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, with the goal of minimizing the weighted costs of waiting time, idle time, and overtime. Previous studies employing stochastic programming were limited to small instances or constrained by restrictive assumptions. We introduce a novel Robust Optimization (RO) approach that considers service times within a specified uncertainty set and minimizes the worst-case costs, necessitating the minimization of a convex function. By integrating advanced methods from Robust Convex Optimization, such as the Reformulation-Perspectification Technique (RPT) and a cutting-set approach, we are the first to establish an exact solution procedure for determining optimal schedules. Our robust framework for ASP is designed to handle large instances and accommodates general convex uncertainty sets. Based on extensive numerical experiments with both polyhedral and ellipsoidal uncertainty sets, and synthetic problem instances involving up to 100 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 becoming increasingly crucial in a world where online booking systems are more prevalent and customer expectations for superior service remain high. Efficient appointment scheduling is essential for optimizing resource utilization and minimizing delays in the system, thereby enhancing overall operational efficiency and service quality across various

[*]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

sectors, 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 compounded by uncertainties, such as the unknown duration of appointments. In healthcare, effective appointment scheduling is crucial 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, 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). The Appointment Scheduling Problem (ASP), first introduced by Bailey (1952), has garnered substantial attention from both researchers and practitioners. ASP involves scheduling a finite number of customers, each with uncertain service times, to be served consecutively by a single server within a fixed time horizon. The primary goal is to design a schedule that minimizes the total weighted costs associated with customer waiting time, and server idle time, and overtime. These scheduling decisions must be made in advance, despite the uncertain duration of services, adding significant complexity to the problem. Allocating overly short time slots can lead to subsequent delays and increased waiting times, diminishing service quality. Conversely, overly long slots can result in periods of underutilization and potentially unnecessary overtime, indicating inefficiencies in the system. Thus, the classic ASP is a stochastic optimization problem that necessitates a balance between these competing factors. It typically requires distributions of service times, a linear cost function that penalizes expected waiting and idle times, and queue dynamics describing waiting and idle times as functions of the appointment slots and service times.

Limitations of stochastic optimization. The classic ASP thus explores optimal scheduling solutions for a finite number of customers through stochastic optimization (SO), with considerations extending to no-shows, multiple servers, and customer preferences. Comprehensive literature reviews focusing on healthcare services include Cayirli and Veral (2003), Gupta and Denton (2008), Ahmadi-Javid et al. (2017), and Ala and Chen (2022). Various methods like sample average approximation (SAA), quasi-gradient methods, and sequential bounding approaches, have been applied to address 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, employs a linear program to approximate the problem based on sampled scenarios of service times, analyzing average costs across

these samples. These traditional methods assume precise input data, fully specifying the service time distribution. Alternatively, Distributionally Robust Optimization (DRO) requires only partial knowledge of distributions, such as mean, moments, and range. DRO seeks to minimize worst-case expected costs among all distributions that align with this partial information. Various studies have approached versions of distributionally robust ASP (Kong et al. 2013, Mak et al. 2015, Qi 2016, Jiang et al. 2017, Padmanabhan et al. 2021, van Eekelen et al. 2024). Despite the effectiveness of SO and DRO techniques in developing schedules while facing uncertainties, they are not without limitations. Both approaches face the 'curse of dimensionality', where the complexity of the problem increases exponentially with the number of customers. This exponential increase generally limits the application of these methods to ASP instances with no more than 20 customers, underlining a significant challenge in scaling these models to larger problems. However, note that SAA, being an approximation method, can potentially extend beyond this range when the scenarios are sampled and managed properly. Similarly, DRO can accommodate larger instances with highly constrained ambiguity sets that simplify the underlying optimization problem. Despite these nuances, the broader issue of scalability persists with traditional stochastic methods, particularly under less restrictive problem configurations. This pervasive issue of scalability has driven the exploration and development of new methodologies based on robust optimization, which are designed to enhance efficiency and scalability for larger problem instances.

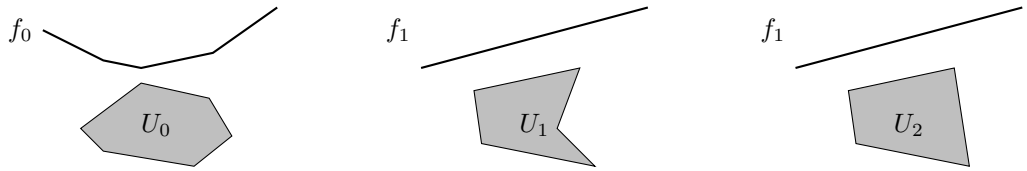
Robust optimization. In recent years, Robust Optimization (RO) has become a compelling approach for tackling complex optimization problems, renowned for its computational tractability and substantial modeling power. The concept of RO was initially introduced in the early 1970s by Soyster (1973) and has attracted considerable attention from the research community, especially since the 2000s. RO has found applications across various fields, including energy, logistics, finance, and healthcare; for detailed reviews, see Beyer and Sendhoff (2007) and Gabrel et al. (2014). The foundational principles of different categories of RO problems are thoroughly outlined in the seminal works of Ben-Tal et al. (2009) and Bertsimas and den Hertog (2022). Unlike traditional stochastic methods and DRO, RO tackles uncertainty by defining a specific uncertainty set where uncertain parameters are presumed to lie. The selection of an appropriate uncertainty set type – commonly box, ellipsoidal or polyhedral – is critical, as outlined in the practical guide by Gorissen et al. (2015). Note that RO does not rely on probability distributions for uncertain parameters but rather can incorporate insights from asymptotic results of probability theory, like the central limit theorem, as discussed in (Bandi and Bertsimas 2012). The primary goal of RO is to identify solutions that offer optimal performance under the worst-case scenarios encapsulated within these uncertainty sets. The core advantage of RO is its capability to convert problems with robust constraints, which must be satisfied for all possible values within the set, into tractable deterministic equivalents known as robust counterparts. This methodological transformation enables RO to effectively manage larger-scale problems and potentially overcome the challenges posed by high dimensionality.

Robust optimization for ASP. Although there have been notable efforts to apply RO to the ASP, these efforts often necessitate restrictive choices regarding uncertainty sets to preserve tractability. Mittal et al. (2014) explore the stochastic linear ASP model, focusing on uncertain service times confined within specific intervals, known as box uncertainty. By recognizing that the worst-case scenario consistently selects one of the interval's endpoints, they derive closed-form optimal schedules that effectively balance the maximum underage and overage costs. Building on this idea, Schulz and Udwani (2019) introduce heterogeneous underage costs, which leads to approximate solutions instead. Bauerhenne et al. (2024) extend the use of box uncertainty to include the probability of no-shows, aiming to minimize idle and overtime costs while ensuring waiting time guarantees. They tackle this by analytically determining worst-case scenarios for service times and no-shows, and then solving a mixed-integer linear program. Bandi and Gupta (2020) examine surgical scheduling within hospitals using a polyhedral uncertainty set, managing staffing and sequencing constraints to optimize surgery schedules. However, a critical analysis reveals two significant misconceptions in their approach, as further elaborated on in Appendix A. As the uncertainty is split over multiple inequalities by introducing definition variables (a common modeling error in RO, see Bertsimas and den Hertog (2022)), they do not solve the original problem with polyhedral uncertainty, but rather simplify the problem to box uncertainty. This highlights the challenges in applying RO to the ASP when dealing with more complex uncertainty sets. We aim to address these challenges by implementing and extending state-of-the-art RO methods suitable for general convex uncertainty sets.

Robust convex optimization. The primary challenge in developing a tractable RO framework for ASP stems from the convex nature of the cost objective with respect to the uncertain parameters. Unlike robust concave optimization, where tractable equivalent reformulations can be derived as demonstrated by Ben-Tal et al. (2015), robust convex optimization presents greater complexity, making such reformulations typically out of reach. Note that with robust convex optimization, we imply convexity in the uncertain parameters and variables (contrary to Ben-Tal et al. (1998) only dealing with convexity in decision variables), as well as a convex uncertainty set. This complexity often arises from the need to identify the worst-case scenario within the uncertainty set, essentially maximizing a convex function – a known difficult task (Selvi et al. 2022). For example for polyhedral uncertainty, where exploring all corner points becomes computationally prohibitive. Recent advancements by Bertsimas et al. (2023b) have introduced new opportunities, unifying and extending existing approaches on robust convex optimization, summarized in four key steps:

(0) Begin with a robust convex optimization problem (Figure 1a).

(1) Transform this into an equivalent bilinear problem with nonconvex uncertainty set (Figure 1b), where the function becomes computationally tractable but the uncertainty set complicates.



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

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

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

(3) Solve this using either the support function or the dual approach as demonstrated by Ben-Tal et al. (2015), yielding a conservative approximation for the optimal objective value and thereby establishing an upper bound in minimization contexts.

Furthermore, Bertsimas et al. (2023b) propose a method 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 this method is promising, several challenges persist when applying this framework to obtain both lower and upper bounds effectively. Given its generality, each step must be carefully tailored to the specific objectives and uncertainty sets of the problem at hand. This process can prove complex, as it requires fitting the problem setting in the notation of the framework, applying RPT to convex uncertainty sets, and deriving support functions. Additionally, obtaining exact robust solutions remains critical in many applications, which is where our extension using a cutting-set approach comes into play.

