The appointment scheduling problem:
a stochastic programming approach

by
Sam Coenen (SNR 2004874)

A thesis submitted in partial fulfillment of the requirements for the degree of
Bachelor in Econometrics and Operations Research

Tilburg School of Economics and Management
Tilburg University

Supervised by: J.S.H. van Leeuwaarden

January 13, 2020

Contents

1 Introduction

It is common practice for a multitude of service providers, especially in healthcare, to set up appointment systems for their activities. Clearly, there are numerous reasons to do so; considerable improvements in efficiency and cost saving can be achieved. For instance, appointments allow service providers with the possibility to allocate customers in future time slots, thereby preventing them from missing out on any job opportunity. In the context of surgery scheduling, efficient appointment scheduling leads to improved resource utilization and consequent cost saving. On the customer's side, appointments are meant to considerably reduce waiting time, which is a crucial factor that determines the quality of service. Although it is generally easy to devise basic appointment systems, obtaining optimal results often presents more than a challenge. In particular, when uncertainty is involved, ideal outcomes cannot be achieved anymore. For instance, it frequently happens that customers experience waiting time regardless of a well scheduled appointment scheme. In order to work with uncertainty, concepts from stochastic programming have been employed for constructing the mathematical model presented in the following sections. When dealing with appointment systems, it is important to make a distinction between scheduling and sequencing. Sequencing refers to the branch of optimization that determines the optimal order in which customers need to be scheduled. Scheduling focuses on establishing the optimal length of the appointments. In this paper we propose a solution to the so-called 'appointment scheduling problem' with particular emphasis on the method known as 'sample average approximation'. In this regard, we will analyse how the effectiveness of the sample average approximation combined with a decomposition method compares to other options. Furthermore, depending on the holding assumptions, we will provide solutions to various settings of the same problem. Section 2 will introduce the parameters involved, the notation used and the outline of the mathematical model for the basic and the extended settings. Section 3 will discuss different approaches to solve the model for discrete distributions and continuous distributions. Finally, in section 4 some numerical results and corresponding analysis will be presented.

2 Problem description

In order to construct a clear appointment scheduling problem, we will analyse a single server or healthcare provider. The time horizon is limited to a single working day with finite length. Hence, steady-state analysis will not be required. The costs involved are relative to the server's idle time, customers' waiting time and overtime work and they are assumed to remain constant for different jobs. Here, idle time, waiting time and overtime work depend on the choice for appointment times as well as the customers' arrival times and service durations. This relation will be explained in more detail later in this section. The ultimate goal will be to devise the optimal appointment schedule that minimizes the expected total cost. Consequently, linear programming theory will be required. More specifically, the presence of uncertainty in customers' arrival times, service durations and number of jobs, leads to the formulation of a stochastic linear program. Because of different assumptions on the random components, we distinguish between a basic model and an extended model. First we introduce the notation used throughout the thesis along with some important remarks.

W_k: waiting time corresponding to customer k;
$\mathbf{W} = (W_1, W_2, ..., W_n)$: vector or waiting times;
$\mathbf{x} = (x_1, x_2, ..., x_{n-1})$: vector of time slots for the last $n-1$ jobs;
$T_k = (T_1, T_2, ..., T_n)$: idle time between customer k-1 and k;
$\mathbf{T}$: vector of idle times;
$\mathbf{S}$: vector of random service times;
c^W: cost relative to waiting time;
c^T: cost relative to idle time;
n: total number of jobs;
d: day length;
l: lateness (overtime);
g: earliness;
c^l: overtime cost;
c^g: earliness cost;
(realizations of random variables are denoted by lowercase letters);

Waiting time, idle time, lateness and earliness corresponding to customer k present the following recursion:

$$
\begin{aligned}
W_k &= (W_{k-1} + S_{k-1} - x_{k-1})^+ , \qquad i = 2, ..., n \\
T_k &= (-W_{k-1} - S_{k-1} + x_{k-1})^+ , \qquad i = 2, ..., n \\
l &= \left(W_k + S_k + \sum_{k=1}^{n-1} x_k - d \right)^+ , \\
g &= \left(-W_k - S_k - \sum_{k=1}^{n-1} x_k + d \right)^+ ,
\end{aligned}
$$

with $W_1 = 0$ and $T_1 = 0$.

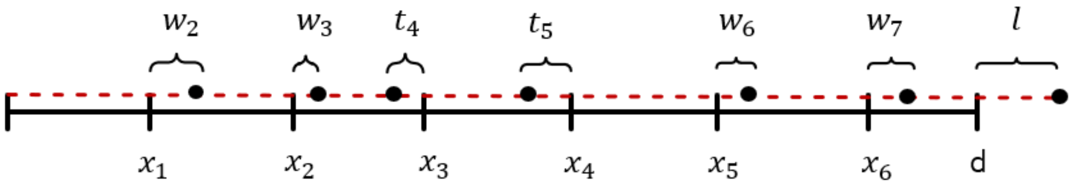


Figure 1: representation of the recursive relation. The dashed line represents service times. Each service time ends with a dot.

2.1 Basic model

The first model we introduce presents uncertainty in service times only. In fact, we assume that the number of jobs is known beforehand and customers are punctual. This basic appointment scheduling problem is described by the following minimization problem:

$$\min_{\mathbf{x}} \sum_{k=2}^{n} c^W E\left[W_k\right] + \sum_{k=2}^{n} c^T E\left[T_k\right] + c^l E\left[l\right] + c^g E\left[g\right] \tag{1}$$

This can be written as the following stochastic linear program which can be solved as a 2-stage linear program (Denton and Gupta,2003):

$$\min_{\mathbf{x}} E\left[\sum_{k=2}^{n} c^W W_k + \sum_{k=2}^{n} c^T T_k + c^l l + c^g g\right] \tag{2}$$

$$\begin{array}{rrcl}
s.t. & w_2 \qquad - t_2 & = & s_1 - x_1 \\
 & -w_2 + w_3 \qquad - t_3 & = & s_2 - x_2 \\
 & \ddots \quad \ddots \qquad \ddots & = & \vdots \\
 & -w_{n-1} + w_n \quad - t_n & = & s_{n-1} - x_{n-1} \\
 & -w_n + l - g & = & s_n - d + \displaystyle\sum_{k=1}^{n-1} x_k \\
 & l, g \geqslant 0, \;\; x_i \geqslant 0, \;\; w_i \geqslant 0, \;\; t_i \geqslant 0 \;\; \forall i = 1, ..., n. & &
\end{array} \tag{3}$$

The first stage is given by

$$\min_{\mathbf{x}} E\left[Q\left(x, \mathbf{S}\right)\right] \tag{4}$$

where

$$Q\left(x, \mathbf{S}\right) = \min_{\mathbf{y}} \left[\mathbf{cy} | \mathbb{T}\mathbf{x} + \mathbb{W}\mathbf{y} = \mathbf{h}, \mathbf{y} \geqslant 0\right] \tag{5}$$

represents the second stage with

$$\mathbb{W} = \begin{bmatrix}
1 & & & & -1 & & & & \\
-1 & 1 & & & & -1 & & & \\
 & \ddots & \ddots & & & & \ddots & & \\
 & & -1 & 1 & & & & & \\
 & & & -1 & & & & 1 & -1
\end{bmatrix}$$

,

$$\mathbb{T} = \begin{bmatrix}
1 & & \\
 & \ddots & \\
 & & 1 \\
-1 & \cdots & -1
\end{bmatrix}, \; \mathbf{c} = \begin{bmatrix} c^W \\ c^T \\ c^l \\ c^g \end{bmatrix}, \; \mathbf{y} = \begin{bmatrix} \mathbf{w} \\ \mathbf{t} \\ l \\ g \end{bmatrix} \text{ and } \mathbf{h} = \begin{bmatrix} S_1 \\ \vdots \\ S_n - d \end{bmatrix}.$$

2-stage linear programs refer to stochastic programming models where the decision maker has to take some first-stage action before random events occur. Once these are observed, a second-stage decision can be made in order to offset the effects of the first-stage decision (Shapiro and Philpott, 2007). In our case, $\mathbf{x}$ is the vector of first stage decision variables and $\mathbf{y}$ is the vector of second stage decision variables. As a first step, before service times become available, an initial guess for $\mathbf{x}$ has to be made. For instance, we could select the $\mathbf{x}$ that minimizes the deterministic part of our objective function, i.e. $\mathbf{x} = [0, ..., 0]$. Once the service times are observed, the second-stage decision variables corresponding to $\mathbf{x} = [0, ..., 0]$ can be obtained, namely $w_2 = s_1, w_3 = s_2$, etc. (plug $\mathbf{x}$ in the recursion)

