Master Project Business Analytics

Routing and Scheduling of Truck Traffic

Karin van Eeden

CWI
Centrum Wiskunde & Informatica

VU
VRIJE
UNIVERSITEIT
AMSTERDAM

Master Project Business Analytics

Routing and Scheduling of Truck Traffic

Karin van Eeden

Centrum Wiskunde & Informatica
Research Group Stochastics
Science Park 123
1098 XG Amsterdam

VU University Amsterdam
Faculty of Sciences
De Boelelaan 1081a
1081 HV Amsterdam

Supervisors:
Prof.dr. Rob van der Mei
Dr. Sandjai Bhulai

July 2014

Preface

With this thesis I conclude the master program Business Analytics at VU University in Amsterdam. Business Analytics is a multi-disciplinary study program, aimed at improving business processes using mathematics, computer science, and business management. The final part of this study program consists of an internship, which has in my case been carried out at Centrum Wiskunde & Informatica (CWI). CWI is the national research institute for mathematics and computer science in the Netherlands, and was founded in 1946.

The project I have been working on has been established in collaboration with Trinité Automation. Trinité is a software company operating in the field of dynamic traffic management with a clear vision: achieving a dynamic traffic management system for the whole Netherlands.

There are several persons I would like to thank. First of all, I would like to thank my supervisors Rob van der Mei and Sandjai Bhulai for their support and feedback. Thanks are also due to Fetsje Bijma, for being the second reader from the university. Furthermore, I would like to thank Daphne van Leeuwen, Peter van Seventer, and Frank Ottenhof of Trinité for their interest into the developed model and the discussions on traffic management. I wish them success with the integration of the program into the software, and the further developments. Finally, I would like to thank all colleagues at CWI for the great working atmosphere.

Karin van Eeden
July 2014

Summary

In this thesis we present different models that can be used for the routing and scheduling of truck traffic near distribution centers. Here we consider the discrete-time setting in which the number of time intervals at which customers can be scheduled is limited. First we present a model based on outpatient appointment scheduling, consisting of two parts. In the first part, a general non-customer-specific schedule will be created that indicates how many truck drivers (or customers) need to be scheduled at a certain time interval. In the second part, a customer-specific schedule will be created by assigning each customer a time interval in which he/she could be served. The aim of the general scheduling model is to minimize a weighted sum of the customer's waiting time and the lateness of a schedule or the idle time of a crossdock. Here, the lateness of a schedule is given by the amount of time that is needed after the planned finish time in order to complete the schedule. Under certain assumptions – such as equally distributed loading and unloading times of trucks – the algorithm results in an optimal schedule. Additionally, it has been found that for many parameter settings the equally-spaced schedule is optimal.

Nevertheless, the appointment scheduling model does not scale well to larger systems due to the long run time (multiple hours). Furthermore, the model can hardly be extended or adjusted with additional requirements and preferences that are desired to take into account. Besides, we show that the expected length of the loading and unloading time of trucks does have a large impact on the expected waiting time of all subsequent scheduled customers.

As alternative to the appointment scheduling model, we present a multi-crossdock job shop scheduling model that can be used to schedule trucks at a distribution center. The advantage of this model is that many different extensions and preferences can be taken into account, whereas the model has a short run time (a few seconds). For multiple service time (or loading and unloading time) distributions a linear relationship has been found between the average service time and the service time delay. We will shown how this relationship can be used to reduce the waiting time of customers within a schedule, if customers have different service time distributions. The job shop scheduling model outperforms the appointment scheduling model in several ways. However, we recommend to validate the developed scheduling model using real data, or by means of a real test case.

Additionally, we investigate how trucks should be routed to (temporary) parking areas in order to use the number of available parking places at the distribution center efficiently. Here, the aim is to find a routing policy that minimizes the costs, which can for instance be the travel time or travel distance that corresponds to a certain route. We develop a simulation model that can be used to to evaluate the performance of different routing policies. When there are no capacity limitations, then it is optimal to route trucks to the parking area with the lowest costs. However, when there are capacity limitations, then this policy can be far from optimal. We illustrate that in some cases a policy in which a certain number of parking places is ‘reserved’ for specific types of customers is significantly better.

Contents

Chapter 1

Introduction

In this chapter some background information will be given on routing and scheduling of truck traffic. First, in Section 1.1 we will give a description of the problem and present a number of research questions. Second, in Section 1.2 the objectives of this study will be described, whereafter in Section 1.3 some background information will be given on air pollution, which plays an important role in this research. Next, in Section 1.4 some terminology will be introduced which will be used throughout this report. Finally, in Section 1.5 the structure of this report will be described.

1.1 Problem description

Several distribution centers that are located in the Netherlands deal with the congestion of trucks during busy hours. In general, at these distribution centers trucks arrive at random moments during the day in order to deliver or pick up goods. Since these distribution centers just have a limited capacity, truck drivers often experience high waiting times during peak hours. In addition to these long, annoying, waiting times, the amount of available space, or the number of available parking places plays an important role. Regularly, the number of available parking places is a bottleneck, which results in truck traffic on the access routes towards the distribution center.

The congestion of trucks is not only disturbing for the truck drivers themselves, but it is also annoying for other road users and damaging to the environment. For instance, when all of the available parking places are occupied, arriving trucks need to wait (in the near neighborhood) until a parking place becomes available. However, in some areas no alternative parking places are present, by which the truck drivers are either forced to wait on the access road, or to drive around in the neighborhood until a parking place becomes available. It is not only disturbing for the truck drivers themselves to be unnecessarily on the road, but it is also annoying for the other road users. The presence of trucks in crowded areas can lead to risky situations, or even can lead to accidents. Moreover, trucks need in general more time to speed up and slow down than other road users, which limits the flow of traffic around traffic lights and roundabouts.