Structural properties of optimal schedules. A key methodological motivation of this study is to develop a computationally efficient robust optimization framework that can handle general uncertainty characterizations and large problem instances. Equally important is understanding the interplay between uncertainty and optimal schedules, providing insights for practitioners seeking to implement robust scheduling solutions. Previous studies have identified several structural properties 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 programming approaches, consistently observed a characteristic 'dome-shaped' pattern. These schedules allocate longer time slots to customers in the middle of the day, which shorter slots at the beginning and end. This pattern can be explained by non-stationary queueing theory: early delays propagate throughout the day, requiring longer slots to absorb them. However, allocating excessive time risks underutilization, so shorter slots at the start and end help reduce idle and overtime. As a result, placing longer slots in the middle balances idle and waiting times. The

dome-shaped pattern is a common feature in ASP literature that assumes independent service times. In contrast, studies on distributionally robust ASP, where service time distributions are only partially known, often identify 'decreasing' schedules, where time slots shorten over time. This results from worst-case scenarios clustering longer service times early in the schedule, requiring more time allocation at the start to prevent excessive delays. These decreasing schedules contrast with Bailey's scheduling rule, which uses increasing schedules to balance idle and waiting times. Recent research by van Eekelen et al. (2024) using DRO methods with independent service times found schedules that resemble Bailey's rule, with minimal early slots. Furthermore, these schedules shift from dome-shaped to increasing when overtime costs rise. Similarly, Benjaafar et al. (2023) found optimal increasing schedules when incorporating service level constraints.

This paper's contributions. In this paper, we develop a general framework for the ASP to determine robust schedules under service time uncertainty. Our key contributions are twofold:

- *Methodological innovations:* We introduce the first exact method for the robust ASP that accommodates uncertain service durations within general convex uncertainty sets. We show that the work of Bandi and Gupta (2020) presenting a method for polyhedral uncertainty sets is incorrect in Appendix A. Our method, based on the robust convex optimization techniques in Bertsimas et al. (2023b) determining robust cost bounds, is enhanced with a cutting-set approach to obtain exact solutions. It minimizes the worst-case weighted sum of waiting, idle, and overtime costs, effectively managing larger-scale instances scalable up to 100 customers. We demonstrate the application of this framework to both polyhedral and ellipsoidal uncertainty sets.
- *Practical implications and insights:* We explore the practical implications of uncertainty within the ASP and its impact on scheduling decisions. Our analysis provides a nuanced look at how the type and magnitude of service time uncertainty, the cost function, and the structural characteristics of optimal schedules interplay. Specifically, we investigate budget and ball uncertainties to show that various optimal robust schedules can emerge under different conditions, providing insights into when schedules are decreasing, dome-shaped or increasing.

Outline. The remainder of this paper is structured as follows. Section 2 formally introduces the robust Appointment Scheduling Problem. In Section 3, we present our general framework for solving the robust ASP, detailing the processes for deriving upper bounds, lower bounds, and exact solutions. Sections 4.1 and 4.2 demonstrate the application of this framework to polyhedral and ellipsoidal uncertainty sets, respectively. Section 5 discusses the results, analyzing optimal robust schedules and worst-case scenarios under budget and ball uncertainty and providing computational insights, including cost bounds and running times. Finally, Section 6 concludes with a summary of our findings and suggestions for future research.

2 Robust appointment scheduling

Consider a service system with a single server required to serve N customers in order of arrival within a fixed planning horizon T. 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 constrained within a predefined convex uncertainty set $\mathcal{X}$, i.e. $\mathbf{x} \in \mathcal{X}$. This set is defined by inequalities $g_i(\mathbf{x}) \leq 0$ for $i = 1, \ldots, K$, where each g_i is a convex function mapping $\mathbb{R}^N$ to $\mathbb{R}$. The goal is to determine the optimal schedule $\mathbf{s} = (s_1, \ldots, s_N)$, where s_n denotes 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 that the total time does not exceed T (equivalently, we can stipulate that $\sum_{n=1}^{N} s_n = T$, assigning any remaining time to the last customer as needed). This defines the feasible schedule set as $\mathcal{S} = \{\mathbf{s} \ : \ \mathbf{s} \geq 0, \ \sum_{n=1}^{N} s_n = T\}$. In this standard model, customers are assumed to show up and arrive punctually. Extensions such as no-shows or patient lateness can be effectively modeled within our framework by expanding the uncertainty parameters and choosing appropriate uncertainty sets, such as setting a cap on the number of potential no-shows.

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. Therefore, we define the cost objective of the robust ASP as follows:

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

Let w_n denote the waiting time of customer n, such that the total waiting time is the sum over all customer waiting times $W = \sum_{n=1}^{N} w_n$. These waiting times can be calculated recursively, starting with $w_1 = 0$ and for $n = 2, \ldots, N$ using $w_n = (w_{n-1} + x_{n-1} - s_{n-1})^+$, stating that the waiting time of customer n is that of the previous customer, plus the service time minus the time slot of the previous customer (nonnegative). By induction, this can be expressed as $w_n = \max_{1 \leq l \leq n-1} \left\{0, \sum_{j=n-l}^{n-1} (x_j - s_j)\right\}$.

The idle time I reflects the total gaps between consecutive services and can be calculated as the difference between the starting time of the last appointment (his appointment time plus his waiting time, $\sum_{j=1}^{N-1} s_j + w_N$) and the cumulative service time of all customers before that ($\sum_{j=1}^{N-1} x_j$). Lastly, the overtime O is calculated as any excess working time beyond T. This can be conceptual-

ized as the waiting time of a hypothetical last customer $N+1$ scheduled at time T. Hence,

$$W = \sum_{n=1}^{N} \max_{1 \le l \le n-1} \left\{0, \sum_{j=n-l}^{n-1} x_j - s_j\right\},\ I = \max_{1 \le l \le N-1} \left\{0, \sum_{j=1}^{l} s_j - x_j\right\},\ O = \max_{1 \le l \le N} \left\{0, \sum_{j=N+1-l}^{N} x_j - s_j\right\}$$

Note that as a result, the cost function (1) 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.

The goal of the robust ASP is to identify a feasible schedule that minimizes the worst-case costs across all possible realizations of the service times within the uncertainty set. This min-max problem can be restated using epigraph notation:

$$\min_{\mathbf{s} \in S} \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{s}, \mathbf{x}) \iff \min_{\mathbf{s} \in \mathcal{S},\, \tau} \tau \quad \text{s.t. } f(\mathbf{s}, \mathbf{x}) \le \tau \ \ \forall \mathbf{x} \in \mathcal{X}. \tag{2}$$

As extensively discussed in Section 1, this formulation of the robust ASP includes a robust constraint that is convex with respect to the uncertain parameters. Solving this robust convex optimization problem therefore presents significant challenges.

3 General robust framework

This section introduces a general robust framework for the robust ASP formulated in (2). Our framework establishes an upper bound for the robust costs utilizing RPT and calculates a lower bound through a strategically chosen scenario set. Using a cutting-set approach, it ultimately identifies the exact robust costs and optimal robust schedule.

3.1 Upper bound

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

Step 1: Exact bilinear reformulation. First, we identify the sum-of-max of linear (SML) structure in the cost function in (1), where each of the $N+1$ terms is the maximum of a set of linear functions, denoted by the index set I_n:

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

This structure confirms that $f(\mathbf{s}, \mathbf{x})$ is convex. Each term, $y_{ni} = \mathbf{a}_{ni}^T \mathbf{x} + b_{ni}(\mathbf{s})$, is affine, defined by constant vectors $\mathbf{a}_{ni}$ and linear functions $b_{ni}(\mathbf{s})$ dependent on the decision variables $\mathbf{s}$. This structural information is encapsulated in the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ and a vector function $\mathbf{b}(\mathbf{s}) \in \mathbb{R}^M$, where $M = \mathcal{O}(N^2)$ reflects the total number of linear terms. The 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 C.1, with an illustrative example for clarity.

For the SML function $h(\mathbf{y})$, the convex conjugate, $h^*(\mathbf{w})$, equals zero, with its domain, dom h^*, defined as $\left\{\mathbf{w} \mid \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\right\}$ (a detailed proof is provided in Appendix B.1). Here, $\mathbf{H} \in \{0,1\}^{M \times (N+1)}$ assigns each linear function m to its respective maximum operator n, with $h_{in} = 1$ if $i \in I_n$ and 0 otherwise. Utilizing Proposition 5 in Bertsimas et al. (2023b) on the bilinear reformulation, one can derive that the constraint in (2) is equivalent to

$$\sup_{(\mathbf{w},\mathbf{x},\mathbf{V}) \in \Theta} \left\{ \mathrm{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \right\} \leq \tau, \tag{4}$$

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

Thus, the reformulated robust constraint becomes linear in terms of both $\mathbf{x}$ and $\mathbf{s}$, although the uncertainty set is nonconvex due to the bilinear equality constraints.

Step 2: Convex outer approximation. In the second step, we construct a safe approximation of (4) by taking a convex outer approximation of the uncertainty set. Initially, we define a preliminary uncertainty set Θ_0 by omitting the nonconvex bilinear constraints $\mathbf{V} = \mathbf{w}\mathbf{x}^T$ from Θ:

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

This use of Θ_0 in (4) yields an upper bound by safely expanding the original set, as $\Theta \subset \Theta_0$. However, this approximation may not be tight due to the exclusion of the bilinear equality constraints, resulting in a significant loss of information.