2.2 Extended model

In appointment scheduling, a relevant issue stems from the high frequency of no-shows. Customers missing appointments can severely impact the efficiency of an appointment system if they are not accounted for. For this reason, the next model allows for no-shows. The effect for customer k is replicated by the random variable NS_k.

$$NS_k = \begin{cases} 0 \text{ with probability } p_k \\ 1 \text{ with probability } 1 - p_k \end{cases}$$

In addition, we relax the assumption of punctuality by including the random variable DE_k that represents the delay of customer k (this type of delay does not produce idle time before the arrival of a customer but it extends the overall service time). Finally, we allow for randomness in the number of jobs by means of dynamic scheduling (Denton and Erdogan, 2011). This choice is motivated by the numerous health-care applications in which urgent patients, or patients on short notice, are accepted. In this context, only the lower and upper bound for the total number of jobs are known in advance. The lower bound consists of the number of customers scheduled upfront. The upper bound is the largest number of customers that can be served in one day. Dynamic scheduling imposes the development of the problem on multiple stages where each stage corresponds to a different number of jobs to be scheduled. The starting stage represents the scenario in which only the customers scheduled upfront are considered. Adding one urgent customer shifts the problem to the next stage. This process continues until we reach the upper bound and therefore the last stage. The following figure depicts this scenario.

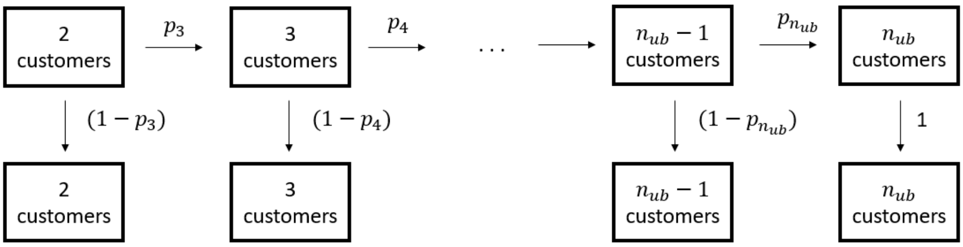


Figure 2: representation of dynamic scheduling. p_k is the probability to shift from stage $k - 1$ to stage k. n_{ub} is the upper bound for the total number of customers.

For the extended model we maintain the previous notation. However, the decision variables other than x now present a superscript indicating the stage they belong to.

For clarity matters, stages are numbered according to the number of customers in the system (e.g. 2 customers in stage 2, 3 customers in stage 3 and so on). Stage 1 is ignored as it is trivial. With this in mind, the new minimization problem can be written as follows:

$$\min_{\mathbf{x}} \left(1-p_3\right) E\left[Q^2\left(\mathbf{x},\mathbf{S}\right)\right] + p_3\left(1-p_4\right) E\left[Q^3\left(\mathbf{x},\mathbf{S}\right)\right] + \cdots + \left(\prod_{i=3}^{n_{ub}} p_i\right) E\left[Q^{n_{ub}}\left(\mathbf{x},\mathbf{S}\right)\right] \tag{6}$$

where

$$Q^i\left(\mathbf{x},\mathbf{S}\right) = \min_{\mathbf{w},\mathbf{t},l,g} \sum_{k=2}^{i} c^W W_k^i + \sum_{k=2}^{i} c^T T_k^i + c^l l^i + c^g g^i \tag{7}$$

$$\begin{array}{lrcl} s.t. & w_2^i \qquad - t_2^i & = & s_1^* - x_1 \\ & -w_2^i + w_3^i \qquad - t_3{}^i & = & s_2^* - x_2 \\ & \ddots \quad \ddots \qquad \ddots & = & \vdots \\ & -w_{n_{ub-1}}^i + w_{n_{ub}}^i \quad - t_{n_{ub}}^i & = & s_{n_{ub}-1}^* - x_{n_{ub}-1} \\ & -w_n^i + l^i - g^i & = & s_{n_{ub}}^* - d + \displaystyle\sum_{k=1}^{n_{ub}-1} x_k \end{array} \tag{8}$$

$Q^i\left(\mathbf{x},\mathbf{S}\right)$ is the total cost corresponding to stage i. n_{ub} is the upper bound for the number of customers. s_k^* is the realization of the random variable $S_k^* = NS_k\left(S_k + DE_k\right)$ for customer k. p_i is the probability of moving from stage $i-1$ to stage i. Note that (6) has a set of constraints of the form of (8) for each stage i.

3 Problem solution

In general, depending on the type of random variables involved, we adopt two different strategies to solve (2) and (6). If the random variables are discrete, (2) and (6) can be directly solved by minimizing the objective function over all possible scenarios. However, when the random variables are continuous, this approach is not viable as there exist infinitely many scenarios. Nevertheless, there are several methods that provide an approximation to the solution such as the 'sample average approximation method' and the 'sequential bounding method'. The latter has been applied to the basic model and thoroughly analysed by Denton and Gupta (2003). In their study, Denton and Gupta obtained an approximated solution, as well as converging bounds for it. Instead, this paper focuses on the sample average approximation method by providing results and comparing its effectiveness to the sequential bounding approach.

3.1 Discrete distribution

Denote the support of S by $\mathbf{\Xi} = \{s \mid \mathbb{P}\left(S=s\right) \neq 0\}$, i.e. the set of values S can take. When $\mathbf{\Xi} \in \mathbb{R}^n$ is finite with J possible scenarios ω_j (S is a discrete random variable), (2) can be written as:

$$\min_{\mathbf{x}} \sum_{j=1}^{J} p_j \mathbf{c}^\top \mathbf{y}\left(\omega_j\right) \tag{9}$$

$$
\begin{array}{llll}
s.t. & \mathbb{T}\mathbf{x} + \mathbb{W}\mathbf{y}\left(\omega_1\right) & & = \mathbf{h}\left(\omega_1\right) \\
& \vdots + & \ddots & = \vdots \\
& \mathbb{T}\mathbf{x} + & \mathbb{W}\mathbf{y}\left(\omega_J\right) & = \mathbf{h}\left(\omega_J\right)
\end{array}
\tag{10}
$$

Each line in (10) corresponds to a set of constraints of the form of (3). Clearly this problem can be solved as a standard LP-problem. Although it could easily be solved by 'brute-force', the number of constraints increases significantly with the number of scenarios, making the problem prohibitive; for n jobs and J scenarios we have nJ constraints. The computational efforts to solve large LP-problems with the same structure as (7) can be greatly reduced by employing the L-shaped method. It is a decomposition method that can be applied to particular problems with the same structure of (10). The key idea is that, instead of solving a single LP-problem with an enormous number of constraints, it prescribes the solution of multiple smaller problems.

3.2 L-shaped method

The L-shaped method consists of an iterative process that requires the solution of a master problem, an LP-problem representative of the full problem, and the solution of all subproblems corresponding to each scenario. At each iteration, some optimality cuts (additional constraints) for the first stage decision variables are derived from the dual solution of the second stage for some given values of the first stage decision variables. The optimality cuts are added to the master problem and its new solution is used to generate more accurate cuts. It can be shown that the procedure terminates with an optimal solution in a finite number of steps. To make things clear, we explain every step of the process applied to our basic model. First, we replace the second-stage cost in (9) with θ and ignore the constraints that involve second-stage decision variables. The resulting LP-problem is the initial master problem

$$
\begin{array}{ll}
\min\limits_{\mathbf{x},\theta} \theta & \\
s.t. & \mathbf{x}, \theta \geqslant 0
\end{array}
\tag{11}
$$

Solving the master problem yields a first solution $\mathbf{x}^0$. Next, for each scenario, we find the optimal solutions to the dual problem of the second-stage (5) resulting from $\mathbf{x}^0$, i.e. we solve

$$
\max \left(\mathbf{h}\left(\omega_j\right) - \mathbb{T}\mathbf{x}^0\right) \boldsymbol{\pi}\left(\omega_j\right) \tag{12}
$$

$$
s.t. \qquad \mathbb{W}^T \boldsymbol{\pi}\left(\omega_j\right) \leqslant \mathbf{c}
$$

