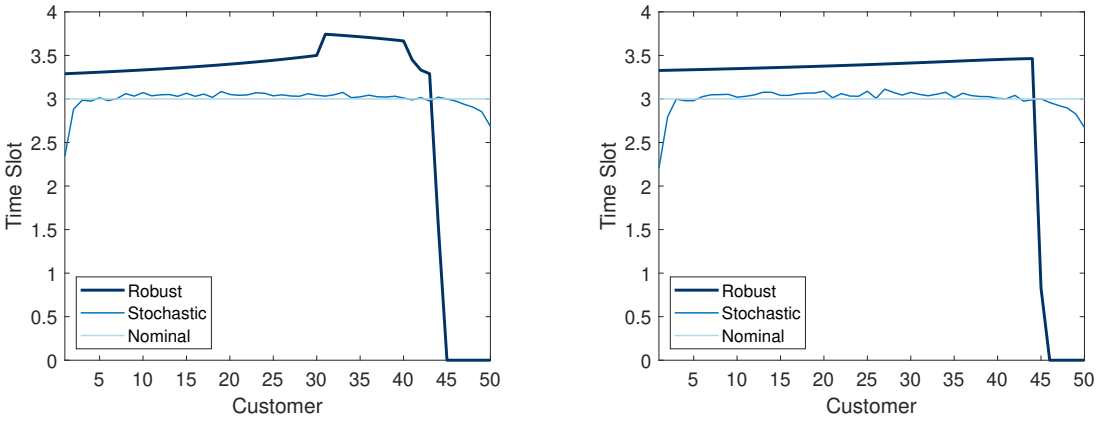
$$\mathbf{x}'' = \arg \max_{\{\mathbf{x}:\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x}-\bar{\mathbf{x}})\|\leq\rho\}} \mathbf{w}'^T\mathbf{A}\mathbf{x} = \arg \max_{\{\mathbf{x}:\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x}-\bar{\mathbf{x}})\|\leq\rho\}} \mathbf{w}'^T\mathbf{A}\mathbf{Q}^{-\frac{1}{2}}\mathbf{Q}^{\frac{1}{2}}(\mathbf{x}-\bar{\mathbf{x}}) = \rho\frac{\mathbf{Q}^{-1}\mathbf{A}^T\mathbf{w}'}{||\mathbf{Q}^{-\frac{1}{2}}\mathbf{A}^T\mathbf{w}'||}+\bar{\mathbf{x}}.$$

This eliminates the need for iterative optimization, significantly boosting efficiency.

For exact solutions under ellipsoidal uncertainty, the procedure aligns with the established framework. Nonetheless, identifying the worst-case scenario in (15) now involves solving a conic quadratic optimization problem due to optimization over the ellipsoidal set, which is tractable but generally more expensive than solving the linear problem for polyhedral uncertainty.

5 Computational results

We now apply the general framework described in Section 3 to problem instances with polyhedral and ellipsoidal uncertainty sets (for which detailed steps were given in Sections 4.1 and 4.2, respectively). We discuss characteristics of worst-case scenarios, qualitative features of optimal schedules, scalability and running times. Numerical experiments are performed on an Intel(R) Core(TM) i7-1165G7 2.80GHz Windows computer with 16GB of RAM. All computations are conducted with MOSEK (10.0.27) and implemented using YALMIP (R20210331) in MATLAB (R2023b).



(a) Budget uncertainty *(b) Ball uncertainty*

Figure 2: Optimal robust schedules for $N = 50$ customers, with the stochastic and nominal schedule for comparison

5.1 Set-up of numerical experiment

We illustrate the functionality of the robust ASP framework under budget uncertainty. This specific type of uncertainty set constrains each service duration within an interval while imposing a bound on the total permissible (absolute) deviation from the nominal values. So, budget uncertainty combines the ∞-norm and 1-norm constraints with parameters ρ and B, respectively. Note that this is a special case of polyhedral uncertainty $\mathcal{U} = \{\mathbf{x} : \mathbf{D}\mathbf{x} \leq \mathbf{d}\}$ with $2N$ uncertain parameters, by introducing an auxiliary variable $\mathbf{w}$ to represent the absolute deviations:

$$\begin{aligned}\mathcal{U} &= \{\mathbf{x} \ : \ ||\mathbf{x} - \bar{\mathbf{x}}||_\infty \leq \rho, \ ||\mathbf{x} - \bar{\mathbf{x}}||_1 \leq B\} \\ &= \{\mathbf{x} \ : \mathbf{x} \leq \rho\mathbf{1} + \bar{\mathbf{x}}, \ -\mathbf{x} \leq \rho\mathbf{1} - \bar{\mathbf{x}}, \ \mathbf{x} - \mathbf{w} \leq \bar{\mathbf{x}}, \ -\mathbf{x} - \mathbf{w} \leq -\bar{\mathbf{x}}, \ \mathbf{1}^T\mathbf{w} \leq B\}.\end{aligned}$$

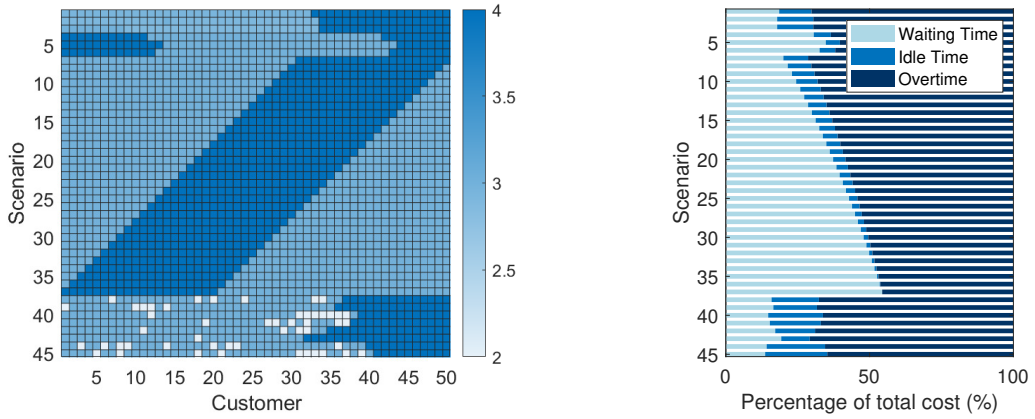
In our numerical experiment, we consider $N = 50$ customers, each with a nominal service duration of $\bar{\mathbf{x}} = 3$. The uncertainty bound $\rho = 1$ allows the durations to vary between 2 and 4. The total budget $B = \rho N/2.5 = 20$ implies that no more than 40% of customers can deviate to either the highest or lowest duration simultaneously. The total available time for appointments, $T = \bar{\mathbf{x}}N = 150$, ensures a full schedule under nominal conditions without uncertainty. Cost parameters are set as $c_O = 3$, $c_W = 0.2$ and $c_I = 1$ for overtime, waiting time and idle time, respectively.

5.2 Analysis of schedules

5.2.1 Base case

The optimal robust schedule from our numerical experiment is depicted in Figure 2a. Clearly, we find a dome-shaped schedule that aligns with those often seen for stochastic models, but with additional strategic scheduling nuances that enhance performance under budget uncertainty:

- *Increasing time slots for initial customers:* The early part of the schedule shows progressively increasing time slots for the first 30 customers. Allocating slightly more time than the nominal



(a) Heatmap of the service duration per customer in each scenario

(b) Distribution of worst-case costs across the three cost components in each scenario

Figure 3: Visualization of the worst case scenarios for budget uncertainty

duration at the beginning helps in absorbing the unpredictability without causing significant idle times. The (increasingly) longer time slots ensure that early slight delays do not snowball into significant waiting times for later customers.

- *Stabilization phase:* At customer 31, there is a noticeable increase in the time slots, which then stabilize for the next ten customers. This plateau can handle any accumulated delays that might have surpassed the buffers provided earlier in the schedule.
- *Reduction towards the end:* As the schedule nears its end (after customer 40), there is a reduction in the time slots allocated, crucial to mitigate the risk of incurring overtime.
- *Zero allocation for final customers:* From customer 45 onwards, the time slots are set to zero, effectively scheduling these appointments at day's end. This decision recognizes and strategically plans for unavoidable overtime due to the combined impact of budget uncertainty and the fixed time constraint.

Our analysis identifies 45 critical worst-case scenarios in the final scenario set $\tilde{\mathcal{U}}_1$, all resulting in costs of 131.52. The first 37 scenarios were identified early in the formation of $\tilde{\mathcal{U}}_1$, while the remaining 8 emerged during the iterative cutting-set approach. Despite the multitude of worst-case scenarios, a relatively small subset suffices to determine the optimal schedule.