To improve this approximation while preserving convexity of the set, we apply the Reformulation-Perspectification Technique (RPT) from Zhen et al. (2021), that introduces additional linear constraints to recapture some lost information. This technique involves multiplying each constraint in $\mathcal{X}$, $g_i(\mathbf{x}) \leq 0$ 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 linear constraint by considering the perspective function and substituting the terms $\mathbf{V} = \mathbf{w}\mathbf{x}^T$. The exact implementation of the RPT depends on the form of $g_i(\mathbf{x})$. Furthermore, we multiply the parameter $\mathbf{x}$ directly with the constraint $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ and apply the same substitution. This enhancement results in an improved convex uncertainty set Θ_1,

which integrates Θ_0 with these 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 (4) allows for a tighter safe approximation than using Θ_0, as $\Theta \subseteq \Theta_1 \subseteq \Theta_0$:

$$\sup_{(\mathbf{w},\mathbf{x},\mathbf{V})\in\Theta_1} \mathrm{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \leq \tau. \tag{5}$$

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

Step 3: Solving robust linear problem. Integrating the new robust constraint (5) into the robust optimization problem (2) results in:

$$\min_{\mathbf{s}\in\mathcal{S},\,\tau} \ \tau \quad \text{s.t.} \quad \mathrm{Tr}(\mathbf{A}^T\mathbf{V}) + \mathbf{b}(\mathbf{s})^T\mathbf{w} \leq \tau \quad \forall(\mathbf{w}, \mathbf{x}, \mathbf{V}) \in \Theta_1. \tag{6}$$

This formulation is advantageous because it presents a linear problem subject to a convex uncertainty set, allowing for traditional robust optimization techniques. There are primarily two non-iterative approaches to solve this problem: computing the support function of the uncertainty set (Ben-Tal et al. 2015) and deriving the dual 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 $\mathbf{z} = \Big[\mathbf{w};\ \mathbf{x};\ \mathbf{V}_1;\ \ldots;\ \mathbf{V}_N\Big]$, where $\mathbf{V}_n$ is the n-th column of $\mathbf{V}$. We define vector $\mathbf{c}(\mathbf{s}) = \Big[\mathbf{b}(\mathbf{s});\ \mathbf{0};\ \mathbf{a}_1;\ \cdots;\ \mathbf{a}_N\Big]$, allowing us to reformulate the robust constraint as

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

where δ^* is the support function of Θ_1, i.e. the convex conjugate of the indicator function for Θ_1. The simplified version of the upper bound problem in (6) is therefore

$$\min_{\mathbf{s}\in\mathcal{S},\,\tau} \ \tau \quad \text{s.t.} \quad \delta^*(\mathbf{c}(\mathbf{s}) \mid \Theta_1) \leq \tau. \tag{7}$$

This model is easily solved using standard solvers, with the complexity dependent on the structural properties of δ^* (detailed in Bertsimas and den Hertog (2022) for various uncertainty sets).

Dual problem: Alternatively, the robust problem can be solved using its dual formulation. We ensure that the optimization variables $\mathbf{s}$ and τ are explicitly defined, incorporating that $\mathbf{b}(\mathbf{s}) = -\mathbf{A}\mathbf{s}$. We introduce a constant $\phi = 1$, forcing uncertainty solely on the left-hand side of the first constraint.

The primal formulation is structured as

$$\begin{aligned} \max_{\mathbf{s},\tau,\rho} \ & -\tau \\ \text{s.t.} \ & -\mathbf{w}^T\mathbf{A}\mathbf{s} - \tau + \rho\,\mathrm{Tr}(\mathbf{A}^T\mathbf{V}) \leq 0 \quad \forall (\mathbf{w},\mathbf{x},\mathbf{V}) \in \Theta_1, \\ & \mathbf{1}^T\mathbf{s} \leq T,\ \mathbf{s} \geq 0,\ \phi = 1. \end{aligned} \tag{8}$$

Beck and Ben-Tal (2009) demonstrate that the dual of a robust problem is derived by dualizing the deterministic problem and then substituting the $\forall$-quantifier in the primal with an $\exists$-quantifier in the dual. Therefore, we introduce dual variables λ_1, λ_2 and λ_3, corresponding to each of the constraints in the primal formulation. Notably, $\lambda_1 = 1$ and $\lambda_3 = -\mathrm{Tr}(\mathbf{A}^T\mathbf{V})$ can be immediately identified and eliminated, reducing the focus to $\lambda := \lambda_2$. The dual formulation of (6) then becomes:

$$\begin{aligned} \min_{\lambda,\mathbf{w},\mathbf{x},\mathbf{V}} \ & T\lambda - \mathrm{Tr}(\mathbf{A}^T\mathbf{V}) \\ \text{s.t.} \ & -\mathbf{A}^T\mathbf{w} + \mathbf{1}\lambda \geq 0,\ \lambda \geq 0, \\ & \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1},\ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T, \\ & \mathbf{x} \in \mathcal{X},\ w_j g_i(\mathbf{x}) \leq 0 \quad \forall i = 1,\ldots,K, j = 1,\ldots,M. \end{aligned} \tag{9}$$

This reformulated problem is suitable for standard optimization solvers, with its complexity depending on the structure of the functions g_i' and the number of customers (N) and constraints in Θ (K), as the number of variables and constraints scales to $\mathcal{O}(N^3)$ and $\mathcal{O}(KN^2)$, respectively. This method offers a straightforward implementation advantage as is circumvents the need to derive a support function. Importantly, the dual of this dual problem is equivalent to Problem (7).

Result 1. *Solving Problem* (7) *that contains the support function, or the dual formulation* (9)*, results in an upper bound for the robust ASP objective in* (2)*.*

3.2 Lower bound

Initial scenario set. To establish a lower bound for the robust ASP, we initially select a finite subset of scenarios, $\tilde{\mathcal{X}}$, from the full uncertainty set $\mathcal{X}$. The quality of the lower bound depends significantly on the selection of these scenarios. Our proposed selection leverages the information of dual problem's optimal solutions, specifically $\mathbf{V}^*$, $\mathbf{x}^*$, and $\mathbf{w}^*$ (which can also be obtained by considering the dual variables of the constraints in Problem (7)). We deduce potential scenarios using that the relationship $\mathbf{V}^* = \mathbf{w}^*\mathbf{x}^{*T}$ ideally holds, so each row $\mathbf{V}_m^{*T}$ should satisfy $\mathbf{V}_m^{*T} = w_m\mathbf{x}^{*T}$ (although this is not explicitly enforced by the model). Potential scenarios $\tilde{\mathbf{x}}_m$ are therefore derived by calculating $\mathbf{V}_m^*/w_m$, which forms the preliminary scenario set $\tilde{\mathcal{X}}$. We then filter this set to include

only unique scenarios that reside within $\mathcal{X}$, resulting in the refined scenario set $\bar{\mathcal{X}}$. Hence,

$$\tilde{\mathcal{X}} = \left\{ \mathbf{x}^*, \frac{\mathbf{V}_1^*}{w_1}, \ldots, \frac{\mathbf{V}_N^*}{w_N} \right\} \quad \text{and} \quad \bar{\mathcal{X}} = \left\{ \tilde{\mathbf{x}}_m \in \tilde{\mathcal{X}} \mid \tilde{\mathbf{x}}_m \in \mathcal{X} \right\}. \tag{10}$$

Imposing a constraint for each scenario in $\bar{\mathcal{X}}$, we solve the problem

$$\min_{\mathbf{s} \in \mathcal{S}, \tau} \ \tau \quad \text{s.t.} \quad f(\mathbf{s}, \mathbf{x}) \leq \tau \quad \forall\, \mathbf{x} \in \bar{\mathcal{X}}. \tag{11}$$

This problem, featuring up to $M = \mathcal{O}(N^2)$ convex constraints, is straightforward to solve. Since $\bar{\mathcal{X}} \subset \mathcal{X}$, the solution provides a lower bound for Problem (2).

Adversarial approach. To improve the scenario set and derive a tighter lower bound, we employ an adversarial approach for scenario selection combined with a hill climbing method for scenario improvement. The process begins with an initially empty set, $\bar{\mathcal{X}}_0$. We start by selecting the scenario from $\bar{\mathcal{X}}$ that incurs the highest cost when evaluated against the upper bound schedule, which is based purely on numerical computations. In subsequent iterations, the optimal schedule is recalculated using the current set $\bar{\mathcal{X}}_0$. We then augment $\bar{\mathcal{X}}_0$ by adding new scenarios that, under the revised schedule, produce the highest costs. This iterative enhancement of $\bar{\mathcal{X}}_0$ continues until the inclusion of additional scenarios ceases to increase the costs.

Hill climbing. Next, we apply hill climbing to refine the scenarios in $\bar{\mathcal{X}}_0$, resulting in an improved set $\bar{\mathcal{X}}_1$. Inspired by the methods of Zhen et al. (2021) and Bertsimas et al. (2023b), we fix the schedule $\mathbf{s}'$ from the adversarial phase and address the following bilinear form of Problem (4):

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

Initially, we fix scenario $\mathbf{x}'$ from $\bar{\mathcal{X}}_0$ and find an optimal $\mathbf{w}'$ within dom h^*. This simplifies to:

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

For each n, the optimal $\mathbf{w}_n$ assigns $w_{ni} = 1$ for the index i^* that corresponds to the maximum value of $(\mathbf{A}\mathbf{x}' + \mathbf{b}(\mathbf{s}'))_n$, and $w_{ni} = 0$ otherwise. Let $\mathbf{e}_i$ be the i-th unit vector, then $\mathbf{w}'_n = \mathbf{e}_{i^*}$.

With this $\mathbf{w}'$ established, we improve $\mathbf{x}''$ by optimizing a linear problem over convex set $\mathcal{X}$:

$$\mathbf{x}'' = \arg\max_{x \in \mathcal{X}} \mathbf{w}'^T \mathbf{A}\mathbf{x} + \mathbf{b}(\mathbf{s}')^T \mathbf{w}' = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathbf{w}'^T \mathbf{A}\mathbf{x}. \tag{12}$$

This problem can be efficiently solved, and for some $\mathcal{X}$ it even has an explicit solution.

This iterative process alternates between refining $\mathbf{w}$ and $\mathbf{x}$ until no further improvements are observed in the objective value. Each scenario in $\bar{\mathcal{X}}_0$ undergoes this process, yielding an improved scenario to include in $\bar{\mathcal{X}}_1$. This refined set is then utilized to solve the deterministic problem (11), providing a tighter lower bound.