This can be done efficiently by means of the following backwards recursion (Denton and Gupta, 2003)

$$
\pi_i^*\left(\mathbf{x}, \omega_j\right) = \begin{cases} -c^t & \text{if } w_i = 0 \\ c^w + \pi_{i+1}^*\left(\mathbf{x}, \omega_j\right) & \text{if } w_i > 0 \end{cases}
$$

$$
\pi_n^*\left(\mathbf{x}, \omega_j\right) = \begin{cases} 0 & \text{if } l = 0 \\ c^l + \pi_n^*\left(\mathbf{x}, \omega_j\right) & \text{if } l > 0 \end{cases}
$$

Solving the dual problems with common LP-solvers would result in a significant loss in efficiency (total run time for 2000 scenarios decreases from about 345 seconds to about 20 seconds by implementing the backwards recursion). Once the dual multipliers $\boldsymbol{\pi}^*(\mathbf{x}, \omega_j)$ are obtained, we can construct the optimality cuts to add to the master problem. From weak duality theory, we know that the objective value of (12) calculated for $\boldsymbol{\pi}^*(\mathbf{x}, \omega_j)$ has to be smaller or equal to the objective value of its primal problem. In other words we have that

$$\theta(\omega_j) \geqslant \boldsymbol{\pi}^*(\omega_j)^\top \mathbf{h}(\omega_j) - \boldsymbol{\pi}^{*\top}(\omega_j)\,\mathbb{T}\mathbf{x}^0 \qquad \text{for all scenarios } \omega_j.$$

From this relation for all scenarios we obtain a single optimality cut (Skyle and Wets, 1969)

$$\theta + E\left[\boldsymbol{\pi}^{*\top}\mathbb{T}\mathbf{x}\right] \geqslant E\left[\boldsymbol{\pi}^{*\top}\mathbf{h}(\omega_j)\right]$$

and we add it to (11). The new master problem yields the solution $(\mathbf{x}^1, \theta^1)$ which, in turn, is used to generate new dual multipliers. If $\theta^1 \geqslant E\left[\left(\mathbf{h} - \mathbb{T}\mathbf{x}^1\right)\boldsymbol{\pi}\left(\mathbf{x}^1\right)\right]$ we terminate with an optimal solution $(\mathbf{x}^1, \theta^1)$, otherwise we add the new optimality cut to the master problem and repeat the process.

In order to avoid any infeasibility of the second stage we pose the simplifying assumption of relative complete recourse; for every feasible first-stage solution and every scenario ω_j there exists a feasible recourse decision $\mathbf{y}$ (i.e. a feasible solution to (5)). The algorithm explained above is the 'single cut L-shaped method', one of the possible variants of the L-shaped method. Slightly different in its construction is the variation known as 'multi-cut L-shaped method'. Instead of a single θ, the Master problem now has a decision variable θ_j for each scenario and it is written as follows:

$$\begin{aligned} &\min_{\mathbf{x},\boldsymbol{\theta}} \sum_{j=1}^{J} p_j \theta_j(\omega_j) \\ &s.t. \qquad \mathbf{x}, \theta_j \geqslant 0 \quad \forall j = 1, ..., J. \end{aligned} \tag{13}$$

At each iteration, a set of J constraints

$$\theta(\omega_j) + \boldsymbol{\pi}^{*\top}(\omega_j)\,\mathbb{T}\mathbf{x} \geqslant \boldsymbol{\pi}^*(\omega_j)^\top \mathbf{h}(\omega_j) \qquad \text{for all scenarios } \omega_j.$$

is added to the master problem instead of the single cut for the previous case. Like the single-cut variation, the multi-cut L-shaped method can be shown to generate an optimal solution in a finite number of steps. The difference with the single-cut L-shaped method is that the multi-cut variation converges faster. On the other hand, because the number of constraints becomes large relatively fast, the single-cut variation might be preferable when the number of iterations is large. Moreover, in order to drastically reduce run time, it is possible to include a tolerance margin so that the procedure ends when θ is close enough to $E\left[(\mathbf{h} - \mathbb{T}\mathbf{x})\,\boldsymbol{\pi}(\mathbf{x})\right]$.

3.3 L-shaped method for the extended model

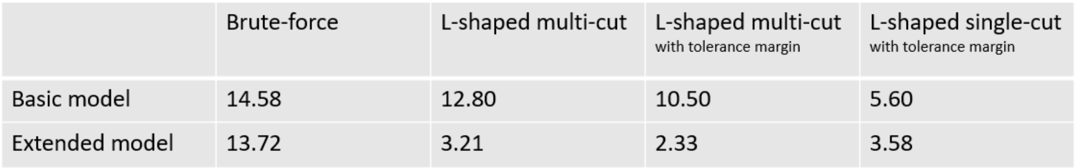
Up to this point, we have shown how to apply the L-shaped method to the basic model. However, applying it to the extended model is not much different. Similarly to the basic model, when we have a finite number of scenarios we can rewrite (6) as the following LP-problem:

$$\min_{\mathbf{x}} (1-p_3)\sum_{j=1}^{J} p_j \mathbf{c}^\top \mathbf{y}^2(\omega_j) + p_3(1-p_4)\sum_{j=1}^{J} p_j \mathbf{c}^\top \mathbf{y}^3(\omega_j) + \cdots + \left(\prod_{i=3}^{n_{ub}} p_i\right)\sum_{j=1}^{J} p_j \mathbf{c}^\top \mathbf{y}^{n_{ub}}(\omega_j) \tag{14}$$

$$\begin{array}{llll} s.t. & \mathbb{T}^{\mathbf{i}}\mathbf{x} + \mathbb{W}^{\mathbf{i}}\mathbf{y}^{\mathbf{i}}(\omega_1) & = \mathbf{h}(\omega_1) & \\ & \vdots + \qquad \ddots & = \vdots & \forall i = 2, ..., n_{ub}. \\ & \mathbb{T}^{\mathbf{i}}\mathbf{x} + \qquad \mathbb{W}^{\mathbf{i}}\mathbf{y}^{\mathbf{i}}(\omega_J) & = \mathbf{h}(\omega_J) & \end{array} \tag{15}$$

The L-shaped method can now be implemented in the same fashion as before. The resulting master problem for the single-cut version presents only one decision variable and has the same form of (11). When it comes to solving the dual problems of the subproblems, we take a set of constraints from each stage at a time. In other words, instead of taking a single set of constraints for each scenario, we take $n_{ub} - 1$ sets of constraints for each scenario. The solution to these dual problems can be readily obtained by using the backwards recursion (section 3.2) and then multiplying the results by the probability to be in stage i. Therefore, $\boldsymbol{\pi}^*(\mathbf{x}, \omega_j)$ will no longer be a $n \times 1$ vector but it will be a $(2 + 3 + ... + n_{ub}) \times 1$ vector.

Beside the remarkable gain in efficiency, the L-shaped method plays a major role in memory consumption. In fact, with a large number of scenarios, the number of constraints involved in the brute-force approach can easily become too large to handle. The next table shows the benefits that come with the L-shaped method by comparing the run time to solve (9) using the 'brute-force' approach and the run time with the two variations of the L-shaped method. Note that the number of scenarios generated for the basic model is 1850 while it is 300 for the extended model. These are the largest numbers of scenarios that could be handled by the brute-force approach. Moreover, we present results for the L-shaped method with a 0.00001 tolerance margin. The single-cut version without margin is not included as it takes an excessive amount of time to converge.



	Brute-force	L-shaped multi-cut	L-shaped multi-cut with tolerance margin	L-shaped single-cut with tolerance margin
Basic model	14.58	12.80	10.50	5.60
Extended model	13.72	3.21	2.33	3.58

Figure 3: run time for the basic and extended models with different techniques. For the basic model we generated 1850 scenarios and solved it for 7 jobs. For the extended model we generated 300 scenarios and solved it for 3 customers scheduled upfront and 5 additional customers.

3.4 Continuous distribution

When the support of the service time $\mathbf{\Xi}$ is infinite (i.e. it follows a continuous distribution), the dual problem of the second stage involves the minimization of

$$Q\left(\mathbf{x}\right)=\int_{\mathbf{\Xi}}\pi\left(\mathbf{x},\mathbf{s}\right)\left(\mathbf{h}-\mathbb{T}\mathbf{x}\right)P\left(d\mathbf{z}\right) \tag{16}$$

Since this integral is generally impossible to solve, our goal is to obtain a finite number of realizations for $\mathbf{S}$ in order to write an LP-problem of the same form of (9). In this section we will provide two different techniques to approximate the optimal result, namely the 'sample average approximation method' and the 'sequential bounding method'.