The heatmap in Figure 3a illustrates the service durations for customers across these scenarios, where darker colors indicate longer durations. Additionally, Figure 3b shows the distribution of total costs across the three components: waiting time, idle time and overtime. The heatmap reveals a tendency for service times to be close to the maximal duration, contributing to a significant portion of the costs being attributed to overtime. Over half of the scenarios show these longer service times clustering, affecting the cost component balance: early clusters increase waiting times, while later ones contribute to more idle and overtime costs. In the last scenarios, extended service times dominate the end of the day, combined with multiple short service periods throughout the day,

	budget uncertainty		ball uncertainty	
	worst-case	average	worst-case	average
nominal	218.00	24.87	193.90	30.48
SAA	217.38	23.33	194.33	28.67
robust	131.52	89.88	112.32	79.22

Table 1: Comparison of the nominal, stochastic and robust schedule in terms of the worst-case costs over the budget and ball uncertainty set and the average costs over 50,000 uniform scenarios

leading to a significant rise in idle time costs. However, idle time generally incurs the least expense, underscoring that the server's time is effectively optimized within the robust schedule.

5.2.2 Comparison with stochastic solution

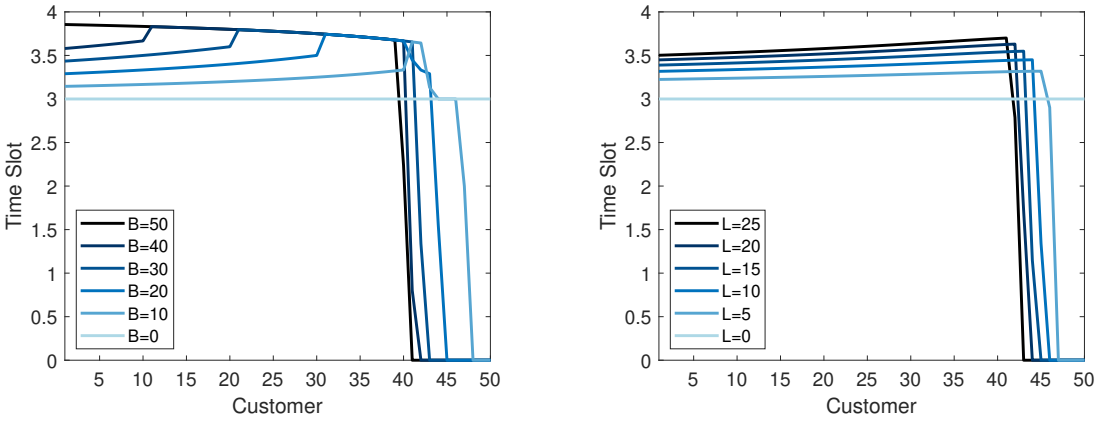
We now compare the robust schedule's performance against two other scheduling approaches: the nominal schedule, which assigns the nominal service time of 3 to each to each customer, hence ignoring the uncertainty, and a stochastic schedule based on Sample Average Approximation (SAA) using a small set of scenarios (10,000) to maintain computational feasibility, uniformly distributed over the support of the uncertainty set. Both alternative schedules are depicted in Figure 2a. To evaluate these strategies, we generate 50,000 new scenarios from the same distribution and determine the average cost. Additionally, we compute the exact worst-case costs for each schedule over the full uncertainty as in (15). The results are reported in Table 1.

The robust schedule significantly outperforms the others in worst-case scenarios, capping the worst-case costs at 131.52, compared to 218.00 for the nominal schedule (found in the single worst-case scenario in which all maximal durations are for first 20 appointments) and 217.38 for the stochastic schedule. This demonstrates its effectiveness in managing the risks associated with uncertainty. However, this comes at a cost: the average cost of the robust schedule (89.88) is considerably higher than both the nominal (24.87) and stochastic schedules (23.33). This indicates a trade-off between robustness and cost-efficiency, highlighting that preparing for worst-case scenarios can significantly increase average costs, particularly when guarding against extensive budget uncertainties.

5.2.3 Budget parameter

To study how the uncertainty budget B influences appointment scheduling, Figure 4a presents schedules for under different budget levels, yielding the following insights:

- $B = 50$: This setting, essentially box uncertainty, allows maximum deviation for each duration. It leads to a decreasing schedule, allocating longer initial time slots to mitigate accumulation of waiting times throughout the day. This pattern aligns with findings from Mittal et al. (2014) under similar uncertainty. Notably, if overtime costs are disregarded ($c_O = 0$ and $T = \infty$), this approach directly yields their optimal schedule.
- $B \in \{40, 30, 20, 10\}$: As Γ decreases, indicating a tighter budget, there is less room for uncertainty in the durations. The optimal schedule responds by progressively reducing the time slots, starting from the earlier ones to align more closely with the nominal value. This adjustment introduces the dome-shaped schedule. Changes in the later slots are influenced by the remaining time available in the day.



(a) Budget uncertainty with varying B

(b) Ball uncertainty with varying L

Figure 4: Optimal robust schedules for $N = 50$ customers with varying level of uncertainty

- $B = 0$: With no budget for deviations from the nominal service time, this setting removes all uncertainty, leading to the straightforward optimal schedule where each customer is allocated their exact nominal service duration.

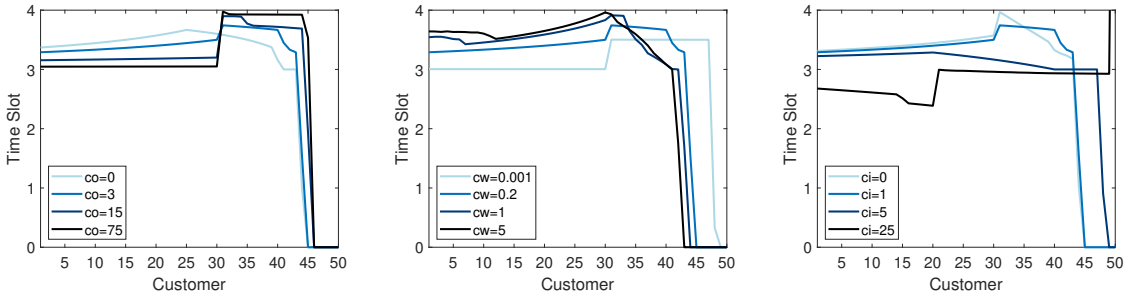
The overall trend is clear: higher uncertainty budgets require longer time slots early in the day to buffer against excessive waiting times, reducing time available for later appointments. Conversely, a lower budget leads to a compact schedule closer to the nominal values.

5.2.4 Cost parameters

To examine the impact of cost parameters, we start with the baseline values of $c_O = 3$, $c_W = 0.2$ and $c_I = 1$. We then modify each parameter independently to observe the effects on the optimal robust schedules. The results of these variations are displayed in Figure 5, and the corresponding worst-case scenarios are detailed in Appendix C.2.

Overtime. Figure 5a demonstrates how varying the overtime cost parameter c_O influences the scheduling pattern. At $c_O = 0$, the schedule exhibits a classical dome-shape. Increasing c_O to 3 (the base case), leads to shorter time slots for early customers, reserving more time for later appointments. For $c_O = 15$, there is a flattened distribution of initial time slots and a sharp increase later in the schedule. At $c_O = 75$, this flattening extends throughout the entire schedule. This pattern progressively shifts towards assigning nominal values of 3 to the first 30 customers and maximal values of 4 for the subsequent customers, leaving zero allocations at the end of day. This becomes even more evident when letting $c_O \to \infty$ (see Figure 10a for the schedule with $N = 25$ and $c_O = 3000$), resulting in the stark time slot pattern $3 \to 4 \to 0$.

Figure 11 displays the worst-case scenarios for these schedules. For $c_O = 0$, similar patterns to the base case ($c_O = 3$) emerge, with more occurrences of short service times, due to the lack of overtime penalties. Increasing c_O to 15 results in scenarios that predominantly feature maximal service times, typically towards the day's end. For $c_O = 75$, all scenarios extend service times to



(a) Overtime cost c_O *(b) Waiting time cost c_W* *(c) Idle time cost c_I*

Figure 5: Optimal robust schedules for varying cost parameters

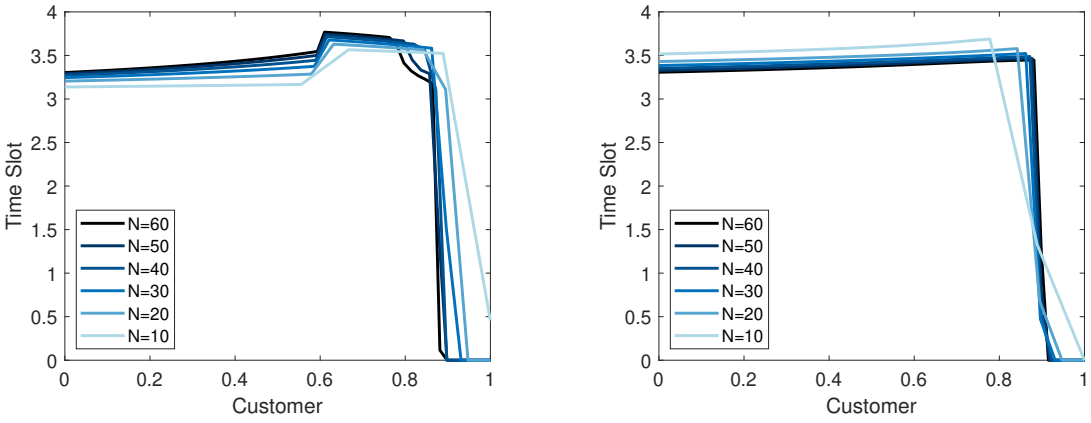
their maximum, spread out over the day, since this leads to the heavy-weighted overtime.