Another argument for reducing the amount of truck traffic around distribution centers is air pollution. The pollution of motor vehicles consists of several substances including particulate matter (PM), nitrogen oxides (NO_x) and carbon monoxide (CO). Air pollution can have a negative effect on the health and well-being of people. Hence, several international and national standards exist for the emission of pollutants, see for instance European Union (2008). Air pollution is an important subject of interest: the government, provinces, and local authorities all take actions to improve the quality of air. More information on air pollution – especially in the Netherlands – will be given in Section 1.3.

One of the approaches that can be used to control the arrival process of trucks, and thus consequently reduce the waiting time of truck drivers and diminish the amount of air pollution, is by scheduling all truck drivers at the distribution center throughout the day. In this report it will be investigated how such a schedule can be created, whilst several predetermined constraints are met. Next to that, it will be investigated how trucks should be routed towards temporary parking areas in the near neighborhood of the distribution center. This will be done in such a way that the blocking probability – i.e., the probability that a truck arrives at a parking area which is completely occupied – is minimal.

1.2 Research objectives

In this study we will develop a mathematical model that can be used to control the number of trucks that arrive at a distribution center. This will be done by creating a schedule in such a way that the waiting times of truck drivers are minimized, whilst certain predetermined constraints are met. The aim of the model is to create a schedule in such a way that customers are scheduled as closely as possible to their preferred time interval. Multiple extensions will be discussed. Next, it will be investigated how temporary parking areas in the neighborhood of the distribution center can be used, and which routing policy should be used in order to use all available parking places efficiently.

1.3 Air pollution

In this section we will give some background information on air pollution, which is one of the main underlying reasons for this study. Air pollution is an important subject of interest worldwide. Air pollution caused by motor vehicles consists of several substances, including particulate matter (PM), hydrocarbons (HC), nitrogen oxides (NO_x), and carbon monoxide (CO). A large number of studies indicate that these substances do have a negative effect on health. See for instance Gehring et al. (2010), Jerrett et al. (2009), and Künzli et al. (2000).

In many countries emission standards are applicable. The European Union has drawn for instance multiple standards with the aim to keep the amount of air pollution down to a minimum. Specifically, the yearly-average European limit for nitrogen dioxides (NO_2) is given by 40 μg/m^3. In addition to such yearly-average limits there exist many other emission standards, see for example European Union (2008).

In the Netherlands the quality of air is closely monitored by the National Air Quality Monitoring Network (LML), a network developed by the National Institute for Public Health and the Environment (RIVM). LML measures every hour at different locations the concentration of several substances that indicate polluted air, including the elements as mentioned above (Nguyen et al., 2009). In Figure 1.1 the yearly-average level of NO_2 emission is shown of the Netherlands during 2013. As

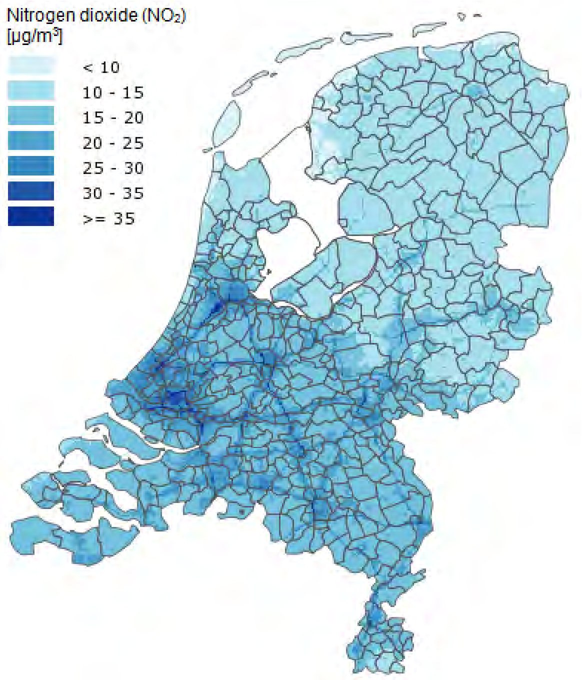


Figure 1.1: Yearly-average level of NO_2 emission in the Netherlands during 2013 (National Institute for Public Health and the Environment (RIVM), 2014).

can be seen in this figure is that the amount of emission differs per location; high levels of emission are mainly found around highways, especially around Amsterdam and Rotterdam. In Figure 1.2 detailed measurements of the level of NO_2 around Rotterdam are shown. From this figure it can be seen that the yearly-average standard of NO_2 emission is exceeded at several locations during 2012.

1.4 Terminology

Some terminology that will be used throughout this report will be clarified in this section. Within this report 'trucks' or 'truck drivers' will also be denoted by 'customers', and the 'loading and unloading time' of a truck will also be called the 'service time'. In fact, the truck drivers can be seen as customers; these customers are served by the distribution center. Here, the service time corresponds to the time that the truck occupies a crossdock.

1.5 Structure of the report

The remaining part of this report is organized as follows. First, in Chapter 2 a literature overview will be given of several subjects that will be treated in or which are relevant for the next chapters. Second, in Chapter 3 we present a model based on outpatient appointment scheduling which can be used to schedule trucks at a distribution center. Instead of scheduling patients at a doctor we schedule trucks at a distribution center. The advantage of this model is that it comes up with