Result 2. *Solving the scenario-based Problem* (11) *results in a lower bound for the robust ASP objective in* (2)*. The initial scenario set* (10)*, that uses the results from Result 1, can be improved by the adversarial approach and hill climbing.*

3.3 Exact solution

To derive the exact robust costs and corresponding optimal robust schedule for Problem (2), we employ a cutting-set approach. This method iteratively refines the lower bound by methodically expanding the scenario set $\bar{\mathcal{X}}_1$, one scenario at a time (Bienstock and Özbay (2008), Bertsimas et al. (2016)). The method starts with a fixed schedule $\mathbf{s}'$, obtained from the lower bound calculations, and identifies the worst-case scenario $\mathbf{x}$ from the uncertainty set to add to the expanded set $\bar{\mathcal{X}}_2$:

$$\mathbf{x}' = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{s}', \mathbf{x}). \tag{13}$$

Due to the presence of max operators in f, this maximization problem is transformed into a mixed integer linear program (MILP), necessitating the introduction of binary variables and additional constraints. The details of this full MILP formulation are provided in Appendix D.1.

Then we recompute the optimal schedule $\mathbf{s}$ and costs for the expanded scenario set $\bar{\mathcal{X}}_2$ by solving the following scenario-based problem, which is then used as basis for the next iteration of (13):

$$\mathbf{s}' = \arg\min_{\mathbf{s}\in\mathcal{S}} \max_{\mathbf{x}\in\bar{\mathcal{X}}_2} f(\mathbf{s}, \mathbf{x}). \tag{14}$$

For each iteration, (13) and (14) result in a new upper and lower bound, respectively. The lower bound progressively increases as scenarios are added to $\bar{\mathcal{X}}_2$. On the other hand, the upper bound, which reflects the cost of the worst-case scenario based on a feasible yet potentially suboptimal schedule, may not consistently decrease with each iteration. The iterative process terminates once the difference between these two bounds contracts below a predetermined precision level, ϵ. Convergence is guaranteed as per Section 5.2 in Mutapcic and Boyd (2009), given that:

- The master problem (14) is solved exactly.
- The subproblem (13) of finding the worst-case scenario is solved exactly.
- The function $f(\mathbf{s}, \mathbf{x})$ is uniformly Lipschitz continuous in $\mathbf{s}$ (see proof in Appendix B.2).
- The feasible set $\mathcal{S}$ is bounded.

A pseudo-code for computing the exact robust costs and schedule is given in Appendix D.2.

Result 3. *The scenario set from Result 2 can be expanded by iteratively solving Problem* (14)*, obtaining a new optimal schedule, and Problem* (13)*, obtaining a worst-case scenario to add to the set. This process converges and results in the exact solution for the robust ASP in* (2)*.*

In summary, the method described above is generally applicable to any convex uncertainty set. However, tailoring it to a specific uncertainty set involves several crucial steps. For the upper bound, the RPT used in Step 2 must be specifically adapted to the convex constraints unique to the given uncertainty set. This customization directly impacts Step 3, affecting both the derivation of the support function and the formulation of constraints within the dual problem. For the lower bound, the complexity of the hill-climbing method is contingent on the characteristics of the uncertainty set, with some sets allowing for analytical solutions. The time required to determine the exact solution is also highly dependent on the complexity and dimensions of the uncertainty set.

4 Application to uncertainty sets

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

4.1 Polyhedral uncertainty

First, we consider polyhedral uncertainty for the service times $\mathbf{x}$, defined by a series of K linear inequalities: $\mathbf{x} \in \mathcal{X} = \{\mathbf{x} : \mathbf{D}\mathbf{x} \leq \mathbf{d}\}$, where $\mathbf{D} \in \mathbb{R}^{K \times N}$ and $\mathbf{d} \in \mathbb{R}^K$.

For the upper bound, we start with the reformulated constraint (4), and consider the preliminary uncertainty set: $\Theta_0^{\text{pol}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \mathbf{D}\mathbf{x} \leq \mathbf{d},\ \mathbf{w} \geq \mathbf{0},\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\}$. To refine this set, we apply RPT by multiplying the linear inequalities $\mathbf{D}\mathbf{x} \leq \mathbf{d}$ with $\mathbf{w} \geq \mathbf{0}$, and the equality $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ with $\mathbf{x}$, then substituting $\mathbf{V} = \mathbf{w}\mathbf{x}^T$. This results in an improved convex uncertainty set:

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

We then use this refined set to solve (6) 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 provided in Appendix C.2. The support function then simplifies to a linear system (Bertsimas and den Hertog 2022):

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

Integrating this into our optimization problem (7) and using the structure of $\tilde{\mathbf{D}}$, $\tilde{\mathbf{d}}$ and $\mathbf{y}(\mathbf{s})$, Appendix C.2 demonstrates how we obtain the following linear optimization problem, characterized by $\mathcal{O}(KN^2)$ decision variables and $\mathcal{O}(N^3)$ constraints:

$$\begin{aligned} \min_{\mathbf{s},\tau,\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3,\mathbf{W}_4,\mathbf{W}_5} \quad & \tau \\ \text{s.t.} \quad & \mathbf{d}^T\mathbf{w}_1 + \mathbf{1}^T\mathbf{w}_3 \leq \tau,\ -\mathbf{w}_2 + \mathbf{H}\mathbf{w}_3 - \mathbf{W}_4\mathbf{d} = \mathbf{b}(\mathbf{s}), \\ & \mathbf{D}^T\mathbf{w}_1 - \mathbf{W}_5^T\mathbf{1} = \mathbf{0},\ \mathbf{W}_4\mathbf{D} + \mathbf{H}\mathbf{W}_5 = \mathbf{A}, \\ & \mathbf{w}_1, \mathbf{w}_2, \mathbf{W}_5 \geq \mathbf{0},\ \mathbf{1}^T\mathbf{s} = T,\ \mathbf{s} \geq \mathbf{0}. \end{aligned} \tag{15}$$

The dual approach incorporates Θ_1^{pol} into (9), leading to the following linear problem:

$$\begin{aligned} \min_{\lambda,\mathbf{w},\mathbf{x},\mathbf{V}} \quad & T\lambda - (\text{Tr}(\mathbf{A}^T\mathbf{V}) + \alpha(\mathbf{w})) \\ \text{s.t.} \quad & -\mathbf{A}^T\mathbf{w} + \mathbf{1}\lambda \geq \mathbf{0},\ \lambda \geq 0, \\ & \mathbf{w} \geq \mathbf{0},\ \mathbf{H}^T\mathbf{w} = \mathbf{1},\ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T, \\ & \mathbf{D}\mathbf{x} \leq \mathbf{d},\ \mathbf{w}\mathbf{d}^T - \mathbf{V}\mathbf{D}^T \geq \mathbf{0}. \end{aligned} \tag{16}$$

The generation of a lower bound adheres to the general framework, with the scenario set $\bar{\mathcal{X}}$ being refined to to ensure all candidates meet the criteria $\mathbf{D}\mathbf{x} \leq \mathbf{d}$. Transitioning from $\bar{\mathcal{X}}_0$ to $\bar{\mathcal{X}}_1$ through hill climbing involves iterative computation of equations (**??**) and (12). Although this last optimization lacks an explicit solution for polyhedral uncertainty, finding a numerical solution remains straightforward due to the linearity of the problem over a polyhedron.

The exact solution process leverages the cutting set approach as outlined in the general framework, which converges without specific requirements on the set $\mathcal{X}$. Theoretically, an optimal solution could also 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. These points could then be used as the scenario set in model (11). However, the exponential increase in the number of extreme points with N makes this approach impractical, and thus our iterative approach is preferred.

Result 4. *Solving either Problem* (15) *or* (16) *results in an upper bound for the robust ASP objective in* (2) *with polyhedral uncertainty. Additionally, Results 2 en 3 yield a lower bound and an exact solution, respectively.*

4.2 Ellipsoidal uncertainty

For ellipsoidal uncertainty, the service times are defined by the set $\mathbf{x} \in \mathcal{X} = \{\mathbf{x} : (\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{Q}(\mathbf{x}-\bar{\mathbf{x}}) \leq \rho^2\} = \{\mathbf{x} : ||\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})|| \leq \rho\}$, 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 the preliminary uncertainty set as $\Theta_0^{\text{ell}} = \{(\mathbf{w}, \mathbf{x}, \mathbf{V}) \mid \|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho, \mathbf{w} \geq 0,\ \mathbf{H}^T\mathbf{w} = \mathbf{1}\}$. Applying RPT, we refine this set by multiplying the ellipsoidal constraint $\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho$ with $\mathbf{w} \geq \mathbf{0}$, and $\mathbf{H}^T\mathbf{w} = \mathbf{1}$ with $\mathbf{x}$, and substitute $\mathbf{V} = \mathbf{w}\mathbf{x}^T$. This results in the refined uncertainty set Θ_1:

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

We then aim to solve (6) using Θ_1. The first approach in Section 3.1 involves computation of the support function of Θ_1, which involves delving into the complexities of matrix-valued support functions (as discussed in Section 13.2 of Bertsimas and den Hertog (2022)) and composition rules. The second method utilizing the dual problem, offers a straightforward implementation that incorporates Θ_1^{ell} directly into the model in (9), resulting in the following tractable conic quadratic optimization problem with $\mathcal{O}(N^3)$ variables and $\mathcal{O}(N^2)$ constraints:

$$\begin{aligned} \min_{\lambda, \mathbf{w}, \mathbf{x}, \mathbf{V}}\ & T\lambda - \text{Tr}(\mathbf{A}^T\mathbf{V}) \\ \text{s.t.}\ & -\mathbf{A}^T\mathbf{w} + \mathbf{1}\lambda \geq 0,\ \lambda \geq \mathbf{0} \\ & \mathbf{w} \geq \mathbf{0},\ \mathbf{H}^T\mathbf{w} = \mathbf{1},\ \mathbf{H}^T\mathbf{V} = \mathbf{1}\mathbf{x}^T \\ & \|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho,\ \rho\mathbf{w} - \left(\|\mathbf{Q}^{\frac{1}{2}}(\mathbf{V}_m - w_m\bar{\mathbf{x}})\|\right)_{m=1}^{M} \geq \mathbf{0} \end{aligned} \tag{17}$$