Waiting time. In Figure 5b, the schedules for varying waiting time costs are given. Starting with $c_W = 0.001$, which is a near-negligible cost (choosing $c_W = 0$ results in highly irregular schedules due to the absence of a trade-off in the cost function), the schedule for the first 30 customers remains consistent, each receiving the nominal time slot, followed by an increase to 3.5 until the day ends. At $c_W = 0.2$, the base case, the schedule allocates increasingly more time to early customers. When increasing c_W to 1 and then to 5 result in similar scheduling patterns, where even more time is allocated to the earliest appointments. Both start with a slight reduction in time slots, followed by a steady increase that peaks at the 30th customer and then declines sharply.

Figure 12 shows how the worst-case scenarios adapt to changes in waiting time costs. With $c_W = 0.001$, scenarios feature either extended service times in concentrated blocks or a combination of short service times during the day and long ones towards the end. When comparing $c_W = 1$ with the base case $c_W = 0.2$, scenarios with short service times vanish, replaced by ones with maximal service times at both the start and end of the day, thereby increasing waiting and overtime. This trend intensifies at $c_W = 5$, where waiting time becomes the dominating cost component.

Idle time. Figure 5c shows a large variability in the optimal schedule depending on the idle time costs. Compared to the base case with $c_I = 1$, for $c_I = 0$ the schedule features a more pronounced peak at customer 31 due to the absence of idle time penalties, followed by a sharper decrease in time slots. When c_I increases to 5, the schedule becomes more uniform, softening the extreme allocations and achieving a balance in the three cost factors, assigning time slots close to 3 as seen in the case without uncertainty. At $c_I = 25$, the early slots are significantly reduced as idle time is costly, and the schedule standardizes the remaining slots at the nominal value. Note that the last receives the residual day time. When c_I goes to infinity, the schedule converges to slots of 2 for the first 20 customers and slots of 3 for the last customers, with the last one receiving the residual time (see Figure 10b for $N = 25$ and $c_I = 1000$).

In Figure 13, for analyze the worst-case scenarios for these values of c_I. When $c_I = 0$ and there is no penalty for idle time, scenarios maximize service durations within the budget constraint, which appear variably throughout the day, contrary to base case ($c_I = 1$) that does include scenarios with shorter service times. Increasing c_I to 5 leads to a marked increase in the frequency of shorter



(a) Budget uncertainty *(b) Ball uncertainty*

Figure 6: Optimal robust schedule with budget uncertainty for varying N, where the horizontal axis represents the fraction of the total number of customers

service times, now predominating in the scenarios, particularly among the first 40 customers. These durations are either evenly spread throughout this initial segment or appear in a random pattern. The highest examined value, $c_I = 25$, shows an even more distinct pattern, with short service times concentrated early in the day when there is no backlog, directly contributing to idle time. Longer durations are less frequent and occur mainly early on, resulting in prolonged waiting times as the day progresses.

5.2.5 Number of customers

Figure 6a illustrates how the optimal scheduling pattern adjusts with varying the number of customers, considering $N = 10, 20, 30, 40, 50, 60$. On the horizontal axis we show the fraction of customers, effectively normalizing the schedules across different customer counts to facilitate a fair comparison of schedule characteristics, independent of the absolute number of customers.

The schedules display a striking 'scale invariance', suggesting that the overall pattern remains relatively consistent despite changes in customer numbers. However, notable shifts in the structure of the schedules occur, particularly at larger values of N. For these values, the initial customers are allocated longer slots, which increase more rapidly early in the schedule, to absorb potential delays that could now effect many subsequent customers. As N increases, the plateau in the time slots is reached progressively earlier and more abruptly in the sequence. This transition is followed by a steeper and earlier decline towards zero as the fraction of served customers approaches 1, accommodating the decreasing amount of remaining time in the day for later customers.

5.2.6 Comparison with ball uncertainty

In this section, we examine the differences and implications of adopting budget uncertainty versus ball uncertainty for modeling service times within the context of robust scheduling.

The comparison focuses on aligning the geometric interpretations of these uncertainty sets by equating their volumes under similar constraints. First, we estimate the volume of the budget uncertainty set via Monte Carlo simulation using the hit-or-miss estimator. We generate $Z = 10^7$ random points within a hypercube defined by the interval constraints of the nominal value $\bar{x}$ and maximal deviation ρ, spanning the space $[2, 4]^{50}$. We count the number of points that comply with the budget constraint, which limits the sum of deviations to $B = 20$. The fraction of points satisfying this constraint, multiplied by the hypercube's volume (2^{50}), gives us an estimated volume for the budget uncertainty set, $V_{\text{budget}} \approx 7.90 \times 10^{12}$.

For a comparable ellipsoidal uncertainty set, we define a simple ball with $\rho = 1$ and $Q = \frac{1}{L}I$, where L is a scaling factor to be determined and I is the identity matrix. Since $\det(Q) = L^N$, the volume formula for this ball, given by $V_{\text{ball}} = \pi^{N/2}L^{N/2}/\Gamma(N/2 + 1)$, with Γ the gamma function, allows us to match the volume of the ball to that of the budget set by adjusting L. This gives

$$L = \left(\frac{\Gamma(N/2 + 1) \cdot V_{\text{budget}}}{\pi^{N/2}} \right)^{2/N} \approx 10.63,$$

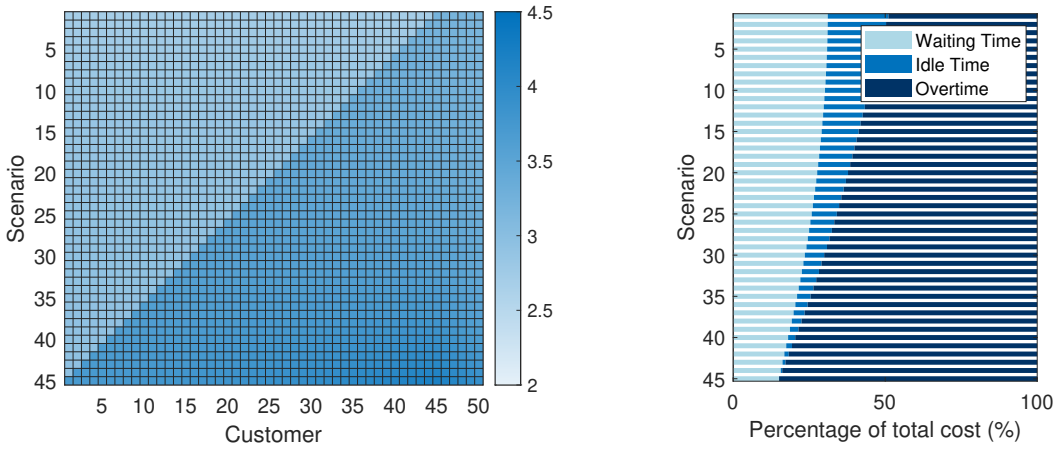
defining a ball uncertainty set with $Q \approx 0.094I$ and $\rho = 1$.

The robust schedule for this ball uncertainty set, depicted in Figure 2b, displays an increasing pattern. Starting with time slots of 3.3 and gradually increasing to 3.5 by the 44th customer, it abruptly drops to zero. Comparing this to the budget uncertainty schedule in Figure 2a, both schedules initially follow a very similar trajectory. However, while the budget schedule reaches a plateau and then decreases, the ball schedule steadily rises throughout.

This difference stems from the intrinsic characteristics of the uncertainty sets. Ellipsoidal sets are inherently smooth, and therefore tend to generate worst-case scenarios without extremities, leading to smoother schedules. These scenarios, as shown in Figure 7a, typically begin with uniformly short service times and then make a clear jump to uniformly longer service times, which can happen at any point throughout the day. Consequently, an increasing schedule is rational as it focuses on risk mitigation in the latter half of the day where most extended service times occur. On the contrary, the budget set is characterized by non-smooth boundaries and leads to worst-case scenarios at its vertices, leading to more pronounced extremities in both the schedule and scenarios as discussed before. However, both type of uncertainty sets feature an initial phase of shorter durations followed by a sharp transition to longer ones, which also aligns with the results for box uncertainty in Mittal et al. (2014), even though the optimal scheduling patterns differ.

The reduced extremity of the worst-case costs under ball uncertainty leads to a significant 15% decrease in worst-case costs, totalling 112.32. Observing the division of cost components in Figure 7b, this cost reduction is largely attributable to lower waiting times, since the scenarios generally feature shorter service durations, which also stabilizes overtime costs across scenarios. Table 1 compares the robust with the nominal and stochastic solutions, which are generated in the same way as for budget uncertainty and are illustrated in Figure 2b. It reveals that the reduction in worst-case costs of the nominal and stochastic solutions is of similar size as for the robust solution, making the robust solution still superior in terms of robustness. Additionally, the robust solution under ball uncertainty maintains better average cost performance than under budget uncertainty.