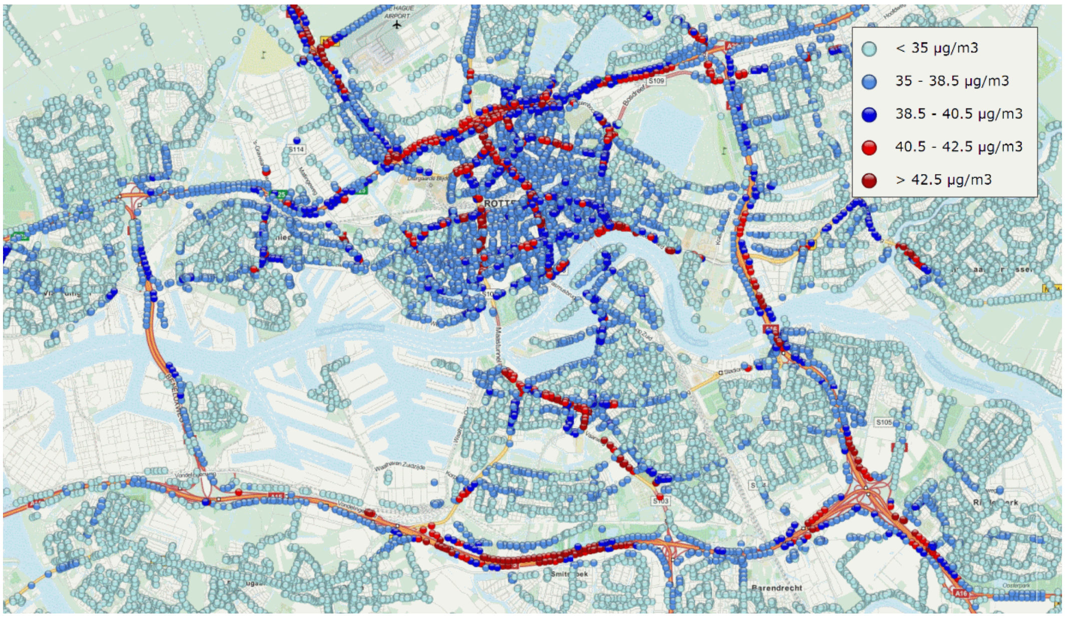


Figure 1.2: Level of NO_2 emission during 2012 around Rotterdam (Dutch National Air Quality Cooperation Programme (NSL), 2014).

an optimal schedule under certain assumptions. Nevertheless, this appointment scheduling model has several limitations and cannot easily be extended. Therefore, we present in Chapter 4 a different model to schedule trucks at a distribution center, which is based on job shop scheduling. This model is much more comprehensive than the appointment scheduling model; many different extensions and restrictions will be discussed. Next, in Chapter 5 multiple performance measures will be presented which can be used to evaluate the performance of a schedule. Within this chapter for both the appointment scheduling model and job shop scheduling model the strengths and weaknesses will be discussed; several examples will be given. In Chapter 6 we discuss how trucks can be routed to temporary parking areas. In this chapter the performance of different routing policies will be shown. Finally, in Chapter 7 several conclusions and directions for further research will be given.

Chapter 2

Literature Review

In this chapter we will give a literature overview of several topics that are covered within this report. First, in Section 2.1 some literature on outpatient appointment scheduling will be discussed. In fact, the underlying problem of outpatient appointment scheduling is similar to those of truck scheduling: instead of scheduling patients at a doctor we schedule trucks at a distribution center. Hence, the outpatient appointment scheduling models will be used as the starting point. Nevertheless, these models have several limitations; job shop scheduling will be used as alternative approach, which will be discussed in Section 2.2. Finally, in Section 2.3 the distribution of the loading and unloading times of trucks will be discussed. In all of the models that are presented in this report the loading and unloading times of trucks plays an important role.

2.1 Appointment scheduling

Outpatient appointment scheduling arises in hospitals and other medical institutions and boils down to the scheduling of patients at a doctor in such a way that both the interests of the patients and doctors are balanced. On the one hand, doctors want to have as little idle time as possible and thus prefer short interarrival times between patients. On the other hand, patients want to wait as little as possible and thus prefer large interarrival times.

Outpatient appointment scheduling has been subject of interest over the last decades. One of the first studies in this area were conducted by Welch and Bailey (1952). By means of an empirical study they showed that patients are usually early instead of late. Besides, in order to prevent extensive waiting of a doctor, they suggest as rule of thumb to start a schedule with two patients and then schedule the other patients evenly spaced throughout the day. This scheduling rule is also known as the *Bailey-Welch rule*.

Subsequent to the results of Welch and Bailey a large number of papers have been published in the field of outpatient appointment scheduling. This literature can roughly be divided into two categories: one using a simulation-based approach, and one using an analytical approach. For a comprehensive review of the existing literature and different solution approaches we refer to Cayirli and Veral (2003). In this study we focus on the discrete-time setting in which the number of available time intervals is limited. Within this area some work is done by Bosch et al. (1999), Bosch and Dietz (2001), Kaandorp and Koole (2007), and Koeleman and Koole (2012). All of these papers aim to minimize a weighted sum the patient's waiting time and the lateness of the schedule or the doctor's idle time. Bosch et al. (1999) propose an approach to find a lower and upper bound for such a schedule. These bounds can be found starting from a specific schedule. In order to find these bounds they use that their cost function is convex and, what they call, submodular. The results of Bosch et al. (1999) are extended with different types of customers and no-shows in Bosch and Dietz (2000) and Bosch and Dietz (2001).

In line with these papers, Kaandorp and Koole (2007) present a local search procedure that converges to an optimal schedule, starting from any schedule. In this paper it is assumed that the service times of patients are exponentially distributed. Koeleman and Koole (2012) extend this model and relax on the assumption that the service times are exponentially distributed. Their algorithm can be used for any service time distribution and includes emergency arrivals and no-shows. However, the computation time of the latter two algorithms is exponential in the number of intervals. Hence, for instances with a large number of time intervals the computation time is quite long.

2.2 Job shop scheduling

An extensive body of literature exists in the field of job shop scheduling. In this field, a frequently used term is the *makespan* of a machine (or schedule), which is defined as the total time that is required to process all jobs. The problem of minimizing the makespan of a schedule, given that there are multiple machines, is considered to be NP-hard. Several approximation algorithms exist in the literature, see for instance Angalakudati et al. (2014), Bübül and Kaminsky (2013), Neumann and Witt (2010), and Zalzala and Fleming (1997). These algorithms use different approaches, including branch and bound, graph theory, genetic algorithms, and LP-based heuristics.