Deriving a lower bound mirrors the general framework outlined earlier. Notably, the hill climbing technique benefits from the well-established 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 $\mathbf{x}''$ in (12) under the ellipsoidal norm constraint, eliminating the need for optimization and boosting efficiency:

$$\mathbf{x}'' = \arg \max_{\{\mathbf{x}: \|\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}})\| \leq \rho\}} \mathbf{w}'^T\mathbf{A}\mathbf{Q}^{-\frac{1}{2}}\mathbf{Q}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}}) = \rho\frac{\mathbf{Q}^{-1}\mathbf{A}^T\mathbf{w}'}{||\mathbf{Q}^{-\frac{1}{2}}\mathbf{A}^T\mathbf{w}'||} + \bar{\mathbf{x}}. \tag{18}$$

For exact solutions under ellipsoidal uncertainty, the procedure aligns with the established framework. Nonetheless, identifying the worst-case scenario in (13) now involves solving a conic quadratic optimization problem due to optimization over the ellipsoidal set, which is tractable, but generally more computationally expensive than solving the linear problem for polyhedral uncertainty.

Result 5. *Solving Problem* (17) *results in an upper bound for the robust ASP objective in* (2) *with ellipsoidal uncertainty. Additionally, Result 2, using the explicit formula* (18)*, yields a lower bound and Result 3 yields an exact solution.*

5 Computational results

Our general framework is applied to problems with specific polyhedral and ellipsoidal uncertainty sets. 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 RAM. Computations are conducted using MOSEK 10.0.27 and Gurobi 9.5.1, implemented through YALMIP R20210331 in MATLAB R2023b.

5.1 Set-up of numerical experiment

We illustrate the functionality of the robust ASP framework under budget uncertainty, which constrains each service duration within an interval while also imposing a bound on the total (absolute) deviation from nominal values. This form of uncertainty merges ∞-norm and 1-norm constraints, with parameters ρ and B, respectively. This is a special case of polyhedral uncertainty with $2N$ uncertain parameters, using auxiliary variable $\mathbf{w}$ to represent absolute deviations:

$$\mathcal{X} = \{\mathbf{x} : \ ||\mathbf{x} - \bar{\mathbf{x}}||_\infty \leq \rho, \ ||\mathbf{x} - \bar{\mathbf{x}}||_1 \leq B\} = \{\mathbf{x} : -\rho\mathbf{1} \leq \mathbf{x} - \bar{\mathbf{x}} \leq \rho\mathbf{1}, \ -\mathbf{w} \leq \mathbf{x} - \bar{\mathbf{x}} \leq \mathbf{w}, \ \mathbf{1}^T\mathbf{w} \leq B\}.$$

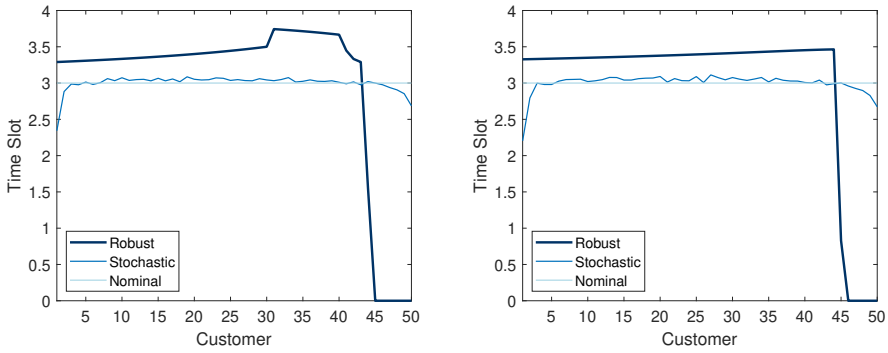
In our numerical experiment, we consider $N = 50$ customers, each with nominal service duration $\bar{\mathbf{x}} = 3$. The uncertainty bound $\rho = 1$ allows the durations to vary between 2 and 4. The total budget $B = \rho N/2.5 = 20$ implies that no more than 40% of customers can deviate to either 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. Result 4 prescribes how to solve the robust ASP with this budget uncertainty set. In particular, for the upper and lower bound calculations we use Mosek, while using Gurobi in determining exact solution.

5.2 Analysis of schedules

5.2.1 Base case

The optimal robust schedule from our numerical experiment is depicted in Figure 2a. Clearly, the schedule exhibits dome-shaped pattern which is typical for stochastic models, but with additional strategic scheduling nuances that enhance performance under budget uncertainty:

- *Increasing time slots for initial customers:* The schedule allocates progressively increasing time slots to the first 30 customers. This strategy allows for absorbing variability in service times early in the schedule, helping to prevent minor delays from cascading into significant waiting times for subsequent customers.

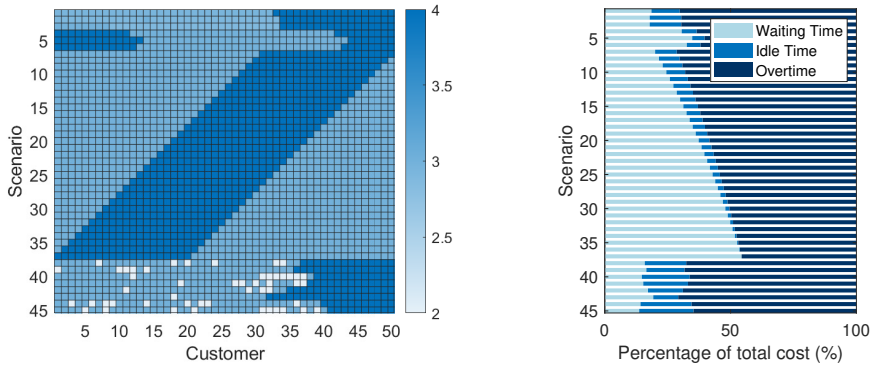


(a) Budget uncertainty *(b) Ball uncertainty*

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

- *Stabilization phase:* At customer 31, there is a noticeable increase in the time slots, which then stabilize for the next ten customers. This plateau is designed to manage any accumulated delays effectively, ensuring they do not disrupt the latter part of the schedule.
- *Reduction towards the end:* Approaching the schedule's end (after customer 40), there is a gradual reduction in the allocated time slots, critical to mitigate the risk of overtime.
- *Zero allocation final customers:* Starting from customer 45, time slots are reduced to zero, scheduling these appointments effectively at day's end. This approach accounts for unavoidable overtime due to the combined impact of budget uncertainty and the fixed time period.

Our analysis identifies 45 critical worst-case scenarios in the final scenario set $\bar{\mathcal{X}}_1$ (37 in the formation $\bar{\mathcal{X}}_1$ and 8 in the iterative cutting-set approach), all resulting in costs of 131.52. Despite the multitude of worst-case scenarios, a relatively small subset suffices to determine the optimal schedule. The heatmap in Figure 3a illustrates service durations across these scenarios, with darker colors indicating longer durations. Additionally, Figure 3b shows how total costs are distributed across waiting time, idle time and overtime. The heatmap reveals a tendency for service times to extend to maximal durations, heavily influencing overtime costs. More than half of the scenarios display these prolonged service times in a cluster, affecting the cost component balance: early clusters increase waiting times, while later ones contribute to more idle and overtime costs. Notably, in the last scenarios, extended service times towards end of day, combined with several short service periods, significantly elevate idle time costs. Nevertheless, idle time generally incurs the least expense, underscoring the server's time is effectively optimized within the robust schedule.



(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

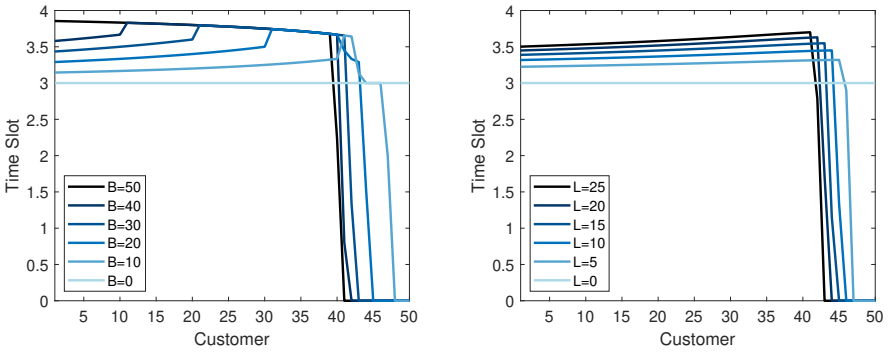
	budget: wc	budget: avg	ball: wc	ball: avg
nominal	218.00	24.87	193.90	30.48
SAA	217.38	23.33	194.33	28.67
robust	131.52	89.88	112.32	79.22

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

5.2.2 Comparison with stochastic solution

We compare the robust schedule's performance against two other strategies: the nominal schedule, which assigns the nominal service time of 3 to each customer, disregarding uncertainty, and a stochastic schedule derived using SAA with 10,000 scenarios for computational manageability, uniformly distributed across the support of the uncertainty set. Both schedules are depicted alongside the robust schedule in Figure 2a. To evaluate these strategies, we generate 50,000 new scenarios from the same distribution to calculate average costs. We also compute the exact worst-case costs for each schedule over the full uncertainty, as defined in (13). The results are given in Table 1.