In Figure 4b, we observe the dynamic adjustment of the schedule as the uncertainty parameter L is varied. Again, for L set to 0, so without uncertainty, an equidistant schedule emerges. Increasing



(a) Heatmap of the service duration per customer in each scenario

(b) Distribution of worst-case costs across the three cost components in each scenario

Figure 7: Visualization of the worst case scenarios for ball uncertainty

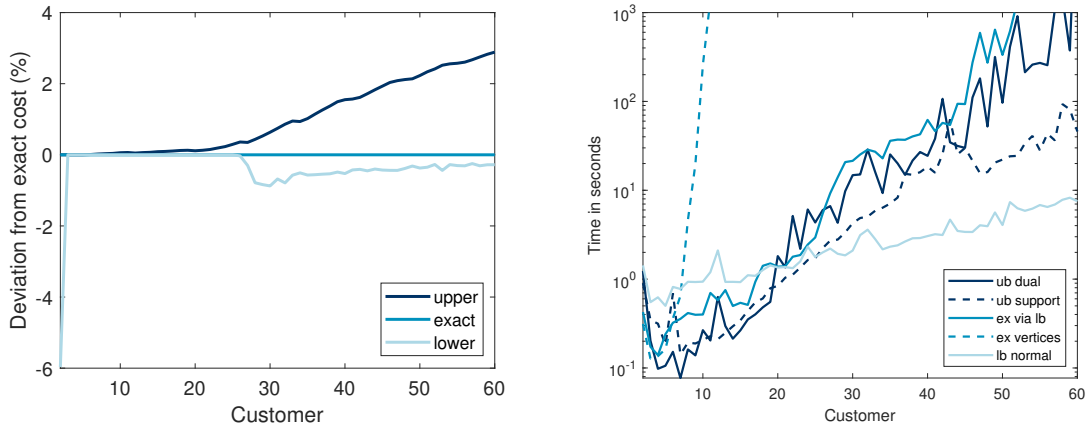
L then demonstrates similar behavior as for budget uncertainty, as it allocates more time to earlier appointments, reflecting a protective response to the higher level of uncertainty. Unlike budget uncertainty, the increment in time slots across nearly uniform across all customers. Notably, since there is no upper bound on the value of L, this pattern can continue indefinitely, potentially assigning progressively more time to the first few customers.

Appendix C, Figure 9, analyzes cost parameters under ellipsoidal uncertainty and draws many parallels with the observations for budget uncertainty in Section 5.2.4 With zero overtime costs, we observe a decreasing schedule, contrasting with the increasing schedule when overtime costs are factored in. Higher overtime costs again result in marginally shorter appointments earlier in the day to accommodate later ones in order to mitigate potential overtime, although these differences are less pronounced compared to budget uncertainty. Changes in waiting time costs consistently yield an overall increasing schedule. For lower costs, time slots sharply decline and approach the nominal value of 3, with a slight peak towards the end, similar to what is seen with budget uncertainty. Higher waiting time costs lead to very similar schedules across both types of uncertainty. The response to idle time cost variations shows the most diversity but follows a pattern akin to budget uncertainty. For low idle time costs, the schedule increases; for intermediate costs, it decreases; and for higher costs, it increases again. With an idle time cost of 25, appointment lengths reduce to below 3, although not to the extent of consistently converging to slots of 2.

5.3 Scalability and computational complexity

We assess the computational performance of our methods used to calculate the lower and upper bounds, and the exact robust costs, for varying number of customers, N.

Cost bounds. In Figure 8a, we observe the percentage deviation of the upper and lower bounds from the exact robust costs for N ranging from 1 to 60. The lower bound demonstrates remarkable



(a) Deviation of the upper and lower bound from the exact value in %

(b) Running time of the five methods in seconds on logarithmic scale

Figure 8: Results of the methods for budget uncertainty for varying N

accuracy, aligning precisely with the exact costs up to $N = 25$. Even as N increases further, the disparity between the lower bound and exact costs remains minimal, with the gap staying well below 1%. This underscores the effectiveness of this approach in closely approximating the true robust costs. On the other hand, the upper bound beings showing greater deviation as N increases. While the deviation remains relatively small up to $N = 35$, it grows more pronounced thereafter, exceeding 1% and displaying a linear increase with N. This suggests that while the upper bound provides a conservative estimate, its accuracy in tightly bracketing the exact costs diminishes as the customer base expands.

Running times. We evaluate the running times of various methods, as illustrated in Figure 8b (logarithmic scale). The methods include the upper bound calculated via the support function as well as via the dual problem, the lower bound, and the exact method building upon this lower bound. For comparison, we also include the exact method that explores all vertices of the budget uncertainty set to demonstrate the effectiveness of our framework.

For $N \leq 20$, the running times of the first four methods are comparably low, remaining under two seconds. As N increases beyond this point, all methods experience almost exponential growth in running times, though they remain computationally feasible up to $N = 60$. The upper bound calculated via the dual problem shows a particularly erratic increase in running times starting from $N = 30$, making it less efficient at larger scales compared to the more stable and faster performance of the upper bound via the support function. The exact method that uses the lower bound, exhibits the overall longest running times, with significant fluctuations due to the varying number of scenarios to be added. In general, the lower bound method proves the most efficient for larger customer sets. In contrast, the vertex exploration method for deriving exact values quickly becomes impractical for instances exceeding 11 customers, emphasizing the computational advantage of the developed methods in our framework, which can handle up to 60 customers effectively.

In summary, the lower bound offers an accurate approximation of the robust costs, providing a valuable and efficient alternative in case the exact method proves computationally too demanding. If even the lower bound become too resource-intensive (since it relies on the output of the upper bound via dual problem), the support function-based upper bound remains a viable option. Although it provides a less precise cost approximation, it maintains validity as an upper bound, suitable for cases where computational simplicity is prioritized.

6 Conclusions and outlook

We have presented a robust framework for ASP with uncertain durations, that can deal with general convex uncertainty sets and larger problem instances. The framework is equipped to deal with the convex cost function of ASP, and incorporates as key innovation a cutting-set approach, which provably converges to the optimal robust schedule. Employing the framework for polyhedral and ellipsoidal uncertainty sets, for instances with 50 customers, gave profound structural insights into how uncertainty in service times impacts the optimal schedule. For budget uncertainty sets, optimal schedules were shown to take a diverse array of dome-shaped patterns, which were confirmed for comparable ball uncertainty sets. The robust schedules were also shown to respond to the degree of uncertainty; conservative time allocations are preferred in settings with large uncertainty, to prevent potential disruptions due to early delays.

Our robust framework was shown to be scalable, able to deal with large instances of 50+ customers, and could be applied to real-world settings such as hospitals, as long as the available data on service times is converted into convex uncertainty sets. These could be polyhedral or ellipsoidal sets, but also general convex cones, information-based sets or intersections or sums of uncertainty sets. Another promising direction is to incorporate additional features into the ASP, such as no-shows, walk-ins, late arrivals, and variable capacity. Moreover, incorporating service levels constraints or integrating appointment sequencing could optimize the efficiency and quality of the schedule even further.

Crucial for our framework is that the ASP cost function takes the form of the sum-of-max of linear functions. Comparable cost functions occur in inventory management (e.g. multi-item newsvendor problems), service operations management (e.g. queueing and staffing), and project planning. Exploring whether our robust framework could be applied in these areas seems promising and worthwhile.

Acknowledgements

This work is supported by an NWO Mathematics Clusters grant and an NWO VICI grant.

References

Ahmadi-Javid, A., Jalali, Z., and Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1):3–34.

Ala, A. and Chen, F. (2022). Appointment scheduling problem in complexity systems of the healthcare services: A comprehensive review. *Journal of Healthcare Engineering*, 2022.

Bailey, N. T. (1952). A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 14(2):185–199.

Bandi, C. and Bertsimas, D. (2012). Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134:23–70.

Bandi, C. and Gupta, D. (2020). Operating room staffing and scheduling. *Manufacturing & Service Operations Management*, 22(5):958–974.

Bauerhenne, C., Kolisch, R., and Schulz, A. S. (2024). Robust appointment scheduling with waiting time guarantees. *arXiv preprint arXiv:2402.12561*.

Beck, A. and Ben-Tal, A. (2009). Duality in robust optimization: primal worst equals dual best. *Operations Research Letters*, 37(1):1–6.

Begen, M. A., Levi, R., and Queyranne, M. (2012). A sampling-based approach to appointment scheduling. *Operations Research*, 60(3):675–681.

Ben-Tal, A., And, T., and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research - MOR*, 23.

Ben-Tal, A., den Hertog, D., and Vial, J.-P. (2015). Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*, volume 28. Princeton university press.

Benjaafar, S., Chen, D., Wang, R., and Yan, Z. (2023). Appointment scheduling under a service-level constraint. *Manufacturing & Service Operations Management*, 25(1):70–87.

Bertsimas, D. and den Hertog, D. (2022). *Robust and adaptive optimization*. Dynamic Ideas, Belmont.