2.3 Loading and unloading times

One of the main components of truck scheduling is the distribution of the loading and unloading times of trucks. In many (appointment) scheduling models the service times are assumed to be exponentially distributed. As far as we know, just a small number of studies have been done in which the distribution of the loading and unloading times of trucks is investigated. Kiesling and Walton (1995) showed in an empirical study about wharf crane operations in shipping ports that the service times within these wharfs are not necessarily exponentially distributed. Based on test results of multiple data sets, they conclude that very tight or very broad distributions are generally appropriate. These results are also obtained by Franz and Stolletz (2012), who showed in an empirical study that the service time distribution of trucks at an air cargo terminal is right-skewed.

More recently, The Tioga Group (2013) analyzed the truck turn times at Vancouver's container terminals. In this empirical study, they showed that the truck turn time – which is defined as the sum of the waiting time and dwell time – differs per terminal. From their data they obtained an average truck turn time of 56 minutes, averaged over three different terminals. Besides, they illustrated that the distribution of the time that is needed to import goods does have a quite different shape than the distribution of the time that is needed to export goods. This suggests to take different types of customers with different service time distributions into account in the model. However, their graphical illustrations clearly show that the service time distribution is right-skewed. Additionally, it was found that the number of trucks that were present in the terminal are subjected to a day pattern, i.e., during the day several peak hours were present.

In addition to the empirical studies that show differences in the loading and unloading times of trucks, models have been developed that estimate the loading or unloading time of a truck. For example, Fatthi et al. (2013) present a decision model for estimating the unloading time of incoming trucks in crossdocking terminals on the basis of three factors: the number of purchase orders carried by supplier, the variation of items listed in the purchase order and the quantity of boxes that were carried by the truck. This model can be used at one of the last states, when this kind of information is known. At earlier moments during the order and delivery process, this information is often not known and thus more general methods for estimating the unloading time are required.

Chapter 3

Appointment Scheduling

As mentioned in Chapter 2, the underlying problem of truck scheduling corresponds to that of outpatient appointment scheduling. Instead of scheduling patients at a doctor we schedule trucks at a distribution center. For an introduction to appointment scheduling and a literature review we refer to Section 2.1.

The appointment scheduling model that will be presented in this chapter can be used to schedule trucks at a single crossdock in advance. The model consists of two parts. In the first part, which will be discussed in Section 3.1, a general non-customer-specific schedule will be created that indicates how many customers need to be scheduled at a certain time interval. In the second part, which will be presented in Section 3.2, a customer-specific schedule will be created by assigning each customer a time interval in which he could be served.

3.1 General scheduling model

In this section the first part of the appointment scheduling model will be described in which it will be determined how many customers should be scheduled at each time interval on a single crossdock. The aim of this model is to minimize a weighted sum of the customer's waiting time and the lateness of a schedule or the idle time of a crossdock. Here, the lateness of a schedule is given by the amount of time that is needed after the planned finish time in order to complete the schedule. To this end, it is assumed that the service times (i.e., the loading and unloading times) of all customers are equally distributed. Nevertheless, the model is applicable for any service time distribution and unscheduled high-priority customers that arrive during the day can be taken into account in some sense. The presented model and algorithm is based on Koeleman and Koole (2012). However, several parts of the model will be extended or adjusted, and some small errors that have been found within this paper will be eliminated. For clarity different notation will be used.

In order to keep the model implementable and executable on various computers without requiring advanced software packages, we present a model with a discrete service time distribution. Nevertheless, the model can also be used for continuous service time distributions. This can be done by approximating the continuous probability density function by a discrete probability distribution function (pdf). This can be done in the following way. Suppose we have a continuous random variable X_c with probability density function $f_{X_c}(\cdot)$. Then, the approximated pdf of the corresponding discrete random variable X_d can be given by

$$\begin{aligned}\mathbb{P}(X_d = 0) &= \int_0^{0.5} f_{X_c}(x)\,\mathrm{d}x,\\ \mathbb{P}(X_d = k) &= \int_{k-0.5}^{k+0.5} f_{X_c}(x)\,\mathrm{d}x, \quad k \in \mathbb{N}_1,\end{aligned}$$

where k could be in any time unit. Here, $\mathbb{N}_1$ represents the set of positive natural numbers $\{1, 2, 3, \ldots\}$. Note that the smaller the time unit of k is chosen, the more accurate the approximated pdf is. In addition, one could also use the empirical service time distribution.

3.1.1 Model description

Suppose that N customers need to be scheduled at T time intervals, where each time interval has length Δ. Here, any time unit – such as minute, quarter or hour – could be chosen, as long as it is used consistently. Let B_s and B_u be random variables representing the service time of scheduled and unscheduled high-priority customers respectively. Assume that the corresponding service time distributions are known and that they have an average of β_s and β_u time units respectively. To be clear, all scheduled customers are assumed to have the same service time distribution; the same assumption is made for unscheduled high-priority customers that arrive during the day.

For simplicity we assume that unscheduled high-priority customers arrive according to a homogeneous Poisson process with rate λ. That means that per time interval on average λ unscheduled customers arrive. However, the described model can easily be extended to a non-homogeneous arrival process of unscheduled high-priority customers by taking for each time interval t a different arrival rate. Unscheduled high-priority customers are served before scheduled customers in the order of arrival; besides, all customers are assumed to arrive just at the beginning of a time interval. When Δ is chosen small enough, this may be a reasonable assumption. However, for large Δ this assumption becomes less sensible.