The robust schedule significantly outperforms others in worst-case scenarios, capping the worst-case costs at 131.52, compared to 218.00 for the nominal schedule and 217.38 for the stochastic schedule. This demonstrates its effectiveness in handling uncertainties. However, this robustness incurs a higher average cost of 89.88, far exceeding that of the nominal (24.87) and stochastic (23.33) schedules. This trade-off between robustness and cost-efficiency suggests that while preparing for worst-case scenarios ensures safety against extreme uncertainties, it results in substantially higher average costs. Robust solutions are typically conservative, but they can be tailored to reduce conservatism and improve cost-effectiveness, as discussed in e.g. Roos and den Hertog (2020).



(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

5.2.3 Budget parameter

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

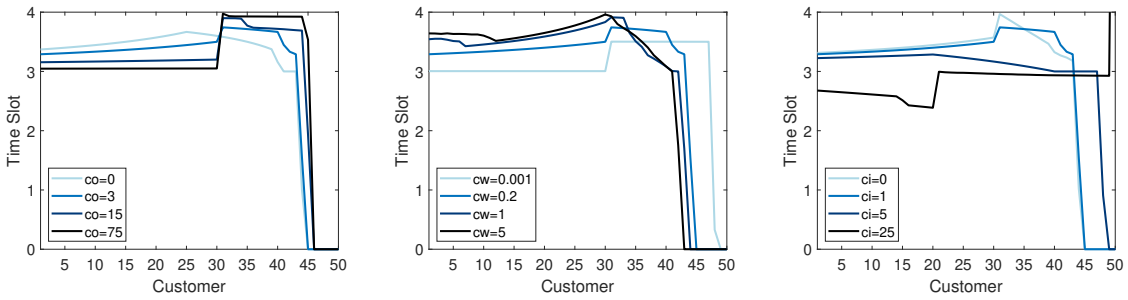
- $B = 50$: Under this box uncertainty setting, maximum deviation is allowed for each duration. This results in a decreasing schedule, which aligns with findings from Mittal et al. (2014) under similar uncertainty (equal if overtime costs are disregarded, so $c_O = 0$ and $T = \infty$).
- $B \in \{40, 30, 20, 10\}$: Decreasing B restricts the allowable uncertainty, leading to tighter scheduling. The schedule adapts by progressively reducing the time slots, especially for earlier appointments, to more closely reflects nominal values, leading to a dome-shaped schedule. Adjustments in later slots are tailored based on the remaining time in the day.
- $B = 0$: This setting, with no allowance for deviations, eliminates uncertainty, resulting in each customer receiving precisely their nominal service duration, producing a uniform schedule.

These findings illustrate a clear trend: larger uncertainty budgets require longer initial time slots to buffer against potential delays, while tighter budgets promote schedules that more closely adhere to nominal durations.

5.2.4 Cost parameters

To examine the impact of cost parameters on scheduling decisions, we modify their values relative to the base setting $c_O = 3$, $c_W = 0.2$, $c_I = 1$. Each parameter is adjusted independently, increased by 5-fold or 25-fold, or reduced to 0. The resulting schedules are given in Figure 5.

Overtime. Figure 5a demonstrates the effect of varying the overtime cost parameter c_O. At



(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

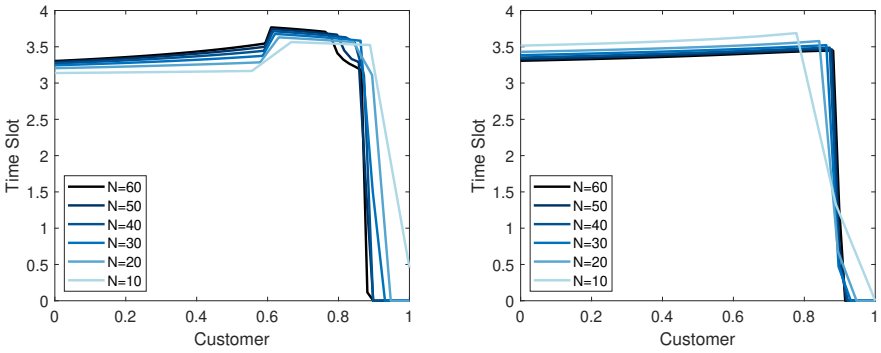
$c_O = 0$, the schedule exhibits a classical dome-shape. Increasing c_O to 3 (the base case) results in shorter early time slots, preserving more time for later appointments. At $c_O = 15$, initial time slots distribute more evenly, followed by a sharp increase. At $c_O = 75$, this flattening spans the entire schedule, progressively shifting towards nominal values for the first 30 customers and maximal values for the subsequent ones, leaving zero allocations towards the end. This becomes stark as $c_O \to \infty$, resulting in the time slot pattern $3 \to 4 \to 0$ (see Figure 10a for $c_O = 10,000$).

Waiting time. Figure 5b displays schedules for different waiting time costs. Starting with $c_W = 0.001$ – a negligible cost that guarantees a trade-off in the cost function and hence produces regular schedules – the first 30 customers receive nominal time slots, followed by an increase to 3.5 for the remainder. The base case at $c_W = 0.2$ gradually increases time slots for earlier customers, a trend that continues as c_W increases to 1 and then 5. Both start with a slight reduction in time slots, followed by a steady increase that peaks at the 30th customer and then declines sharply.

Idle time. Figure 5c reveals variability in scheduling based on idle time costs. Compared to the base case with $c_I = 1$, the absence of idle time penalties at $c_I = 0$ results in a more pronounced peak at customer 31, followed by a steep decline. Increasing c_I to 5 standardizes time slots, softening extreme allocations to nearly nominal values. At $c_I = 25$, early slots decrease significantly due to high idle costs, and the remaining slots approach nominal value. As $c_I \to \infty$, the schedule converges to $2 \to 3 \to \max$, with the last one receiving residual time (see Figure 10b for $c_I = 10,000$).

5.2.5 Number of customers

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



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

Figure 6: Optimal robust schedule with budget uncertainty for varying N.

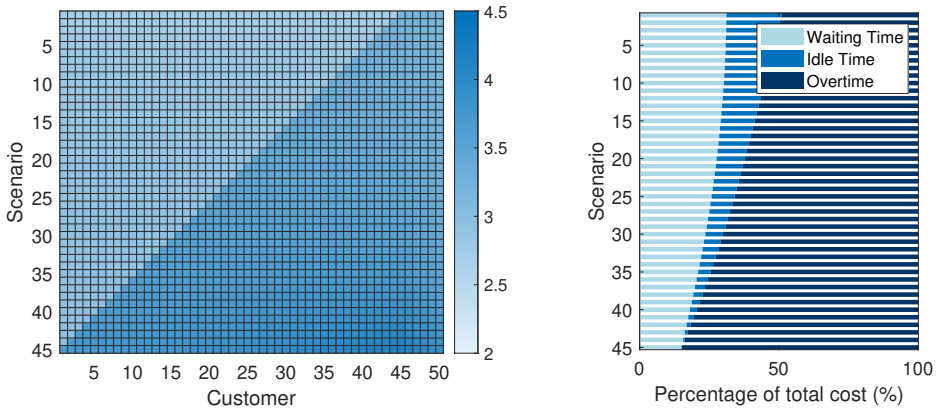
The schedules display a striking 'scale invariance', indicating that the general pattern remains relatively consistent despite changes in customer numbers. However, some differences are visible. The initial slots are slightly longer for larger values of N, increasing more rapidly early in the schedule, to absorb potential delays that could now effect many subsequent customers. Also, the plateau in the time slots is reached earlier and more abruptly in the sequence, followed by a steeper and earlier reduction towards zero as the fraction of served customers approaches 1, managing the smaller amount of remaining time in the day for later appointments.

5.2.6 Comparison with ball uncertainty

This section examines the implications of using budget uncertainty versus ball uncertainty for modeling service times in the robust ASP. The comparison aligns the geometric interpretations of these uncertainty sets by equating their volumes; details are provided in Appendix E.1, setting $Q \approx 0.094I$ and $\rho = 1$ in $\mathcal{X}$. Then Result 5 is applied for this particular ellipsoidal set.

The robust schedule under ball uncertainty, depicted in Figure 2b, displays a gradually increasing pattern from time slots of 3.3 to 3.5 by the 44th customer, before dropping to zero. Compared to the budget uncertainty schedule in Figure 2a, both schedules start very similar. However, the ball schedule consistently rises, whereas the budget schedule reaches a plateau and then decreases.

This difference stems from the intrinsic characteristics of the uncertainty sets. Ellipsoidal sets, being inherently smooth, generate worst-case scenarios without extremities, leading to smoother schedules, compared to budget uncertainty sets with non-smooth boundaries. The worst-case scenarios under ball uncertainty typically start with uniformly short service times and transition to longer duration later at any point during the day, as shown in Figure 7a. Consequently, an increasing schedule is rationalized to mitigate risk later in the day, where extended durations occur. The



(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

reduced extremity under ball uncertainty leads to a significant 15% decrease in worst-case costs, totaling 112.32. As seen in Figure 7b, this reduction is mainly due to lower waiting times, as scenarios often feature shorter service durations, also stabilizing overtime costs across scenarios.