Bertsimas, D., den Hertog, D., Pauphilet, J., and Zhen, J. (2023). Robust convex optimization: A new perspective that unifies and extends. *Mathematical Programming*, 200(2):877–918.

Bertsimas, D., Dunning, I., and Lubin, M. (2016). Reformulation versus cutting-planes for robust optimization: A computational study. *Computational Management Science*, 13:195–217.

Bertsimas, D., Sim, M., and Zhang, M. (2018). Adaptive distributionally robust optimization. *Management Science*, 65(2):604–618.

Beyer, H.-G. and Sendhoff, B. (2007). Robust optimization–a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218.

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

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer, New York.

Cayirli, T. and Veral, E. (2003). Outpatient scheduling in health care: a review of literature. *Production and Operations Management*, 12(4):519–549.

Denton, B. and Gupta, D. (2003a). A sequential bounding approach for optimal appointment scheduling. *IIE Transactions*, 35(11):1003–1016.

Denton, B. and Gupta, D. (2003b). A sequential bounding approach for optimal appointment scheduling. *IIE transactions*, 35(11):1003–1016.

Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

Gorissen, B. L. and den Hertog, D. (2013). Robust counterparts of inequalities containing sums of maxima of linear functions. *European Journal of Operational Research*, 227(1):30–43.

Gorissen, B. L., Yanıkoğlu, İ., and den Hertog, D. (2015). A practical guide to robust optimization. *Omega*, 53:124–137.

Green, L. V. and Savin, S. (2008). Reducing delays for medical appointments: A queueing approach. *Operations Research*, 56(6):1526–1538.

Gupta, D. and Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, 40(9):800–819.

Hassin, R. and Mendel, S. (2008). Scheduling arrivals to queues: A single-server model with no-shows. *Management Science*, 54(3):565–572.

Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., and Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health Systems*, 1:129–175.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229.

Kong, Q., Lee, C.-Y., Teo, C.-P., and Zheng, Z. (2013). Scheduling arrivals to a stochastic service delivery system using copositive cones. *Operations Research*, 61(3):711–726.

Kuiper, A. and Lee, R. H. (2022). Appointment scheduling for multiple servers. *Management Science*, 68(10):7422–7440.

Kuiper, A., Mandjes, M., and de Mast, J. (2017). Optimal stationary appointment schedules. *Operations Research Letters*, 45(6):549–555.

Liu, N. (2016). Optimal choice for appointment scheduling window under patient no-show behavior. *Production and Operations Management*, 25(1):128–142.

Mak, H.-Y., Rong, Y., and Zhang, J. (2015). Appointment scheduling with limited distributional information. *Management Science*, 61(2):316–334.

Mercer, A. (1960). A queueing problem in which the arrival times of the customers are scheduled. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 108–113.

Mittal, S., Schulz, A. S., and Stiller, S. (2014). Robust appointment scheduling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Mutapcic, A. and Boyd, S. (2009). Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381–406.

Padmanabhan, D., Natarajan, K., and Murthy, K. (2021). Exploiting partial correlations in distributionally robust optimization. *Mathematical Programming*, 186:209–255.

Qi, J. (2016). Mitigating delays and unfairness in appointment systems. *Management Science*, 63(2):566–583.

Robinson, L. W. and Chen, R. R. (2003). Scheduling doctors' appointments: optimal and empirically-based heuristic policies. *IIE Transactions*, 35(3):295–307.

Schulz, A. S. and Udwani, R. (2019). Robust appointment scheduling with heterogeneous costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157.

van Eekelen, W., den Hertog, D., and van Leeuwaarden, J. S. H. (2024). Distributionally robust appointment scheduling that can deal with independent service times. *Working paper, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4892477*.

Wang, P. P. (1993). Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Research Logistics*, 40(3):345–360.

Zacharias, C. and Pinedo, M. (2014). Appointment scheduling with no-shows and overbooking. *Production and Operations Management*, 23(5):788–801.

Zacharias, C. and Yunes, T. (2020). Multimodularity in the stochastic appointment scheduling problem with discrete arrival epochs. *Management Science*, 66(2):744–763.

Zhen, J., de Moor, D., and den Hertog, D. (2021). An extension of the reformulation-linearization technique to nonlinear optimization. *Available at Optimization Online*.

Zhou, S., Ding, Y., Huh, W. T., and Wan, G. (2021). Constant job-allowance policies for appointment scheduling: Performance bounds and numerical analysis. *Production and Operations Management*, 30(7):2211–2231.

A Proofs

A.1 Convex conjugate of SML function

Consider the maximum of linear functions, so $f(\mathbf{y}) = \max_i y_i$, where each y_i is a linear function. The convex conjugate is defined as $f^*(\mathbf{w}) = \sup_{\mathbf{y}}\{\mathbf{y}^T\mathbf{w} - \max_i y_i\}$. When any $w_j \leq 0$, the supremum becomes ∞ as $y_j \to \infty$. Similarly, when $\sum_i w_i < 1$ or $\sum_i w_i > 1$, the supremum reaches ∞ by driving all y_i towards $-\infty$ or $+\infty$, respectively. Otherwise, the supremum is 0, achievable by setting all $y_i = 0$. Consequently, the value and domain of the conjugate are given by

$$f^*(\mathbf{w}) = 0 \;\text{ with }\; \text{dom } f^* = \{\mathbf{w} \mid \mathbf{w} \geq 0,\, \mathbf{1}^T\mathbf{w} = 1\}.$$

Extending this to the sum of the maximum of linear functions, denoted by $h(\mathbf{y}) = \sum_n f_n(\mathbf{y}) = \sum_n \max_i y_{ni}$, the convex conjugate and its domain, by the convolution property (see Theorem 2.1c in Bertsimas and den Hertog (2022)), are

$$h^*(\mathbf{w}) = \inf_{\{\mathbf{w}^n\}}\{\sum_n f_n^*(\mathbf{w}^n) | \sum_n \mathbf{w}^n = \mathbf{w}\} = 0,$$

$$\text{with dom } h^* = \{\mathbf{w} \mid \mathbf{w}_n \geq 0, \mathbf{1}^T\mathbf{w}_n = 1\, \forall n\}.$$

A.2 Lipschitz continuity sum-of-max

Here we prove that the function $g(\mathbf{s})$ on $\mathcal{S}$, being the sum of the maximum of linear functions in $\mathbf{s}$, is (uniformly) Lipschitz continuous. This function can generally be written as

$$g(\mathbf{s}) = \sum_i \max_j\{\mathbf{a}_{ij}^T\mathbf{s} + b_{ij}\}.$$

Then, for any $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}$, we have that

$$\begin{aligned}
|g(\mathbf{s}_1) - g(\mathbf{s}_2)| &= \Big| \sum_i \Big(\max_j\{\mathbf{a}_{ij}^T\mathbf{s}_1 + b_{ij}\} - \max_k\{\mathbf{a}_{ik}^T\mathbf{s}_2 + b_{ik}\} \Big) \Big| \\
&\leq \sum_i \Big| \max_j\{\mathbf{a}_{ij}^T\mathbf{s}_1 + b_{ij}\} - \max_k\{\mathbf{a}_{ik}^T\mathbf{s}_2 + b_{ik}\} \Big| \\
&\leq \sum_i \max_j \Big\{ \Big| \mathbf{a}_{ij}^T(\mathbf{s}_1 - \mathbf{s}_2) \Big| \Big\} \\
&\leq \sum_i \max_j \{\|\mathbf{a}_{ij}\|\} \cdot \|\mathbf{s}_1 - \mathbf{s}_2\|,
\end{aligned}$$

where we use the triangle inequality in the first inequality, the norm properties of the infinity norm in the second inequality, and the Cauchy-Schwarz inequality in the last one.

Thus, $g(\mathbf{s})$ is (uniformly) Lipschitz continuous with respect to $\mathbf{s}$, with constant $L_g = \sum_i \max_j\{\|\mathbf{a}_{ij}\|\}$.

B Derivations upper bound

B.1 Cost function

Below, we define the matrix $\mathbf{A}$ and vector $\mathbf{b}$ for the SML formulation of the ASP costs as given in (3), consisting of three different cost components:

1. For $n = 1, \ldots, N-1$: Waiting time of customer $n+1$, and therefore involves $n+1$ linear functions. We have $\mathbf{a}_{n0} = \mathbf{0}$ and $b_{n0}(\mathbf{s}) = 0$ to ensure the inclusion of 0 in the max operator. Furthermore, for $l = 1, \ldots, n$, we set $\mathbf{a}_{nl}$ having zeros except for entries c_W from $n-l+1$ to n, and $b_{nl}(\mathbf{s}) = -c_W \sum_{j=n-l+1}^{n} s_j$.
2. For $n = N$: Idle time, so it involves N linear functions. Also here, $\mathbf{a}_{n0} = \mathbf{0}$ and $b_{n0}(\mathbf{s}) = 0$. Then, for $l = 1, \ldots, N-1$, $\mathbf{a}_{nl}$ contains entries $-c_I$ in the first l positions, the rest are zeros, and $b_{nl}(\mathbf{s}) = c_I \sum_{j=1}^{l} s_j$.
3. For $n = N+1$: Overtime, consisting of $N+1$ linear components. Again, $\mathbf{a}_{n0} = \mathbf{0}$ and $b_{n0}(\mathbf{s}) = 0$. For entries $l = 1, \ldots, N$, $\mathbf{a}_{nl}$ has zeros except for the last $N-l+1$ to N entries filled with c_O, and $b_{nl}(\mathbf{s}) = -c_O \sum_{j=N-l+1}^{N} s_j$.

Matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ and vector $\mathbf{b}(\mathbf{s}) \in \mathbb{R}^M$ are then composed by concatenating all $\mathbf{a}_{nl}$ and $b_{nl}(\mathbf{s})$, respectively, resulting in $M = \frac{1}{2}N^2 + \frac{5}{2}N = \mathcal{O}(N^2)$ rows. Note that $\mathbf{b}(\mathbf{s})$ simplifies to $-\mathbf{A}\mathbf{s}$.

Example: We consider $N = 3$ customers.

The first term ($n = 1$), corresponding to the waiting time of customer 2, is $c_W \max\{0, x_1 - s_1\}$. We define $a_{10} = [0,0,0]$ and $b_{10} = 0$ for the 0 in the max operator, and $\mathbf{a}_{11}^T = [c_W, 0, 0]$ and $b_{11}(\mathbf{s}) = -c_W s_1$ for the linear function $c_W(x_1 - s_1)$. For the second term ($n = 2$), corresponding to the waiting time of customer 3, we have $c_W \max\{0, x_2 - s_2, x_1 - s_1 + x_2 - s_2\}$. We set $\mathbf{a}_{20}^T = [0,0,0]$ and $b_{20} = 0$, $\mathbf{a}_{21}^T = [0, c_W, 0]$ and $b_{21} = -c_W s_2$, and $\mathbf{a}_{22}^T = [c_W, c_W, 0]$ with $b_{22} = -c_W(s_1 + s_2)$. The third term ($n = 3$), $c_I \max\{0, s_1 - x_1, s_2 - x_2 + s_1 - x_1\}$, corresponds to the idle time. Therefore, we configure $\mathbf{a}_{30}^T = [0,0,0]$ and $b_{30} = 0$, $\mathbf{a}_{31}^T = [-c_I, 0, 0]$ and $b_{31} = c_I s_1$, and $\mathbf{a}_{32}^T = [-c_I, -c_I, 0]$ with $b_{32} = c_I(s_1 + s_2)$. The last term ($n = 4$) captures the overtime cost, being $c_O \max\{0, x_3 - s_3, x_2 - s_2 + x_3 - s_3, x_1 - s_1 + x_2 - s_2 + x_3 - s_3\}$, so $\mathbf{a}_{40}^T = [0,0,0]$ and $b_{40} = 0$, $\mathbf{a}_{41}^T = [0,0,c_O]$ and $b_{41} = -c_O s_3$, $\mathbf{a}_{42}^T = [0, c_O, c_O]$ and $b_{42} = -c_O(s_2 + s_3)$, and $\mathbf{a}_{43}^T = [c_O, c_O, c_O]$ with $b_{43} = -c_O(s_1 + s_2 + s_3)$.

The matrix $\mathbf{A}$ and vector $\mathbf{b}(\mathbf{s})$ are constructed as follows, with $M = \frac{1}{2} \cdot 3^2 + \frac{5}{2} \cdot 3 = 12$ rows:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ c_W & 0 & 0 \\ 0 & 0 & 0 \\ 0 & c_W & 0 \\ c_W & c_W & 0 \\ 0 & 0 & 0 \\ -c_I & 0 & 0 \\ -c_I & -c_I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_O \\ 0 & c_O & c_O \\ c_O & c_O & c_O \end{bmatrix} \quad \textit{and} \quad \mathbf{b}(\mathbf{s}) = \begin{bmatrix} 0 \\ -c_W s_1 \\ 0 \\ -c_W s_2 \\ -c_W(s_1 + s_2) \\ 0 \\ c_I s_1 \\ c_I(s_1 + s_2) \\ 0 \\ -c_O s_3 \\ -c_O(s_2 + s_3) \\ -c_O(s_1 + s_2 + s_3) \end{bmatrix} = -\mathbf{A}\mathbf{s}.$$

B.2 Support function polyhedral uncertainty

For polyhedral uncertainty, we consider the convex uncertainty set:

$$\Theta_1^{\text{pol}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \mathbf{D}\mathbf{x} \leq \mathbf{d}, \mathbf{w} \geq 0, \mathbf{H}^T\mathbf{w} = \mathbf{1},\ \mathbf{V}\mathbf{D}^T - \mathbf{w}\mathbf{d}^T \leq 0,\ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T\}.$$

This set can be expressed as a polyhedral set $\Theta_1^{\text{pol}} = \{\mathbf{z} \mid \tilde{\mathbf{D}}\mathbf{z} \leq \tilde{\mathbf{d}}\}$, where $\mathbf{z} = (\mathbf{w}; \mathbf{x}; \mathbf{v}_1; \ldots; \mathbf{v}_N)$, and the matrix $\tilde{\mathbf{D}}$ and vector $\tilde{\mathbf{d}}$ are defined as:

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & \mathbf{D} & 0 & \cdots & 0 \\ -\mathbf{I}_M & 0 & 0 & \cdots & 0 \\ \mathbf{H}^T & 0 & 0 & 0 & 0 \\ -\mathbf{H}^T & 0 & 0 & 0 & 0 \\ -d_1\mathbf{I}_M & 0 & d_{11}\mathbf{I}_M & \cdots & d_{1N}\mathbf{I}_M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -d_K\mathbf{I}_M & 0 & d_{K1}\mathbf{I}_M & \cdots & d_{KN}\mathbf{I}_M \\ 0 & -\mathbf{1}_N\mathbf{e}_1^T & \mathbf{H}^T & \cdots & 0 \\ 0 & \mathbf{1}_N\mathbf{e}_1^T & -\mathbf{H}^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -\mathbf{1}_N\mathbf{e}_N^T & 0 & \cdots & \mathbf{H}^T \\ 0 & \mathbf{1}_N\mathbf{e}_N^T & 0 & \cdots & -\mathbf{H}^T \end{bmatrix} \text{ and } \tilde{\mathbf{d}} = \begin{bmatrix} \mathbf{d} & \to \mathbf{u}_1 \\ \mathbf{0}_M & \to \mathbf{u}_2 \\ \mathbf{1}_N & \to \mathbf{u}_{3,1} \\ -\mathbf{1}_N & \to \mathbf{u}_{3,-1} \\ \mathbf{0}_M & \to \mathbf{u}_{4,1} \\ \vdots & \vdots \\ \mathbf{0}_M & \to \mathbf{u}_{4,K} \\ \mathbf{0}_N & \to \mathbf{u}_{5,1} \\ \mathbf{0}_N & \to \mathbf{u}_{5,-1} \\ \vdots & \vdots \\ \mathbf{0}_N & \to \mathbf{u}_{5,N} \\ \mathbf{0}_N & \to \mathbf{u}_{5,-N} \end{bmatrix} \tag{20}$$

To derive the upper bound, we consider the problem (9), employing the support function for the polyhedral set $\delta^*(\mathbf{y}(\mathbf{s})|\Theta_1^{\text{pol}}) = \min_{\mathbf{u}}\{\tilde{\mathbf{d}}^T\mathbf{u} \mid \tilde{\mathbf{D}}^T\mathbf{u} = \mathbf{y}(\mathbf{s}), \mathbf{u} \geq 0\}$, leading to the formulation:

$$\begin{aligned} \min_{\mathbf{s},\tau}\ & \tau \\ \text{s.t.}\ & \tilde{\mathbf{d}}^T\mathbf{u} \leq \tau \\ & \tilde{\mathbf{D}}^T\mathbf{u} = \mathbf{y}(\mathbf{s}) \\ & \mathbf{u} \geq 0 \\ & \mathbf{s} \in S. \end{aligned}$$