Next, let n_t be the number of scheduled customers at time t. A schedule is defined as a vector $x = (n_1, \ldots, n_T)$ where $\sum_{t=1}^{T} n_t = N$, and $n_t \in \{1, \ldots, N\}$ for $t = 1, \ldots, T$. Let $W(x)$ be the expected waiting time, $L(x)$ the expected lateness and $I(x)$ the expected idle time of schedule x. The lateness of a schedule is the amount of time that is needed after time interval T to complete the schedule. The cost function of schedule x is defined as $C(x) = \alpha_W W(x) + \alpha_I I(x) + \alpha_L L(x)$, where α_W, α_I, and α_L are the weights of the waiting time, idle time, and lateness respectively. Now define Y as the number of unscheduled high-priority customers that arrive at the beginning of any time interval, and let S_i be a random variable indicating the number of time units of work that arrives at the beginning of any time interval, given that i customers are scheduled to arrive. Consequently, S_i consists of both scheduled and unscheduled work. Then, the pdf of the number of time units of unscheduled arriving work is given by

$$\begin{aligned}
\mathbb{P}(S_0 = 0) &= \mathbb{P}(Y = 0) + \sum_{k=1}^{\infty} \mathbb{P}(Y = k)\mathbb{P}(kB_u = 0), \\
\mathbb{P}(S_0 = j) &= \sum_{k=1}^{\infty} \mathbb{P}(Y = k)\mathbb{P}(kB_u = j), \qquad j \in \mathbb{N}_1,
\end{aligned}$$

where Y is Poisson distributed with parameter λ. For $i \in \mathbb{N}_1$, the pdf of the amount of arriving work is given by

$$\mathbb{P}(S_i = j) = \sum_{k=0}^{j} \mathbb{P}(S_0 = k)\mathbb{P}(iB_s = j - k), \quad j \in \mathbb{N}_0.$$

Now let X_t^- be a random variable indicating the amount of work in the system just before any arrivals at time interval t and let X_t^+ be the amount of work in the system just after any arrivals at time interval t, for $t = 1, \dots, T$. The pdfs of both variables are related to each other in the following way:

$$\begin{aligned}
&\mathbb{P}(X_1^- = 0) = 1, \\
&\mathbb{P}(X_1^+ = j) = \mathbb{P}(S_{n_1} = j), && j \in \mathbb{N}_0, \\
&\mathbb{P}(X_t^- = 0) = \sum_{k=0}^{\Delta} \mathbb{P}(X_{t-1}^+ = k), && t = 2, \dots, T+1, \\
&\mathbb{P}(X_t^- = j) = \mathbb{P}(X_{t-1}^+ = j + \Delta), && j \in \mathbb{N}_1,\ t = 2, \dots, T+1, \\
&\mathbb{P}(X_t^+ = j) = \sum_{k=0}^{j} \mathbb{P}(X_t^- = k)\mathbb{P}(S_{n_t} = j - k), && j \in \mathbb{N}_0,\ t = 2, \dots, T.
\end{aligned}$$

By recursion, $\mathbb{P}(X_t^- = j)$ is given by

$$\begin{aligned}
&\mathbb{P}(X_1^- = 0) = 1, \\
&\mathbb{P}(X_2^- = 0) = \sum_{k=0}^{\Delta} \mathbb{P}(S_{n_1} = k), \\
&\mathbb{P}(X_2^- = j) = \mathbb{P}(S_{n_1} = j + \Delta), && j \in \mathbb{N}_1, \\
&\mathbb{P}(X_t^- = 0) = \sum_{k=0}^{\Delta} \sum_{m=0}^{k} \mathbb{P}(X_{t-1}^- = m)\mathbb{P}(S_{n_{t-1}} = k - m), && t = 3, \dots, T+1, \\
&\mathbb{P}(X_t^- = j) = \sum_{k=0}^{j+\Delta} \mathbb{P}(X_{t-1}^- = k)\mathbb{P}(S_{n_{t-1}} = j + \Delta - k), && j \in \mathbb{N}_1, \\
& && t = 3, \dots, T+1.
\end{aligned}$$

The pdf of X_t^- will be used to evaluate the costs $C(x)$ of schedule x. How this is done will be explained in the next subsection.

3.1.2 Cost function evaluation

Below we will give an expression for each of the three elements of the cost function $C(x) = \alpha_W W(x) + \alpha_I I(x) + \alpha_L L(x)$. These components are dependent on each other; hence, by calculating them in the same ordering as listed below and reusing the results, the calculations can be performed efficiently.

Waiting time The expected waiting time $W(x)$ of a schedule can be determined by summing the expected waiting times for all scheduled customers. Note that this is a recursive process: the waiting time of a certain customer depends on the waiting and service time of the previously scheduled customer. To this end, define $W_{i,t}$ as the waiting time of the ith scheduled customer at time interval t. We have

$$\begin{aligned}
\mathbb{P}(W_{1,1}=j) &= \mathbb{P}(S_0=j), && j\in\mathbb{N}_0,\\
\mathbb{P}(W_{1,t}=j) &= \sum_{k=0}^{j}\mathbb{P}(X_t^-=k)\mathbb{P}(S_0=j-k), && j\in\mathbb{N}_0,\ t=2,\dots,T,\\
\mathbb{P}(W_{i,1}=j) &= \sum_{k=0}^{j}\mathbb{P}(W_{i-1,1}=k)\mathbb{P}(B_s=j-k), && i=2,\dots,n_1,\ j\in\mathbb{N}_0,\\
\mathbb{P}(W_{i,t}=j) &= \sum_{k=0}^{j}\mathbb{P}(W_{i-1,t}=k)\mathbb{P}(B_s=j-k), && i=2,\dots,n_t,\ j\in\mathbb{N}_0,\\
& && t=2,\dots,T.
\end{aligned}$$

Then, the expected waiting time of schedule x is given by

$$W(x)=\sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{j=1}^{\infty} j\mathbb{P}(W_{i,t}=j).$$