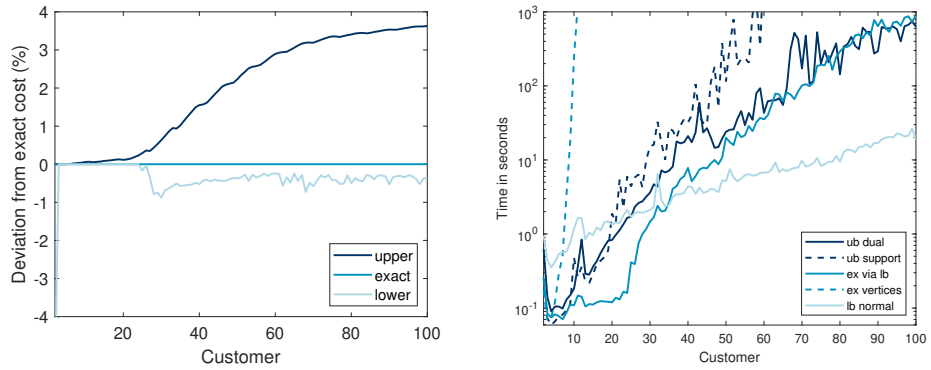
Table 1 compares the robust solution with the nominal and stochastic solutions, which are generated similarly as for budget uncertainty, illustrated in Figure 2b. The table reveals that the reductions in worst-case costs of the nominal and stochastic solutions are comparable in magnitude to those for the robust solution, affirming the robust solution's superiority in terms of managing risks. Furthermore, the robust solution under ball uncertainty maintains better average cost performance compared to its performance under budget uncertainty.

Figure 4b shows the dynamic adjustment of the schedule as the uncertainty parameter L varies. With L set to 0, eliminating uncertainty, a uniform schedule emerges. Increasing L mirrors the behavior for budget uncertainty, allocating more time to earlier appointments as a protective measure against higher uncertainty levels. Unlike budget uncertainty, the increment in time slots is nearly uniform across all customers, and could continue indefinitely without an upper bound on L.

5.3 Scalability and computational complexity

We evaluate 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. Figure 8a shows the percentage deviation of the upper and lower bounds from the exact robust costs as N ranges from 1 to 100. The lower bound closely aligns with the exact



(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

costs, showing exceptional accuracy up to $N = 25$. Even as N increases, the disparity between the lower bound and exact costs stays well below 1%, confirming the lower bound's effectiveness in approximating the true robust costs. Conversely, the upper bound begins to diverge more noticeably as N grows, particularly beyond $N = 35$, with deviations rising above 1% and trending upwards linearly, indicating reduced precision at larger scales. Nonetheless, the deviation seems to stabilize, reaching a plateau at a maximum of around 4%.

Running times. We assess the running times for various methods, depicted in Figure 8b using a logarithmic scale. These methods include the calculation of the upper bound (calculated via the support function as well as via the dual problem), the lower bound, and the exact method. For comparison, we also include the exact method that explores all vertices of the budget uncertainty set to demonstrate the efficiency of our framework.

For customer sets of $N \leq 20$, the running times for our four methods are comparably low, all staying under two seconds, whereas the vertex exploration method becomes highly impractical with more than 10 customers. As N increases, running times for all methods grow almost exponentially, yet they remain computationally feasible even up to $N = 100$. The upper bound calculated via the dual problem shows a particularly erratic increase in running times starting at $N = 30$, which suggests diminished efficiency at larger scales compared to the more consistent and faster performance of the upper bound via the support function. Remarkably, the exact method maintains a very stable running time across different customer sizes, as the number of scenarios to be added stays relatively constant. Overall, the lower bound method proves the most time-efficient for larger customer sets, with running times not exceeding 20 seconds for any N.

6 Conclusions and outlook

We have presented a robust framework for the ASP with uncertain durations, capable of dealing 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 that provably converges to the optimal robust schedule. Employing this framework for both polyhedral and ellipsoidal uncertainty sets, with instances of 50 customers, gave profound structural insights into how uncertainty in service times impacts optimal scheduling. For budget uncertainty sets, optimal schedules were shown to take a diverse array of dome-shaped patterns, similar to those observed 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 has proven scalable, handling instances up to 100 customers, and could be directly applied to real-world settings such as hospitals, provided that service time data is converted into appropriate convex uncertainty sets. These could be polyhedral or ellipsoidal, or general convex cones, information-based sets or intersections and sums of uncertainty sets. It is crucial to tailor these sets carefully to avoid overly conservative solutions. Further development could include additional ASP features such as no-shows, walk-ins, late arrivals, and variable capacity by expanding the uncertainty parameters. Moreover, incorporating service levels constraints or 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 a sum-of-max of linear function. 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 application of our robust framework 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., Bertsimas, D., and Youssef, N. (2015). Robust queueing theory. *Operations Research*, 63(3):676–700.

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., de Moor, D., den Hertog, D., Koukouvinos, T., and Zhen, J. (2023a). A novel algorithm for a broad class of nonconvex optimization problems. *Optimization Online*.

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

Bertsimas, D., den Hertog, D., Pauphilet, J., and Zhen, J. (2023b). 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.

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.

Jiang, R., Shen, S., and Zhang, Y. (2017). Integer programming approaches for appointment scheduling with random no-shows and service durations. *Operations Research*, 65(6):1638–1656.

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.

Roos, E. and den Hertog, D. (2020). Reducing conservatism in robust optimization. *INFORMS Journal on Computing*, 32(4):1109–1127.

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.

Selvi, A., Ben-Tal, A., Brekelmans, R., and den Hertog, D. (2022). Convex maximization via adjustable robust optimization. *INFORMS Journal on Computing*, 34(4):2091–2105.

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. J., den Hertog, D., and van Leeuwaarden, J. S. (2024). Distributionally robust appointment scheduling that can deal with independent service times. *Production and Operations Management*, 0(0).

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 Error in Bandi and Gupta (2020)

The work of Bandi and Gupta (2020) claims to address appointment scheduling under polyhedral uncertainty sets. However, a critical analysis reveals two significant misconceptions in their approach, suggesting they inadvertently simplify the problem to resemble box uncertainty instead.

In the original formulation of the ASP (as defined in Section 2), uncertainty in the durations is only present in the objective function, affecting the waiting times, idle times, and overtime. Contrary to this, Bandi et al. (2015) employs auxiliary variables within their objective function (Equation 8) and additional constraints to enforce correct values of the variables in the solution. This results in two sets of robust constraints involving the durations (Equations 14 and 24). Therefore, the uncertainty, originally only present in the objective, now is spread out across multiple inequalities in the model.

Although two nominal problems might be equivalent, their robust counterparts are not necessarily equivalent and thus can lead to different solutions. This discrepancy arises when the uncertainty in one constraint is split over multiple constraints, in particular when introducing definition variables (which is only allowed if we consider adaptive variables). This is a common misconception in RO that is further explained in Chapter 12.1 of Bertsimas and den Hertog (2022). This discrepancy is due to the non-disjoint nature of the uncertainty, and becomes apparent when recognizing that each uncertain duration is now involved in several constraints, allowing the worst-case to be selected differently each time. Having only one uncertain parameter in each 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 possible scenarios for each parameter are interdependent.

A second methodological error that further complicating the issue, appears when Bandi and Gupta (2020) introduces the dual variables $\mathbf{q}$ to obtain the robust counterpart (in Equation 19). The dual variables are uniformly chosen over all constraints $j = 1, \ldots, N_h$, which imposes an unnecessary restriction as the variables $\mathbf{q}_j$ are allowed to vary across inequalities j as required.

These two critical points underline that the approach taken by Bandi and Gupta (2020) aligns more closely with box uncertainty principles, as previously explored by Mittal et al. (2014). Ultimately, their approach does not provide a valid solution for robust appointment scheduling under polyhedral uncertainty.

B Proofs

In this appendix, we formally derive the convex conjugate and prove Lipschitz continuity of the sum-of-max-of-linear (SML) function.

B.1 Convex conjugate of SML function

Consider the maximum of linear functions, so $g(\mathbf{y}) = \max_i y_i$, where each y_i is a linear function. The convex conjugate is defined as $g^*(\mathbf{w}) = \sup_{\mathbf{y}}\{\mathbf{y}^T\mathbf{w} - \max_i y_i\}$. For $w_i \leq 0$, the supremum diverges to ∞ as $y_i \to \infty$. Likewise, when $\sum_i w_i < 1$ or $\sum_i w_i > 1$, the supremum becomes unbounded by driving all y_i to $-\infty$ or $+\infty$. Conversely, when $\sum_i w_i = 1$ and all $w_i \geq 0$, the supremum is 0, achieved by setting all $y_i = 0$. Thus, the value and domain of the convex conjugate:

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

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

$$h^*(\mathbf{w}) = \inf_{\{\mathbf{w}^n\}}\{\sum_n f_n^*(\mathbf{w}^n) | \sum_n \mathbf{w}^n = \mathbf{w}\} = 0 \;\text{ with }\; \text{dom } h^* = \{\mathbf{w} \mid \mathbf{w}_n \geq 0, \mathbf{1}^T\mathbf{w}_n = 1\,\forall n\}.$$

B.2 Lipschitz continuity of SML function

Here, we prove that the function $g(\mathbf{s}) = \sum_i \max_j\{\mathbf{a}_{ij}^T\mathbf{s} + b_{ij}\}$ on $\mathcal{S}$, being the sum of the maximum of linear functions in $\mathbf{s}$, is (uniformly) Lipschitz continuous. Then, for any two points $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}$, we evaluate the difference in function values:

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

where the first inequality follows from the triangle inequality, the second one from the norm properties of the infinity norm, and the last one from the Cauchy-Schwarz inequality.

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

C Derivations upper bound

In this appendix, we extensively derive two results used for the upper bound. We elaborate on the ASP cost function and write it in the form $h(\mathbf{A}\mathbf{x} + \mathbf{b}(\mathbf{s}))$, and we derive the linear optimization

problem using the support function for polyhedral uncertainty.