The vector $\mathbf{u}$ is partitioned into five segments, $\mathbf{u}_1$ through $\mathbf{u}_5$, each corresponding to distinct sets of constraints within Θ_1. A secondary index is applied for further subdivision, as denoted alongside the definitions of in (20). Utilizing the structure embedded within $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{d}}$, we obtain the optimization

problem

$$
\begin{aligned}
\min_{\mathbf{s},\tau} \quad & \tau \\
\text{s.t.} \quad & \mathbf{d}^T\mathbf{u}_1 + \mathbf{1}^T\mathbf{u}_{3,1} - \mathbf{1}^T\mathbf{u}_{3,-1} \leq \tau \\
& -\mathbf{u}_2 + \mathbf{H}\mathbf{u}_{3,1} - \mathbf{H}\mathbf{u}_{3,-1} - d_1\mathbf{u}_{4,1} - \ldots - d_K\mathbf{u}_{4,K} = \mathbf{b}(\mathbf{s}) \\
& \mathbf{D}^T\mathbf{u}_1 - \mathbf{e}_1\mathbf{1}^T\mathbf{u}_{7,1} + \mathbf{e}_1\mathbf{1}^T\mathbf{u}_{7,-1} - \ldots - \mathbf{e}_N\mathbf{1}^T\mathbf{u}_{7,N} + \mathbf{e}_N\mathbf{1}^T\mathbf{u}_{7,-N} = \mathbf{0} \\
& d_{11}\mathbf{u}_{4,1} + \ldots + d_{K1}\mathbf{u}_{4,K} + \mathbf{H}\mathbf{u}_{5,1} - \mathbf{H}\mathbf{u}_{5,-1} = \mathbf{a}_1 \\
& \vdots \\
& d_{1N}\mathbf{u}_{4,1} + \ldots + d_{KN}\mathbf{u}_{4,K} + \mathbf{H}\mathbf{u}_{5,N} - \mathbf{H}\mathbf{u}_{5,-N} = \mathbf{a}_N \\
& \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5 \geq 0 \\
& \mathbf{s} \in S.
\end{aligned}
$$

For variables $\mathbf{u}_3$ and $\mathbf{u}_5$, initially associated with equality constraints, we exclude variables with secondary index -1 and their corresponding constraints, retaining variables indexed with $+1$ and omitting their nonnegativity constraints. This adjustment leads to reformulating the problem in terms of variables $\mathbf{v}$:

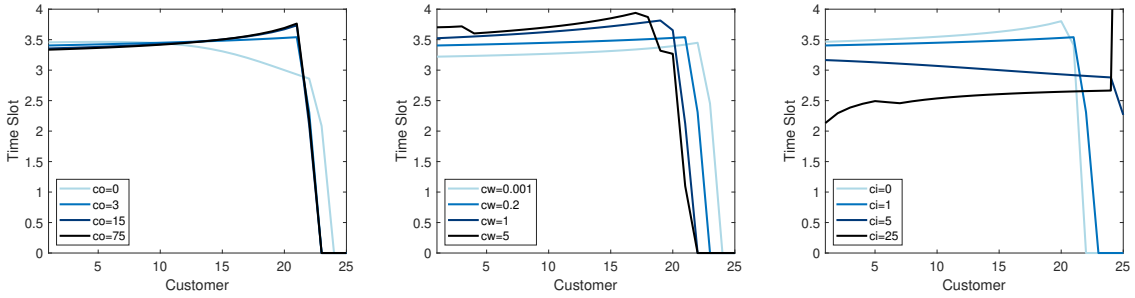
$$
\begin{aligned}
\min_{\mathbf{s},\tau} \quad & \tau \\
\text{s.t.} \quad & \mathbf{d}^T\mathbf{v}_1 + \mathbf{1}^T\mathbf{v}_3 \leq \tau \\
& -\mathbf{v}_2 + \mathbf{H}\mathbf{v}_3 - d_1\mathbf{v}_{4,1} - \ldots - d_K\mathbf{v}_{4,K} = \mathbf{b}(\mathbf{s}) \\
& \mathbf{D}^T\mathbf{v}_1 - \mathbf{e}_1\mathbf{1}^T\mathbf{v}_{5,1} - \ldots - \mathbf{e}_N\mathbf{1}^T\mathbf{v}_{5,N} = \mathbf{0} \\
& d_{11}\mathbf{v}_{4,1} + \ldots + d_{K1}\mathbf{v}_{4,K} + \mathbf{H}\mathbf{v}_{5,1} = \mathbf{a}_1 \\
& \vdots \\
& d_{1N}\mathbf{v}_{4,1} + \ldots + d_{KN}\mathbf{v}_{4,K} + \mathbf{H}\mathbf{v}_{5,N} = \mathbf{a}_N \\
& \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_5 \geq 0 \\
& \mathbf{s} \in S.
\end{aligned}
$$

For further simplification, we introduce decision matrices $\mathbf{W}_4 = (\mathbf{v}_{4,1}, \ldots, \mathbf{v}_{4,K}) \in \mathbb{R}^{M\times K}$ and $\mathbf{W}_5 = (\mathbf{v}_{5,1}, \ldots, \mathbf{v}_{5,N}) \in \mathbb{R}^{N\times N}$. Finally, the resulting linear optimization problem is:

$$
\begin{aligned}
\min_{\mathbf{s},\tau,\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3,\mathbf{W}_4,\mathbf{W}_5} \quad & \tau \\
\text{s.t.} \quad & \mathbf{d}^T\mathbf{w}_1 + \mathbf{1}^T\mathbf{w}_3 \leq \tau \\
& -\mathbf{w}_2 + \mathbf{H}\mathbf{w}_3 - \mathbf{W}_4\mathbf{d} = \mathbf{b}(\mathbf{s}) \\
& \mathbf{D}^T\mathbf{w}_1 - \mathbf{W}_5^T\mathbf{1} = \mathbf{0} \\
& \mathbf{W}_4\mathbf{D} + \mathbf{H}\mathbf{W}_5 = \mathbf{A} \\
& \mathbf{w}_1, \mathbf{w}_2, \mathbf{W}_5 \geq 0 \\
& \mathbf{s} \in \mathcal{S}.
\end{aligned}
$$

C Figures

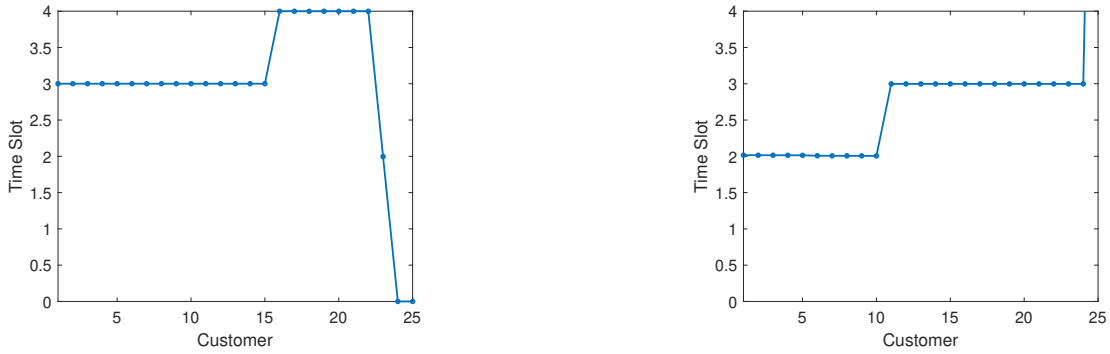
C.1 Analysis ball uncertainty



(a) Overtime cost c_O *(b) Waiting time cost c_W* *(c) Idle time cost c_I*

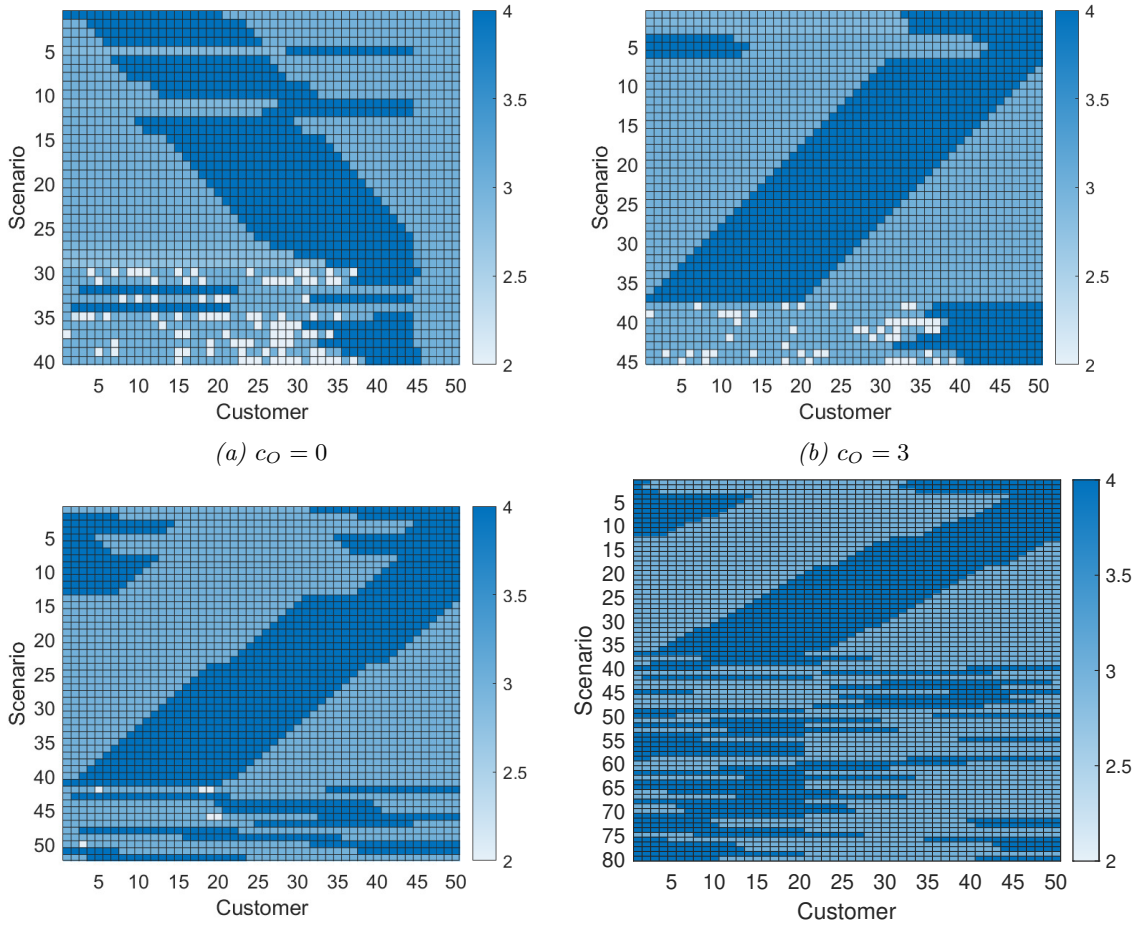
Figure 9: Schedules for varying cost parameters for ball uncertainty