Lateness The expected lateness $L(x)$ of a schedule is equal to the expected time that is needed after time interval T to finish the schedule. Hence, this is equal to the amount of work in the system just before time interval $T+1$, i.e.,

$$L(x)=\sum_{k=1}^{\infty} k\mathbb{P}(X_{T+1}^-=k).$$

Idle time The expected idle time $I(x)$ of a schedule is equal to the time in which the system is not working. This is given by the sum of the duration and the lateness of the schedule minus the time the system is working. Hence, we get

$$I(x)=\Delta T+L(x)-\lambda T\beta_u-N\beta_s.$$

3.1.3 Solution method

Summarized, the optimization problem that we want to solve is given by

$$\begin{aligned}
\text{minimize}\quad & C(x)\\
\text{subject to}\quad & \sum_{t=1}^{T} n_t = N,\\
& n_t\in\mathbb{N}_0.
\end{aligned}$$

The simplest way to solve this optimization problem is by means of brute force. However, in total there are $\binom{N+T-1}{N}$ different schedules possible; trying all these schedules will result in a huge computation time. Hence, in order to find the optimal solution, the local search procedure as described in Kaandorp and Koole (2007) will be used, which is much faster than brute force. This procedure starts with any feasible schedule, and improves the schedule iteratively by searching in the neighborhood for a schedule with lower costs. The neighborhood of a schedule consists of all schedules in which a set of customers is shifted to a different time interval. This search process continues until a local optimum is found. Kaandorp and Koole (2007) proved that their cost function is multimodular – a property that is related to convexity – and that due to this property the local search procedure converges to a global optimum. Koeleman and Koole (2012) extended this proof and showed that their cost function is also multimodular. The cost function $C(x)$ as given in this chapter is similar to those of the latter model, and is thus multimodular. Hence, for this specific cost function, the local search procedure as described in Kaandorp and Koole (2007) will converge to a global optimum.

3.1.4 Numerical results

Many different combinations of input parameters are possible; depending on these parameters different schedules are optimal. Especially the weights α_W, α_I, and α_L of the waiting time, idle time, and lateness respectively do influence the resulting optimal schedule. However, in practice these weights are not clearly defined. Since the optimal schedule strongly depends on the number of time intervals and the number customers to be scheduled, we will not illustrate all these different schedules. Instead we will summarize the results and conclusions that can be drawn from the numerical experiments.

Service time distributions The most standard schedule is the evenly-spaced schedule. When the service times are not submissive to variability, i.e., the service times are deterministic, then an evenly spaced schedule is optimal. When the service times are submissive to variability, and they are for instance exponentially or normally distributed, then a small number of customers are shifted to an earlier time interval, compared to the equally-spaced schedule. The higher the coefficient of variation of the service time distributions is, the more customers are shifted to the beginning of the schedule.

Parameter weights Depending on the weights of the customer's waiting time α_W, the idle time of the crossdock α_I, and the lateness of the schedule α_L, the resulting optimal schedules can be quite different. Numerical experiments have shown that as α_W increases with respect to α_L that the optimal schedule tends to the equal-spaced schedule. Even though, when α_W is significantly larger than α_L, then in the optimal schedule customers are slightly shifted towards the end of the schedule. In this case, customers are often scheduled evenly-spaced at the beginning of the schedule; a slightly larger number of customers are scheduled at the end of the schedule. In these type of schedules customers have large interarrival times, resulting in low waiting times.

Unscheduled high-priority customers Numerical experiments have shown that the more unscheduled high-priority customers are expected to arrive, the more customers are scheduled at the beginning of the schedule. The probability that a high-priority customer has arrived at the beginning of the schedule is very small; therefore empty space will result in unnecessary idle time by which the lateness will also increase.

3.2 Customer assignment

As mentioned before, the appointment scheduling model consists of two parts. In this section the second part of the algorithm will be described, in which a customer-specific schedule will be created, using the resulting schedule of the model as described in Section 3.1.

3.2.1 Model description

Suppose that N customers need to be scheduled at T time intervals, where each time interval consists of Δ time units. Let the schedule start at time 0, and let the end time of the schedule be given by $T_{\text{end}} := \Delta T$. Suppose that the time interval at which customer i prefers to be scheduled is given by $[a_i, b_i]$, for $0 \leq a_i \leq b_i \leq T_{\text{end}}$. Additionally, let $f(i,t)$ be a general function indicating the costs of scheduling customer i at time interval t. First create a schedule $x = (n_1, \ldots, n_T)$, according to the model as described in Section 3.1. Then, under the restriction that at time interval t exactly n_t customers are scheduled, a customer-specific schedule with minimal costs can be obtained by solving the *integer linear program* (ILP) as given below. To this end, define

$$x_{i,t} = \begin{cases} 1, & \text{if customer } i \text{ is scheduled at time interval } t, \\ 0, & \text{otherwise,} \end{cases}$$

and let $\mathcal{N} = \{1, \ldots, N\}$ be the set of customers that need to be scheduled. Denote with $\mathcal{T} = \{1, \ldots, T\}$ the set of time intervals at which the customers can be scheduled. Notice that if a customer has no preference with respect to the scheduled time, then the preferred time interval can be denoted by $[0, T_{\text{end}}]$. An assignment of customers to time intervals with minimal costs can be obtained by solving the following ILP:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} f(i,t) x_{i,t} && && (3.1\text{a}) \\ \text{subject to} \quad & \sum_{t \in \mathcal{T}} x_{i,t} = 1, && i \in \mathcal{N}, && (3.1\text{b}) \\ & \sum_{i \in \mathcal{N}} x_{i,t} = n_t, && t \in \mathcal{T}, && (3.1\text{c}) \\ & x_{i,t} \in \{0,1\}, && i \in \mathcal{N},\ t \in \mathcal{T}. && (3.1\text{d}) \end{aligned}$$

3.2.2 Cost function

The cost function $f(i,t)$ as given in (3.1a) is a general function indicating the costs of scheduling customer i at time interval t. Within this function many different factors could be taken into account. When it is for instance preferred to schedule customers as close as possible to their preferred time interval, then, when $c_{W,i}$ represents the waiting costs per unit of time for customer i, one could define the cost function

$$\begin{aligned} f(i,t) = c_{W,i}[a_i - \Delta(t-1)]\mathbb{1}\left\{t < \frac{a_i}{\Delta} + 1\right\} \\ + c_{W,i}[\Delta(t-1) - b_i + \beta_i]\mathbb{1}\left\{t > \frac{b_i - \beta_i}{\Delta} + 1\right\}. \end{aligned} \tag{3.2}$$

This cost function is linear, meaning that the costs increase linearly as a customer is scheduled further away from his/her preferred time interval. For a thoroughly description of the cost functions that can be used, we refer to Section 4.1.2.