3.4.1 Sequential bounding

The first technique we examine is known as the 'sequential bounding method' (Denton and Gupta, 2003). In principle, it consists of partitioning the n-dimensional support (n-dimensional as we have n jobs) into v hyper-rectangles $V_k, (k = 1, ..., v)$ of the form

$$V_k = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_n, b_n] \tag{17}$$

with p_k the probability of being in the hyper-rectangle k and $\mathbf{s}^k$ the vector of corresponding expectations. Upper and lower bounds to the optimal solution can be computed for each partition. For example, a lower bound is provided by the Jensen inequality as follows:

$$E\left[f\left(\mathbf{x},\mathbf{S}\right)\right]\geqslant\sum_{k=1}^{v}p^k f\left(\mathbf{x},\mathbf{s}_k\right) \tag{18}$$

while an upper bound can be obtained from, for instance, the Edmundson-Madansky upper bound for convex functions by calculating the objective value at the extremes of each hyper-rectangle. When the partition is refined such that $v \to \infty$, the approximate probability distribution converges to the true distribution and the lower and upper bound converge to the optimal solution. In order to avoid the large computational time involved in the calculation of the Edmundson-Madansky upper bound, Denton and Gupta (2003) devised an alternative upper bound which is independent of the distribution of the random components.

3.4.2 Sample average approximation

A second and more tractable approach involves the sample average approximation method (SAA). It is important to note that the sample average approximation method is considered to be a general approach to stochastic optimization rather than a fully-fledged algorithm. The key idea is that we can approximate the solution of the expected value function by a sample average. In particular, with J scenarios at our disposal, we can approximate $E\left[Q\left(x,\mathbf{S}\right)\right]$ with $\frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right)$. Hence, the solution obtained from (9)-(10) is an approximation of the solution for (2).

We will now explain in more detail how to apply SAA to our specific problem. First, we need to generate a set of scenarios. Note that this is analogous to the first step of the sequential bounding method that requires support partitioning. Through Monte Carlo simulation, a number J of i.i.d. realizations of $\mathbf{S}$ is drawn from some assumed service time distribution. Then, the resulting problem is written as (9) with $p_j = \frac{1}{J}\left(j = 1, ..., J\right)$. Clearly, as it was mentioned before, this problem can be solved by brute-force as well as with the help of a decomposition method such as the L-shaped method. Finally, the result provides an approximation of (2). This can be made arbitrarily accurate by generating a larger number of scenarios.

An important feature of SAA is the fact it produces unbiased and consistent estimators of the expected value function; because it holds that

$$E\left[\frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right)\right] = E\left[Q\left(x,\mathbf{S}\right)\right]$$

the estimator is unbiased. Moreover, according to the weak law of large numbers,

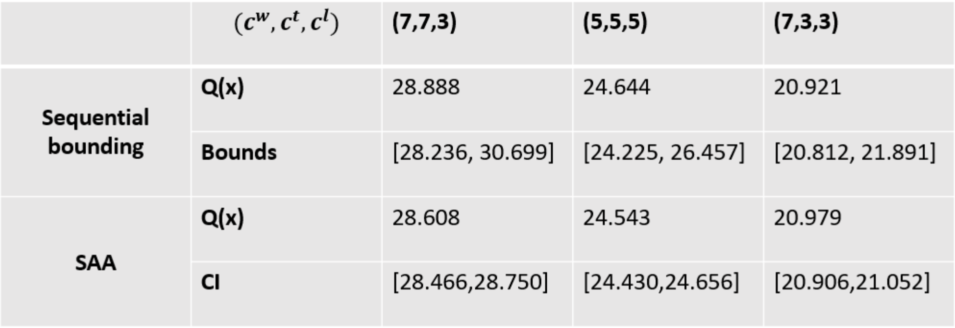
$$\frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right) \xrightarrow{p} E\left[Q\left(x,\mathbf{S}\right)\right]$$

so the estimator is consistent too. Additionally, in order to determine the accuracy of our measurements, it is possible to construct $100\left(1-\alpha\right)\%$ approximate confidence intervals as

$$\left[\frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right) - \Phi^{-1}\left(1-\frac{\alpha}{2}\right)\frac{\widehat{\sigma}}{\sqrt{J}}, \frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right) + \Phi^{-1}\left(1-\frac{\alpha}{2}\right)\frac{\widehat{\sigma}}{\sqrt{J}}\right]$$

where $\widehat{\sigma}$ is the sample standard deviation of $\frac{1}{J}\sum_{j=1}^{J}\mathbf{c}^T\mathbf{y}\left(\omega_j\right)$.

The next table compares results obtained by Denton and Gupta (2003) with sequential bounding and the results we obtained with SAA.



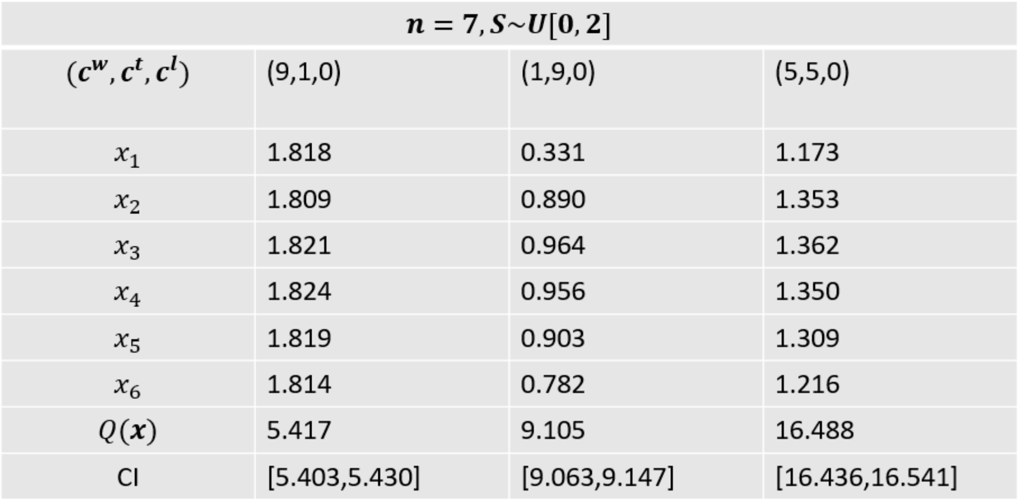
	(c^w, c^t, c^l)	**(7,7,3)**	**(5,5,5)**	**(7,3,3)**
Sequential bounding	**Q(x)**	28.888	24.644	20.921
	Bounds	[28.236, 30.699]	[24.225, 26.457]	[20.812, 21.891]
SAA	**Q(x)**	28.608	24.543	20.979
	CI	[28.466,28.750]	[24.430,24.656]	[20.906,21.052]

Figure 4: results obtained for $n = 7$, $S \sim U\,[0, 2]$, $d = 7$. $Q\,(\mathbf{x})$ is the estimated total cost. CI is the 95% confidence interval

Denton and Gupta obtained their results after 50 iterations of the sequential bounding method, adding 500 cells at each iteration. Thus, we decided to generate 25000 scenarios for the SAA method. Moreover, note that the bounds they obtained are guaranteed to contain the optimal solution. On the other hand, with SAA we obtained 95% confidence intervals, i.e. we are 95% sure the sample we generated produces an interval that contains the optimal solution, but we have no guarantee.

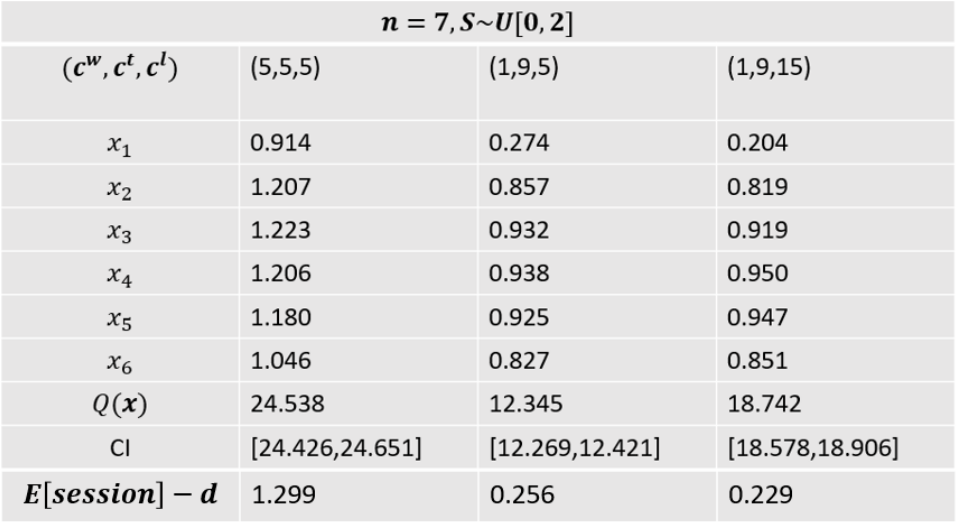
4 Results and analysis

In this section numerical results and insights to different settings of the ASP will be provided. Figure 5 through 9 present results relative to the basic model. The remaining tables are relative to the extended model. In all instances, the day length is assumed to be $d = 7$ and earliness cost $c^g = 0$. All results are obtained by generating 25000 scenarios. The values presented include optimal appointment times, minimum total costs and 95% confidence intervals for the total cost.