C.1 Cost function

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

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

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 rows $\mathbf{a}_{ni}$ and entries $b_{ni}(\mathbf{s})$, respectively, resulting in $M = \frac{1}{2}N^2 + \frac{5}{2}N = \mathcal{O}(N^2)$ rows. Furthermore, 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 costs 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 costs 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 costs. 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 costs, 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 by concatenating these rows and entries, with $M =$

$\frac{1}{2} \cdot 3^2 + \frac{5}{2} \cdot 3 = 12$ *rows in total:*

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

C.2 Support function polyhedral uncertainty

For polyhedral uncertainty, we consider the following refined convex uncertainty set after applying RPT:

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

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:

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & \mathbf{D} & 0 & \cdots & 0 \\ -\mathbf{I}_M & 0 & 0 & \cdots & 0 \\ \mathbf{H}^T & 0 & 0 & 0 & 0 \\ -\mathbf{H}^T & 0 & 0 & 0 & 0 \\ -d_1\mathbf{I}_M & 0 & d_{11}\mathbf{I}_M & \cdots & d_{1N}\mathbf{I}_M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -d_K\mathbf{I}_M & 0 & d_{K1}\mathbf{I}_M & \cdots & d_{KN}\mathbf{I}_M \\ 0 & -\mathbf{1}_N\mathbf{e}_1^T & \mathbf{H}^T & \cdots & 0 \\ 0 & \mathbf{1}_N\mathbf{e}_1^T & -\mathbf{H}^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -\mathbf{1}_N\mathbf{e}_N^T & 0 & \cdots & \mathbf{H}^T \\ 0 & \mathbf{1}_N\mathbf{e}_N^T & 0 & \cdots & -\mathbf{H}^T \end{bmatrix} \text{ and } \tilde{\mathbf{d}} = \begin{bmatrix} \mathbf{d} & \to \mathbf{u}_1 \\ \mathbf{0}_M & \to \mathbf{u}_2 \\ \mathbf{1}_N & \to \mathbf{u}_{3,1} \\ -\mathbf{1}_N & \to \mathbf{u}_{3,-1} \\ \mathbf{0}_M & \to \mathbf{u}_{4,1} \\ \vdots & \vdots \\ \mathbf{0}_M & \to \mathbf{u}_{4,K} \\ \mathbf{0}_N & \to \mathbf{u}_{5,1} \\ \mathbf{0}_N & \to \mathbf{u}_{5,-1} \\ \vdots & \vdots \\ \mathbf{0}_N & \to \mathbf{u}_{5,N} \\ \mathbf{0}_N & \to \mathbf{u}_{5,-N} \end{bmatrix} \tag{19}$$

To derive the upper bound, we consider the problem (7), employing the support function for the polyhedral set $\delta^*(\mathbf{c}(\mathbf{s})\,|\,\Theta_1^{\text{pol}}) = \min_{\mathbf{u}}\{\tilde{\mathbf{d}}^T\mathbf{u} \mid \tilde{\mathbf{D}}^T\mathbf{u} = \mathbf{c}(\mathbf{s}),\, \mathbf{u} \geq 0\}$, leading to the formulation:

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

The vector $\mathbf{u}$ is partitioned into five segments, named $\mathbf{u}_1$ through $\mathbf{u}_5$, each corresponding to distinct sets of constraints within Θ_1^{pol}. A secondary index is applied for further subdivision, as denoted alongside the definitions in (19). Utilizing the structure embedded within $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{d}}$, we obtain the optimization problem

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

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

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

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:

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

D Code exact solution

In this appendix, we provide some additional information on how to implement the procedure for obtaining the exact solution, by elaborating on the MILP to solve the master problem and providing the full pseudo-code.

D.1 Worst-case scenario given schedule

Problem (13) identifies the worst-case scenarios given the fixed schedule $\mathbf{s}'$ by maximizing the cost function subject to constraints on service and waiting times. It can be reformulated as a MILP by incorporating binary variables $\mathbf{z} \in \{0,1\}^N$, where $z_n = 1$ indicates that customer $n+1$ has no waiting time, and $z_n = 0$ that his waiting time equals $w_n + x_n - s'_n$. Here, M is a sufficiently large constant used to linearize the conditional constraints involving the binary variables z_n.

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{w},\mathbf{z}} \quad & c_W \sum_{j=1}^{N} w_i + c_I \left(w_N + \sum_{j=1}^{N-1} s'_n - \sum_{j=1}^{N-1} x_n \right) + c_O w_{N+1} \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X}, \quad \mathbf{z} \in \{0,1\}^N \\
& w_1 = 0, \quad w_n \geq 0 \quad \forall n = 2, \ldots, N+1, \\
& w_n \geq w_{n-1} + x_{n-1} - s'_{n-1} \quad \forall n = 2, \ldots, N+1, \\
& w_n \leq M(1 - z_{n-1}) \quad \forall n = 2, \ldots, N+1, \\
& w_n \leq w_{n-1} + x_{n-1} - s'_{n-1} + M z_{n-1} \quad \forall n = 2, \ldots, N+1.
\end{aligned}$$

D.2 Pseudo-code

Here, we provide a pseudo-code for obtaining the exact solution of the robust ASP.

Algorithm 1 Exact solution for robust ASP

Input: ASP instance (N, T, c_I, c_O, c_W), uncertainty set $\mathcal{X}$, convergence tolerance ϵ
Output: Robust schedule $\mathbf{s}'$, exact robust costs

— UB: Apply RPT to obtain robust upper bound
Solve Problem (9) using RPT to get upper bound
Retrieve dual variables $\mathbf{V}^*$, $\mathbf{w}^*$, $\mathbf{x}^*$

— LB: Generate scenario set and compute lower bound
Use $\mathbf{V}^*$, $\mathbf{w}^*$, $\mathbf{x}^*$ to generate scenario set $\bar{\mathcal{X}}$ in (10)
Apply adversarial approach and hill climbing on $\bar{\mathcal{X}}$
$\mathbf{s}' \leftarrow \arg\min_{\mathbf{s}\in\mathcal{S}} \max_{\mathbf{x}\in\bar{\mathcal{X}}} f(\mathbf{s},\mathbf{x})$

— Exact: Extend scenario set until convergence
While not converged **do**

  — Solve subproblem to find worst-case scenario
  $\mathbf{x}', UB \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{s}',\mathbf{x})$ (13) by solving MILP
  Update scenario set $\bar{\mathcal{X}}$ to include $\mathbf{x}'$

  — Solve master problem to refine schedule
  $\mathbf{s}', LB \leftarrow \arg\min_{\mathbf{s}\in\mathcal{S}} \max_{\mathbf{x}\in\bar{X}} f(\mathbf{s},\mathbf{x})$

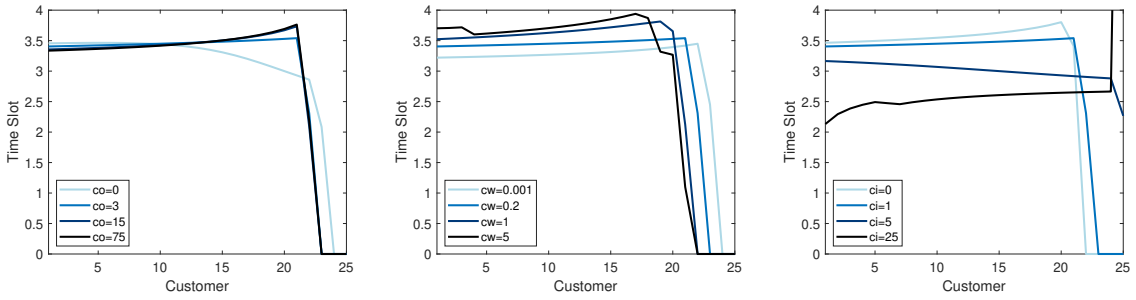
  if $UB - LB < \epsilon$ **then**
    break
end while

E Additional results

This appendix gives some additional results on ball and budget uncertainty.

E.1 Analysis ball uncertainty

Matching volume sets. 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 $\bar{\mathbf{x}} = 3$ and maximal deviation $\rho = 1$, 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,



(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

multiplied by the hypercube's volume (2^{50}), gives us an estimated volume for the budget uncertainty set, finding that $V_{\text{budget}} \approx 7.90 \times 10^{12}$.

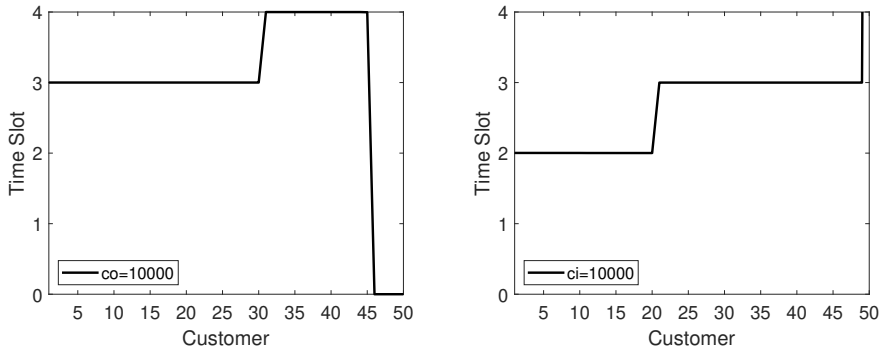
For a comparable ellipsoidal uncertainty set, we define a simple ball with $\rho = 1$ and $Q = \frac{1}{L}I$, where L is a scaling factor to be determined and I is the identity matrix. Since $\det(Q) = L^N$, the volume formula for this ball, given by $V_{\text{ball}} = \pi^{N/2}L^{N/2}/\Gamma(N/2+1)$,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

Cost parameters. 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.

E.2 Analysis cost parameters budget uncertainty



(a) Overtime cost $c_O = 10,000$ *(b) Idle time cost $c_I = 10,000$*

Figure 10: Optimal robust schedules with extreme cost parameters