3.3 Multi-crossdock scheduling

The model as presented in the previous sections can be used to schedule customers at a single crossdock. However, in practice there will often be multiple crossdocks available; hence, an appropriate schedule for multi-crossdock systems is desired. When the service times of all customers are equally distributed, and when there are no additional constraints, then the optimal multi-crossdock schedule can be reduced to multiple single-crossdock schedules. Then within the optimal schedule, at each crossdock a similar number of customers is scheduled. Specifically, the number of customers that is scheduled at each crossdock differs at most by one customer.

Suppose that there are in total M crossdocks available; let $\mathcal{M} := \{1,\dots,M\}$ be the set of available crossdocks. Then, create a general schedule $x = (n_{1,1},\dots, n_{1,T},\dots,n_{M,1},\dots,n_{M,T})$, according to the model as described in Section 3.1. Here, $n_{j,t}$ denotes the number of customers that is scheduled at crossdock j at time interval t. When in total N customers need to be scheduled over M crossdocks, then at $N \pmod M$ crossdocks $\lceil N/M \rceil$ customers will be scheduled; at $M-N \pmod M$ crossdocks $\lfloor N/M \rfloor$ customers will be scheduled. For instance, when $N = 20$ customers need to be scheduled at $M = 3$ crossdocks, then at $20 \pmod 3 = 2$ crossdocks in total $\lceil 20/3 \rceil = 7$ customers will be scheduled; at $3 - 20 \pmod 3 = 1$ crossdock in total $\lfloor 20/3 \rfloor = 6$ customers will be scheduled.

When considering such a multi-crossdock system, customers can be assigned to a crossdock and time interval by solving the ILP as given in (3.3). To this end, define

$$x_{i,j,t} = \begin{cases} 1, & \text{if customer } i \text{ is scheduled at crossdock } j \text{ at time interval } t, \\ 0, & \text{otherwise.} \end{cases}$$

Let $f(i,j,t)$ be a general function indicating the costs of scheduling customer i at crossdock j at time interval t. Then, a customer-specific schedule with minimal costs can be obtained by solving the following optimization problem:

$$\begin{aligned}
&\text{minimize} && \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} f(i,j,t)x_{i,j,t} && && \text{(3.3a)}\\
&\text{subject to} && \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} x_{i,j,t} = 1, && i\in\mathcal{N}, && \text{(3.3b)}\\
& && \sum_{i\in\mathcal{N}} x_{i,j,t} = n_{j,t}, && j\in\mathcal{M},\ t\in\mathcal{T}, && \text{(3.3c)}\\
& && x_{i,j,t}\in\{0,1\}, && i\in\mathcal{N},\ j\in\mathcal{M},\ t\in\mathcal{T}. && \text{(3.3d)}
\end{aligned}$$

Nevertheless, when the service time distribution of customers are not equally distributed, or when other restrictions are applicable, then the presented appointment scheduling model may result in a quite bad schedule. The limitations of this model will be discussed below.

3.4 Conclusions and discussion

The appointment scheduling model as described in this chapter results in an optimal schedule under certain circumstances. The schedule is optimal in the sense that a weighted sum of the customer's waiting time and the lateness or the idle time of the schedule is minimized, whereas (as second objective) customers are scheduled as closely as possible to their preferred time interval. However, when the assumptions that are made – such as equally distributed service times – do not hold, then the resulting schedule can be quite bad. In practice customers regularly have different service time distributions, which is also shown by empirical studies (see Section 2.3). There may be customers that only have to deliver a small packet, while other customers need to pick up a large number of goods that should be fitted in the truck. When the service time of a customer that is scheduled at the beginning of the day is heavily delayed, then this influences the waiting time of all subsequent customers negatively. For an illustration of the influence of different service times we refer to Section 5.5. Summarized, it is desirable to take different service time distributions into account.

Nonetheless, there are many other restrictions that have a large impact on the resulting schedule. Some extensions or additional requirements that are desirable to take into account are:

- Customers with different service times distributions: some jobs require more time than other jobs.
- Multi-crossdock scheduling with for each customer limitations or preferences with respect to the crossdock at which he/she is scheduled.
- Preferences or limitations with respect to the time interval at which customers are scheduled. Some customers may for instance not be scheduled at the end of the day.
- Requirements with respect to the ordering in which customers are scheduled.

Some of these requirements can be taken into account in the second part of the model in which customers are assigned to a time interval. However, since these additional constraints are not included in the general scheduling part, the resulting schedule can be far from optimal. Even though it may be possible that no allowed schedule is found, whereas there exists an allowed schedule. Ideally, the described extensions should be included in the general scheduling part. Unfortunately, this cannot easily be done. As mentioned before, the cost function should be multimodular in order to ensure that an optimal solution is found. Then the question arises how all these constraints should be taken into account, in such a way that the cost function is multimodular.

Consider for instance the last listed requirement, which indicates that customers need to be scheduled in a certain order. Such a constraint could be included in the cost function by defining $C(x) = \infty$ if the customers are not served in the right ordering within schedule x. However, it can be proven that when this requirement is included in this way, that the cost function is not multimodular. Such troubles do also arise when other constraints are added. However, even though when the model can be extended with all these features, then the run time of the algorithm is so long that it is practically useless for real-time scheduling (see also Section 5.3). In the next chapter we present a different scheduling model, in which all of these extensions can be taken into account.