C.2 Analysis cost parameters budget uncertainty



(a) $c_O = 3000$ *(b) $c_I = 1000$*

Figure 10: Schedule for $N = 25$ with extreme costs



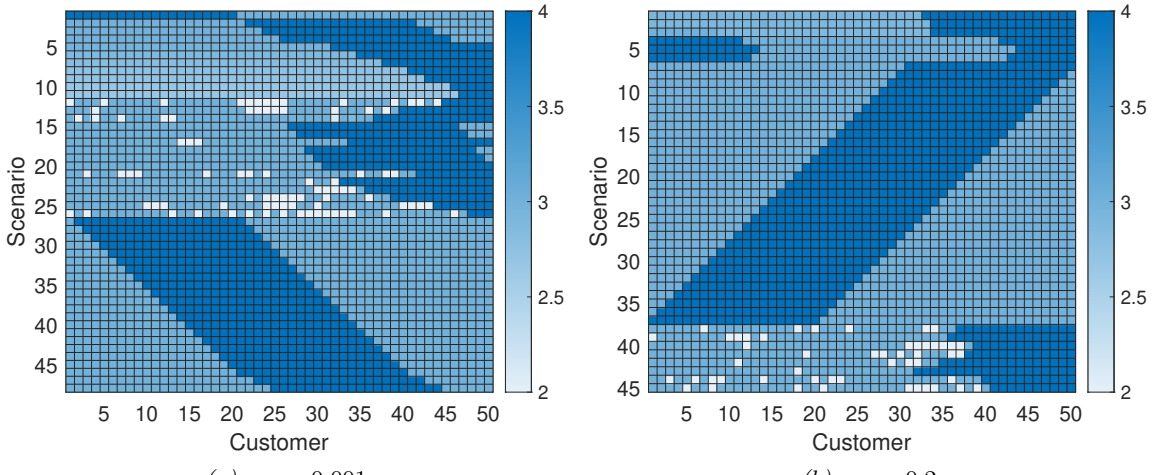
(a) $c_O = 0$

(b) $c_O = 3$

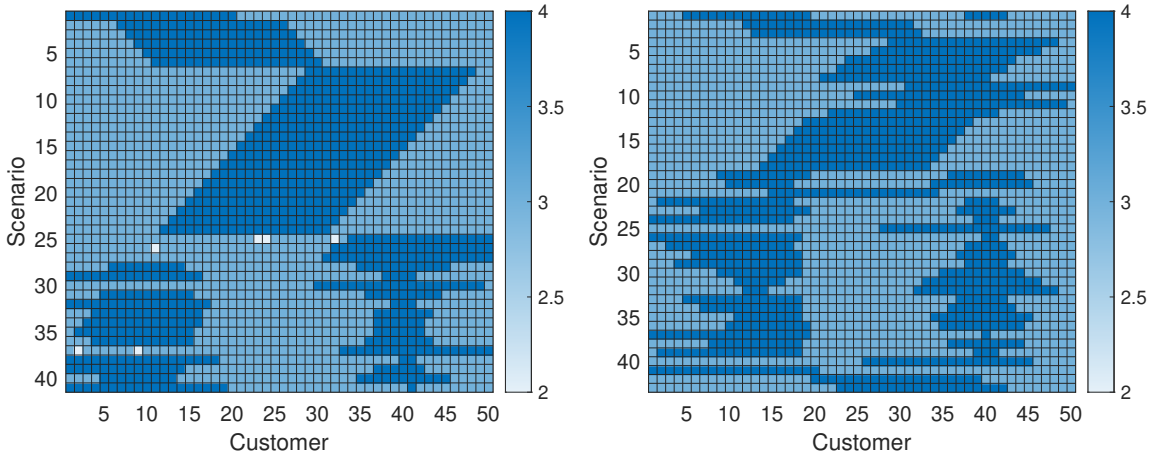
(c) $c_O = 15$

(d) $c_O = 75$

Figure 11: Worst case scenarios for varying c_O

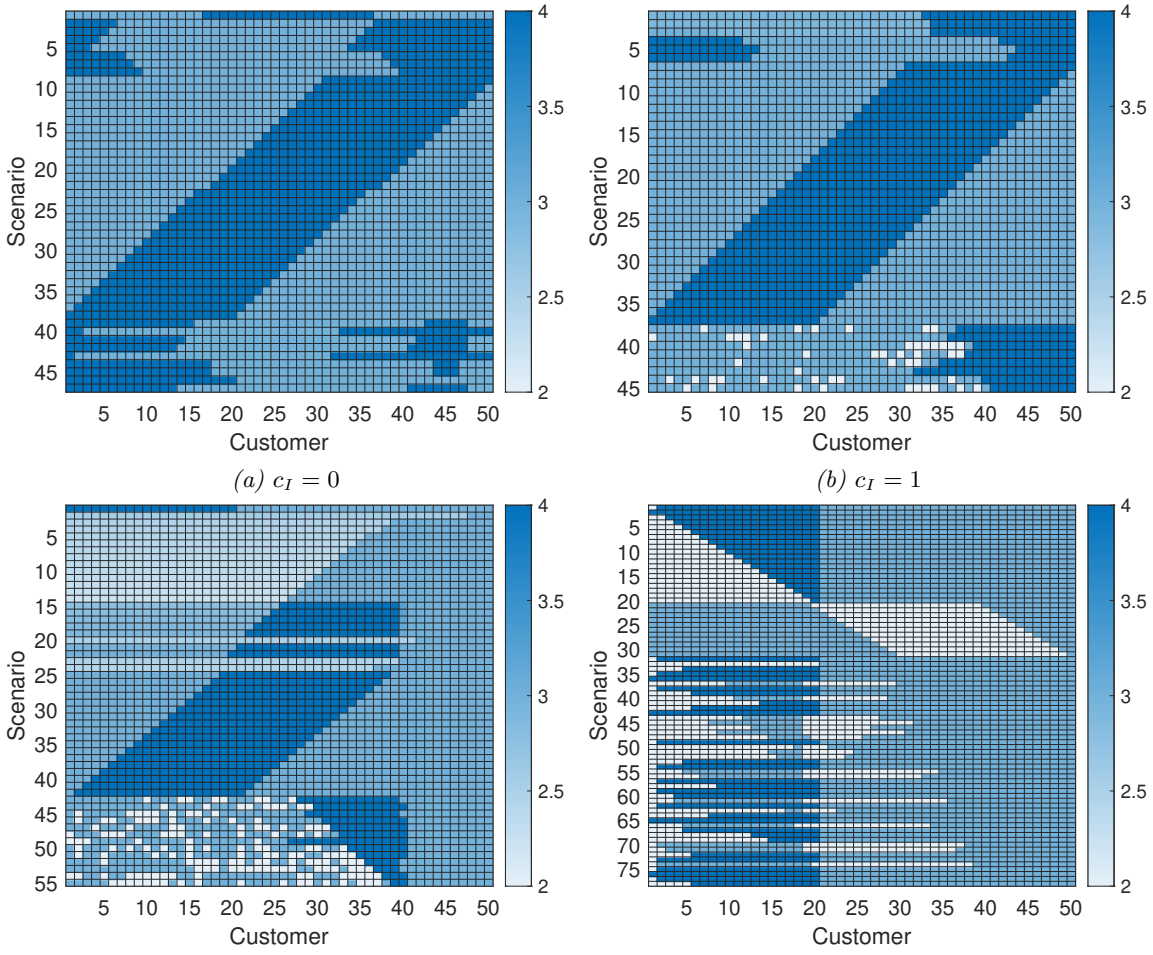


(a) $c_W = 0.001$ *(b)* $c_W = 0.2$



(c) $c_W = 1$ *(d)* $c_W = 5$

Figure 12: Worst case scenarios for varying c_W



(a) $c_I = 0$ *(b)* $c_I = 1$

(c) $c_I = 5$ *(d)* $c_I = 25$

Figure 13: Worst case scenarios for varying c_I