$\boldsymbol{n = 7, S \sim U[0,2]}$			
$(\boldsymbol{c^w, c^t, c^l})$	(9,1,0)	(1,9,0)	(5,5,0)
x_1	1.818	0.331	1.173
x_2	1.809	0.890	1.353
x_3	1.821	0.964	1.362
x_4	1.824	0.956	1.350
x_5	1.819	0.903	1.309
x_6	1.814	0.782	1.216
$Q(\boldsymbol{x})$	5.417	9.105	16.488
CI	[5.403,5.430]	[9.063,9.147]	[16.436,16.541]

Figure 5: shows the effect of different sets of costs.



$n = 7, S \sim U[0,2]$			
(c^w, c^t, c^l)	(5,5,5)	(1,9,5)	(1,9,15)
x_1	0.914	0.274	0.204
x_2	1.207	0.857	0.819
x_3	1.223	0.932	0.919
x_4	1.206	0.938	0.950
x_5	1.180	0.925	0.947
x_6	1.046	0.827	0.851
$Q(\boldsymbol{x})$	24.538	12.345	18.742
CI	[24.426,24.651]	[12.269,12.421]	[18.578,18.906]
$\boldsymbol{E[session] - d}$	1.299	0.256	0.229

Figure 6: shows the effect of adding an overtime cost. It also includes the difference between the expected session length and d.

A striking result that emerges from Figure 5 and Figure 6 is the "dome shape" distribution attained by $\mathbf{x}$ (see figure 7). The effect becomes more pronounced for higher idle costs to waiting costs ratios while it mitigates for lower idle costs to waiting costs ratios. Indeed, for $\left(c^w, c^t, c^l\right) = (9, 1, 0)$ the distribution of $\mathbf{x}$ attains an almost flat line. When waiting costs are high, expected waiting times are minimized by taking appointment times close to the upper bound of $\boldsymbol{\Xi}$ (the support of the service times). In fact, for $c^w > 0$, $c^t = 0$ and $c^l = 0$, $x_k = 2$, $\forall k = 1, ...6$.. On the other hand, when idle costs are high, the expected waiting time for customer 1 is considerably large. The effect is carried over according to the recursion of waiting times and, in turn, weighs negatively on idle times. This is taken to the extreme for $c^w = 0$, $c^t > 0$ and $c^l = 0$, which yield $x_k = 0$, $\forall k = 1, ...6$ (0 being the lower bound of $\boldsymbol{\Xi}$). Furthermore, Figure 6 includes the difference between the expected working day length and d. Clearly, when d equals the sum of the mean service times ($d = 7$ in Figure 6), the optimal solution requires some buffer even for high overtime costs. This is a significant result since it is common practice to schedule jobs based on the sum of the mean service times.

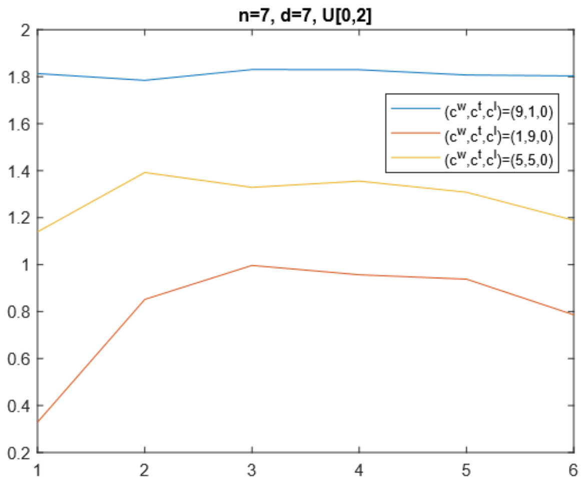
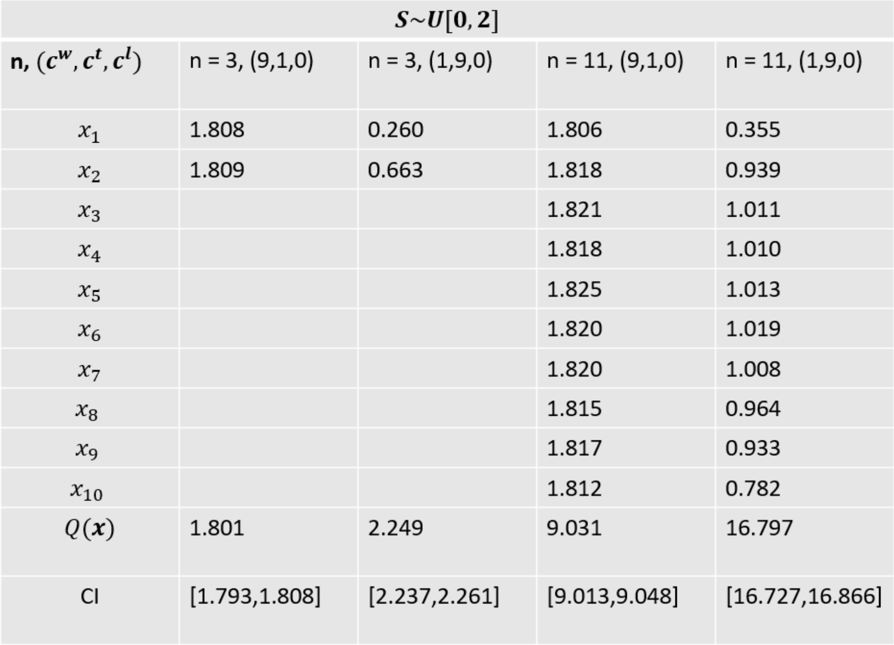


Figure 7: plot of x for 3 different sets of costs. Note the 'dome shape' attained by x for a high idle cost to waiting cost ratio.

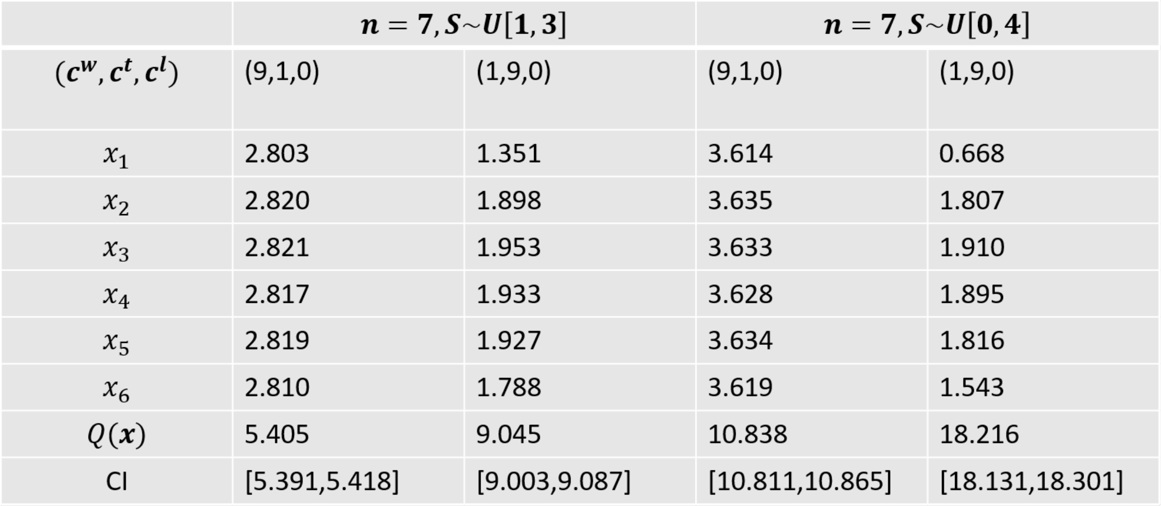
Figure 8 shows that increasing the number of jobs has a different consequence depending on the ratio between idle costs and waiting costs; when the ratio is high, increasing n leads to longer appointment times. Vice versa, when the ratio is low, there appear to be no significant changes in appointment times. Figure 9 presents results obtained from service times drawn from $U\left[1,3\right]$ and $U\left[0,4\right]$ distributions. In other words, the distribution we analysed so far has been shifted by a factor 1 in the first case and scaled by a factor 2 in the second case. These results provide empirical evidence to the following proposition.

Proposition 1: the effect of the transformation $\mathbf{S}\mapsto a\mathbf{S}+b$, where $a\in$ and $b\in^{n}$, on the optimal solution, is $Q\left(\mathbf{x}\right)\mapsto aQ\left(\mathbf{x}\right)$ and $\mathbf{x}\mapsto a\mathbf{x}+b$ (Denton and Gupta, 2003).



$S \sim U[0,2]$				
n, (c^w, c^t, c^l)	n = 3, (9,1,0)	n = 3, (1,9,0)	n = 11, (9,1,0)	n = 11, (1,9,0)
x_1	1.808	0.260	1.806	0.355
x_2	1.809	0.663	1.818	0.939
x_3			1.821	1.011
x_4			1.818	1.010
x_5			1.825	1.013
x_6			1.820	1.019
x_7			1.820	1.008
x_8			1.815	0.964
x_9			1.817	0.933
x_{10}			1.812	0.782
$Q(\boldsymbol{x})$	1.801	2.249	9.031	16.797
CI	[1.793,1.808]	[2.237,2.261]	[9.013,9.048]	[16.727,16.866]

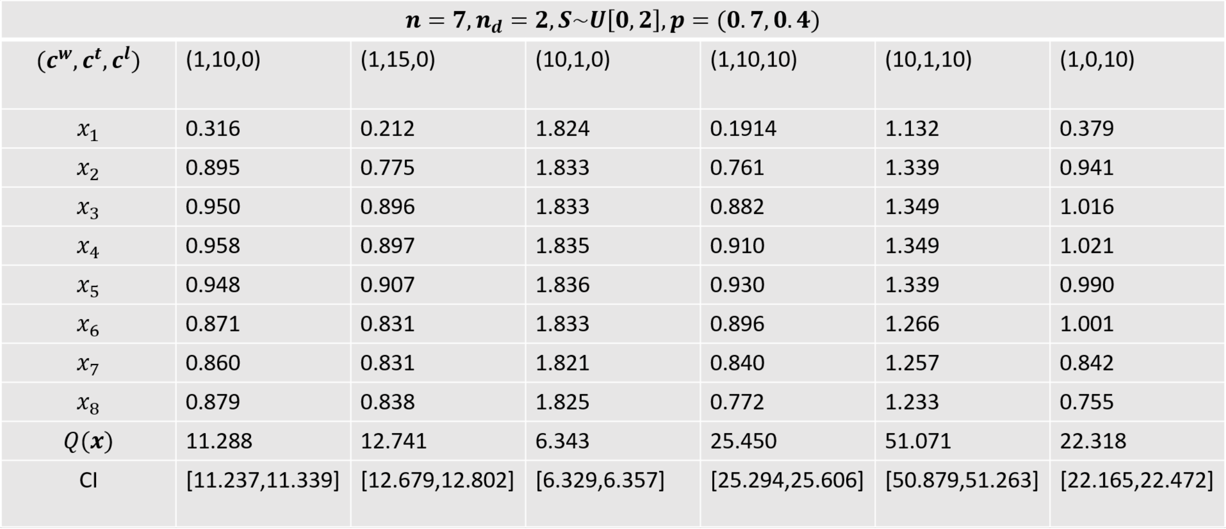
Figure 8: shows the effect of increasing n.



	$\boldsymbol{n = 7, S \sim U[1,3]}$		$\boldsymbol{n = 7, S \sim U[0,4]}$	
$(\boldsymbol{c^w, c^t, c^l})$	(9,1,0)	(1,9,0)	(9,1,0)	(1,9,0)
x_1	2.803	1.351	3.614	0.668
x_2	2.820	1.898	3.635	1.807
x_3	2.821	1.953	3.633	1.910
x_4	2.817	1.933	3.628	1.895
x_5	2.819	1.927	3.634	1.816
x_6	2.810	1.788	3.619	1.543
$Q(\boldsymbol{x})$	5.405	9.045	10.838	18.216
CI	[5.391,5.418]	[9.003,9.087]	[10.811,10.865]	[18.131,18.301]

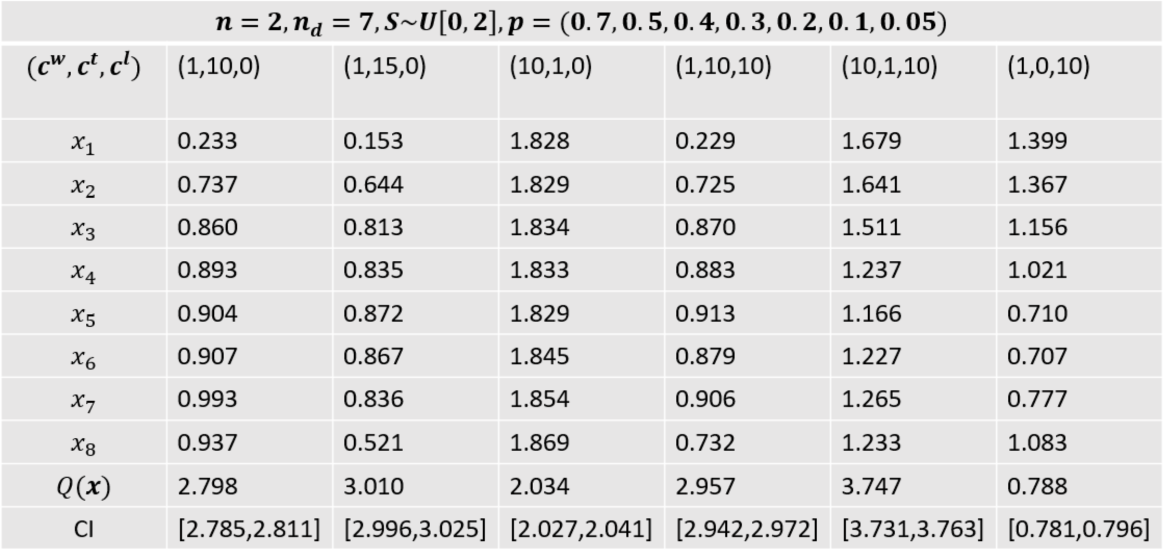
Figure 9: shows the effect of shifting and scaling the service time.

We now proceed to analyse the results obtained from the extended model. Once again, a crucial role is played by the ration between idle cost and waiting time cost. From Figure 10 we can see that when the number of additional customers n_d is relatively low, the effects of different idle cost to waiting time cost ratios are preserved; similarly to the basic model, a low ratio causes the appointment times to attain a flat distribution. On the contrary, a high ratio yields the ‘dome shape’ distribution and, as we have already seen, the effect becomes more pronounced by increasing the ratio. Furthermore, the addition of an overtime cost reduces the appointment times overall. More interesting results can be found by increasing the number of additional customers with respect to the customers scheduled upfront and by taking decreasing probabilities for their arrival. This is done in Figure 11. When the overtime cost is 0, the ‘dome shape’ and the flat distribution once again appear for the same ratios as before. This being said, we turn our attention to the instances that present an overtime cost. When the waiting time cost is large compared to the idle time cost, the initial appointment times are relatively long. They then decrease before increasing again in the end (see figure 14). This because irrelevant idle time costs incentivize the minimization of waiting time and overtime. As a consequence, long appointment times are scheduled at the beginning when there is a high probability of arrival in order to prevent waiting time from accumulating. Long appointment times are also scheduled at the end in order to minimize the probability of overtime work in the rare event all customers arrive.



$n = 7, n_d = 2, S \sim U[0,2], p = (0.7, 0.4)$						
(c^w, c^t, c^l)	(1,10,0)	(1,15,0)	(10,1,0)	(1,10,10)	(10,1,10)	(1,0,10)
x_1	0.316	0.212	1.824	0.1914	1.132	0.379
x_2	0.895	0.775	1.833	0.761	1.339	0.941
x_3	0.950	0.896	1.833	0.882	1.349	1.016
x_4	0.958	0.897	1.835	0.910	1.349	1.021
x_5	0.948	0.907	1.836	0.930	1.339	0.990
x_6	0.871	0.831	1.833	0.896	1.266	1.001
x_7	0.860	0.831	1.821	0.840	1.257	0.842
x_8	0.879	0.838	1.825	0.772	1.233	0.755
$Q(\boldsymbol{x})$	11.288	12.741	6.343	25.450	51.071	22.318
CI	[11.237,11.339]	[12.679,12.802]	[6.329,6.357]	[25.294,25.606]	[50.879,51.263]	[22.165,22.472]

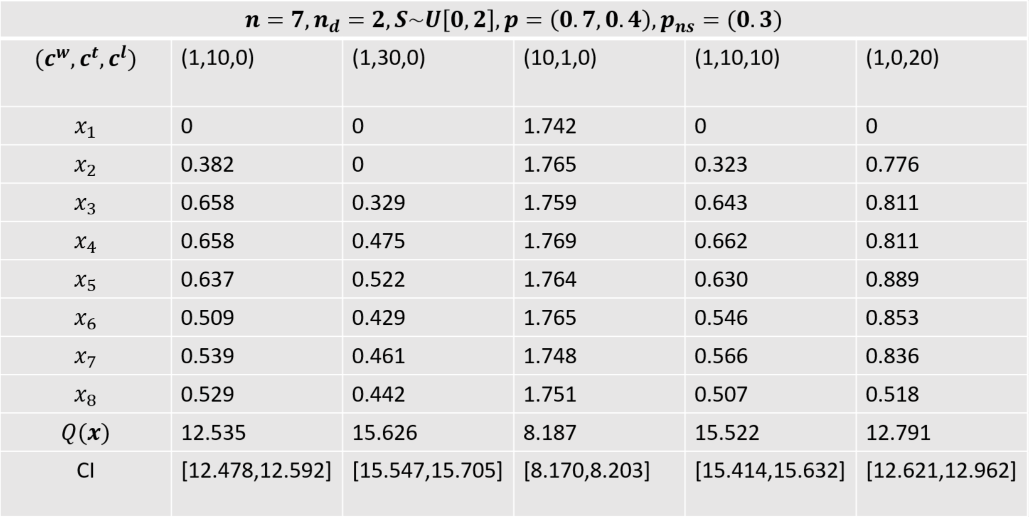
Figure 10: shows the effect of different sets of costs for the dynamic scheduling case. n is the number of customers scheduled upfront. n_d is the number of additional customers. p is the vector of probabilities for the arrival of a new customer.



$n = 2, n_d = 7, S \sim U[0,2], p = (0.7, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05)$						
(c^w, c^t, c^l)	(1,10,0)	(1,15,0)	(10,1,0)	(1,10,10)	(10,1,10)	(1,0,10)
x_1	0.233	0.153	1.828	0.229	1.679	1.399
x_2	0.737	0.644	1.829	0.725	1.641	1.367
x_3	0.860	0.813	1.834	0.870	1.511	1.156
x_4	0.893	0.835	1.833	0.883	1.237	1.021
x_5	0.904	0.872	1.829	0.913	1.166	0.710
x_6	0.907	0.867	1.845	0.879	1.227	0.707
x_7	0.993	0.836	1.854	0.906	1.265	0.777
x_8	0.937	0.521	1.869	0.732	1.233	1.083
$Q(\boldsymbol{x})$	2.798	3.010	2.034	2.957	3.747	0.788
CI	[2.785,2.811]	[2.996,3.025]	[2.027,2.041]	[2.942,2.972]	[3.731,3.763]	[0.781,0.796]

Figure 11: shows the effect of increasing the number of additional customers with respect to scheduled upfront customers.

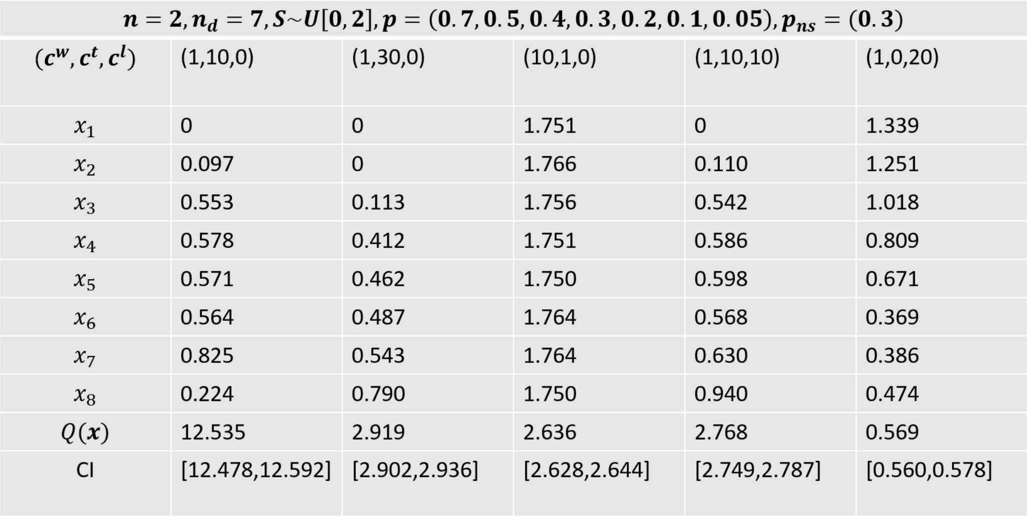
Next we examine the problem that accounts for no-shows. What draws our attention from the results in Figure 12 is that, for some sets of costs, the first appointment time is left empty. This behaviour is known as 'double-booking' and it is a common practice employed in appointment systems that deal with no-shows. In particular, this occurs when the ratio between idle time cost and waiting time cost is large. Clearly, when this is the case, there is a high incentive to minimise idle time. As a consequence, it is optimal to incur the risk of high waiting time early in the working day. Because of the recursive relation, waiting time piles up and reduces the chance of having any idle time for the following jobs. In this context, Figure 13 shows that increasing the number of additional customers with respect to scheduled upfront customers increases the slope of the distribution of $\mathbf{x}$, i.e. shorter time slots for the early jobs and longer time slots for the late jobs. Another remarkable result is highlighted by the last column, which presents a 0 idle time cost and a large overtime cost. From Figure 12 we can see that this combination of costs leads to double-booking in the first time slot. On the other hand, when no-shows are included, the effect found in Figure 11 is carried over instead; large time slots are assigned to early jobs. These decrease before slightly increasing again towards the end. This happens because, although the waiting time cost is low, we still have an incentive to minimise waiting time.



$\boldsymbol{n = 7, n_d = 2, S \sim U[0, 2], p = (0.7, 0.4), p_{ns} = (0.3)}$

$(\boldsymbol{c^w}, \boldsymbol{c^t}, \boldsymbol{c^l})$	(1,10,0)	(1,30,0)	(10,1,0)	(1,10,10)	(1,0,20)
x_1	0	0	1.742	0	0
x_2	0.382	0	1.765	0.323	0.776
x_3	0.658	0.329	1.759	0.643	0.811
x_4	0.658	0.475	1.769	0.662	0.811
x_5	0.637	0.522	1.764	0.630	0.889
x_6	0.509	0.429	1.765	0.546	0.853
x_7	0.539	0.461	1.748	0.566	0.836
x_8	0.529	0.442	1.751	0.507	0.518
$Q(\boldsymbol{x})$	12.535	15.626	8.187	15.522	12.791
CI	[12.478,12.592]	[15.547,15.705]	[8.170,8.203]	[15.414,15.632]	[12.621,12.962]

Figure 12: shows the effect of allowing for no-shows. p_{ns} is the probability of no-shows.



$n = 2, n_d = 7, S \sim U[0,2], p = (0.7, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05), p_{ns} = (0.3)$					
(c^w, c^t, c^l)	(1,10,0)	(1,30,0)	(10,1,0)	(1,10,10)	(1,0,20)
x_1	0	0	1.751	0	1.339
x_2	0.097	0	1.766	0.110	1.251
x_3	0.553	0.113	1.756	0.542	1.018
x_4	0.578	0.412	1.751	0.586	0.809
x_5	0.571	0.462	1.750	0.598	0.671
x_6	0.564	0.487	1.764	0.568	0.369
x_7	0.825	0.543	1.764	0.630	0.386
x_8	0.224	0.790	1.750	0.940	0.474
$Q(\boldsymbol{x})$	12.535	2.919	2.636	2.768	0.569
CI	[12.478,12.592]	[2.902,2.936]	[2.628,2.644]	[2.749,2.787]	[0.560,0.578]

Figure 13: shows the effect of increasing the number of additional customers with respect to scheduled upfront customers while no-shows are allowed.

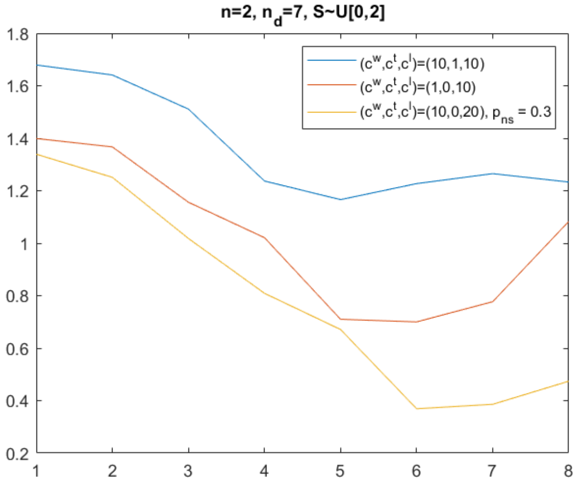
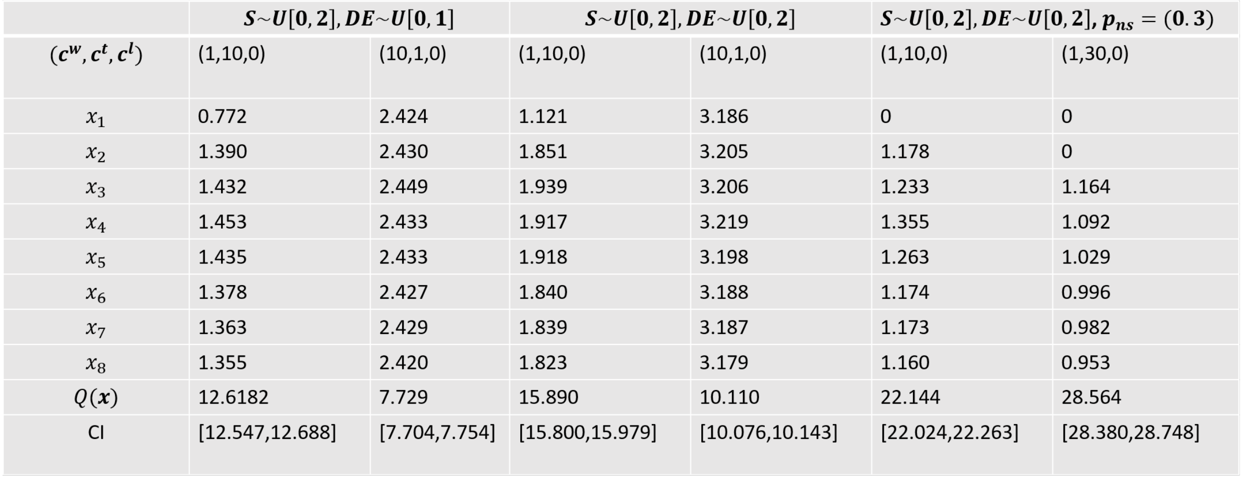


Figure 14: plot of x for a large number of additional customers with high waiting time and overtime costs. One instance includes no-shows.

Finally, for the last experiment we relax the assumption of punctuality by including a random delay. This was achieved by adding a random variable to the service time. Figure 15 shows the results obtained with a uniformly distributed delay. In the first instance, the delay is $U[0,1]$ distributed with mean 0.5. As we can see by comparing Figure 15 to Figure 10, the addition of delay leads to an approximate shift of 0.5 in the appointment times. We also note a slight increase in the total cost caused by the overall increase in variance of the random component. This result is again an application of proposition 1. In the second instance, the delay is scaled by a factor 2 hence its new mean is equal to 1. Here, the overall increase in appointment times is noticeably larger than 1. This is motivated by the larger variance of the delay which acts as a scaling factor for the service time. The last instance includes the probability of



	$\boldsymbol{S\sim U[0,2], DE\sim U[0,1]}$		$\boldsymbol{S\sim U[0,2], DE\sim U[0,2]}$		$\boldsymbol{S\sim U[0,2], DE\sim U[0,2], p_{ns}} = (\boldsymbol{0.3})$	
$(\boldsymbol{c^w, c^t, c^l})$	(1,10,0)	(10,1,0)	(1,10,0)	(10,1,0)	(1,10,0)	(1,30,0)
x_1	0.772	2.424	1.121	3.186	0	0
x_2	1.390	2.430	1.851	3.205	1.178	0
x_3	1.432	2.449	1.939	3.206	1.233	1.164
x_4	1.453	2.433	1.917	3.219	1.355	1.092
x_5	1.435	2.433	1.918	3.198	1.263	1.029
x_6	1.378	2.427	1.840	3.188	1.174	0.996
x_7	1.363	2.429	1.839	3.187	1.173	0.982
x_8	1.355	2.420	1.823	3.179	1.160	0.953
$Q(\boldsymbol{x})$	12.6182	7.729	15.890	10.110	22.144	28.564
CI	[12.547,12.688]	[7.704,7.754]	[15.800,15.979]	[10.076,10.143]	[22.024,22.263]	[28.380,28.748]

Figure 15: shows the effect of including a delay in our model. In columns 3 and 4 the delay is scaled by a factor 2. In columns 5 and 6 no-shows are included. $n = 7$ and $n_d = 2$.

no-shows and only analyses high idle time cost scenarios. Despite the 'double-booking' behaviour being carried over, the results are quite different than before. The first job after double-bookings presents an increase approximately equal to the mean of the delay. This is motivated by the fact that, in order to minimise idle time, we try to anticipate any delay from any of the double-booking customers.

5 Conclusion and outlook

5.1 Technical contributions

- Applied the sample average approximation method to the appointment scheduling problem and solved it with different variations of the L-shaped method (made a MATLAB program available);

- Confirmed the results obtained by Denton and Gupta (2003) with the sequential bounding method and compared them to the results obtained with SAA. The latter showed clear efficiency related advantages as it does not involve the computational effort of partitioning, yet it provides particularly accurate results;

- Combined the no-show model with the dynamic scheduling model from Denton and Erdogan (2011);

- Relaxed the assumption of punctuality by including delay in the model although the random variable utilized does not fully replicate a realistic behaviour (no idle time is generated before the arrival of late customers).

5.2 Insights into the appointment scheduling problem

- High idle time cost to waiting time cost ratios yield a peculiar 'dome-shape' distribution. The reason is that high idle time costs create an incentive to minimise idle time. This can be achieved by assigning short appointment times to the early jobs so that waiting time accumulates from the start and weighs negatively on idle time. Then, appointment times gradually increase to prevent waiting time from becoming too expensive. Finally, the appointment times decrease again towards the end;

- When no-shows are included in the model, a result that stands out is obtained for low waiting time costs. Depending on the magnitude of the ratio between overtime costs and waiting time costs (or the ratio between idle time costs and waiting time costs), the initial jobs present 'double-booking'. This occurs in order to minimise the chance of large idle time;

- The extended model presents a striking result when it is analysed for a high number of additional customers and irrelevant idle time costs. Here, initial appointment times are relatively long. They then decrease before increasing again in the end. The reason is that idle time costs require the minimisation of waiting time and overtime. Consequently, the aforementioned distribution is obtained.

5.3 Future research

- Elaborating the extended model by implementing a more realistic delay;

- analysing the problem with multiple servers;

- relaxing the i.i.d. service times assumption;

6 References

- Denton, B., Gupta, D., (2003) A sequential bounding approach for optimal appointment scheduling, IIE Transactions (2003) 35, 10031016, ISSN: 0740-817X print, DOI: 10.1080/07408170390230169

- Van Slyke, R. M., Wets, R. (1969) L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, SIAM Journal on Applied Mathematics, Vol. 17, No. 4 (Jul., 1969), pp. 638-663

- Erdogan, S., A., Denton, B., (2011) Dynamic Appointment Scheduling of a Stochastic Server with Uncertain Demand, INFORMS Journal on Computing, Articles in Advance, pp. 117, issn1091-9856, eissn1526-5528

- Shapiro, A., Philpott, A., (2007) A Tutorial on Stochastic Programming

- Verweij, B., Ahmed, S., Kleywegt, A., Nemhauser, G., Shapiro, A., (2001) The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study