Chapter 4

Multi-crossdock Job Shop Scheduling

In this chapter we present a multi-machine job shop scheduling model that can be used to control the number of trucks that arrive during the day at a distribution center with multiple crossdocks. Consequently, the model can also be applied for single-crossdock systems. Job shop scheduling is about assigning jobs to machines in such a way that a certain objective function is optimized, while some predetermined constraints are met. Truck scheduling can be seen as a certain type of job shop scheduling. The trucks that need to pick up or deliver goods at the distribution center can be seen as jobs, and the distribution center can be seen as machine or manufacturing system.

There exist many different job shop scheduling problems, for which various algorithms and heuristics exist. However, many of these algorithms are developed for special cases and cannot be easily extended with other advanced options. In this chapter we will use *linear programming* to solve the job shop scheduling problem. Initially we assume that all customers (or trucks) arrive on time, and that each customer has a deterministic loading and unloading time, also to be called the *service time* in the sequel.

This chapter is organized as follows. First, in Section 4.1 we present the basic job shop scheduling model that can be used to schedule trucks in advance. Second, in Section 4.2 a number of different scenarios and extensions will be described that can be added to the model. In Section 4.3 multiple examples will be given, illustrating the model. Next, in Section 4.4 we describe how unscheduled (high-priority) customers that arrive during the day can be added to an ongoing schedule. Finally, in Section 4.5 an example will be given of rescheduling during the day.

4.1 Scheduling in advance

In this section we present a model that can be used to schedule trucks in advance. This means that the schedule will be created a considerable amount of time ahead, say one day. It is assumed that at the moment of scheduling all essential information – such as the service times and preferred time intervals – is known. This can for instance be achieved by letting all truck drivers sign up in advance via a website or mobile phone application. Once the schedule is made, the result will be communicated to the customers such that they know in advance at which time they should be at the distribution center.

4.1.1 Model description

Consider a time period of T intervals, each of length Δ, which is a multiple of some time unit like minute, quarter or hour. The model that will be presented is independent of the chosen time unit, as long as it is used consistently. Let the schedule start at time 0, and denote with $T_{\text{end}} = \Delta T$ the time at which the schedule should be finished. Suppose that we want to schedule N customers on M different crossdocks during this time period. Let the service time of customer i be $\beta_i > 0$ time units and suppose that customer i prefers to be served between time $[a_i, b_i]$, where $0 \leq a_i \leq b_i \leq T_{\text{end}}$. Without loss of generality it can be assumed that each customer has such a preferred time interval. In particular, when a customer does not have any preference, then the 'preferred' time interval can be denoted by $[0, T_{\text{end}}]$.

In order to keep the notation of the model concise, denote by $\mathcal{N} = \{1, \ldots, N\}$ the set of customers that need to be scheduled, $\mathcal{M} = \{1, \ldots, M\}$ the set of available crossdocks, and $\mathcal{T} = \{1, \ldots, T\}$ the set of time intervals for which the schedule should be made. Basically, when a customer is scheduled at a certain time interval, it is assumed that he is on time and could go immediately in service at the beginning of the scheduled time interval. Furthermore, define

$$x_{i,j,t} = \begin{cases} 1, & \text{if customer } i \text{ is scheduled at crossdock } j \text{ at time interval } t, \\ 0, & \text{otherwise,} \end{cases}$$

$$s_{i,j,t} = \begin{cases} 1, & \text{if customer } i \text{ is served at crossdock } j \text{ during time interval } t, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$d_{i,j} = \begin{cases} 1, & \text{if customer } i \text{ is allowed to be scheduled at crossdock } j, \\ 0, & \text{otherwise.} \end{cases}$$

Let $D_i = \lceil \beta_i / \Delta \rceil$ be the (upwards rounded) number of time intervals in which customer i will be in service. Define $\mathcal{T}_i^{\text{end}}$ as the set of time intervals at which customer i is not allowed to be scheduled in order to finish the schedule before T_{end}, i.e.,

$$\mathcal{T}_i^{\text{end}} = \{T - D_i + 2, \ldots, T\}. \tag{4.1}$$

Additionally, let $f(i, j, t)$ be a general function indicating the costs of scheduling customer i at crossdock j at time interval t. It is assumed that at most one customer can be served at the same time. A schedule with minimal costs can be obtained by solving the following *integer linear program* (ILP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} f(i,j,t)x_{i,j,t} && && \text{(4.2a)}\\
\text{subject to} \quad & \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} x_{i,j,t} = 1, && i\in\mathcal{N}, && \text{(4.2b)}\\
& s_{i,j,t} = \sum_{s=\max\{1,t-D_i+1\}}^{t} x_{i,j,s}, && i\in\mathcal{N},\ j\in\mathcal{M},\ t\in\mathcal{T}, && \text{(4.2c)}\\
& \sum_{i\in\mathcal{N}} s_{i,j,t} \le 1, && j\in\mathcal{M},\ t\in\mathcal{T}, && \text{(4.2d)}\\
& \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_i^{\text{end}}} x_{i,j,t} = 0, && i\in\mathcal{N}, && \text{(4.2e)}\\
& \sum_{t\in\mathcal{T}} x_{i,j,t} \le d_{i,j}, && i\in\mathcal{N},\ j\in\mathcal{M}, && \text{(4.2f)}\\
& x_{i,j,t}\in\{0,1\}, && i\in\mathcal{N},\ j\in\mathcal{M},\ t\in\mathcal{T}. && \text{(4.2g)}
\end{aligned}$$

Here, the objective (4.2a) is to minimize the costs of a certain schedule. Constraint (4.2b) indicates that each customer needs to be scheduled exactly once, constraint (4.2c) denotes the relation between $x_{i,j,t}$ and $s_{i,j,t}$, and constraint (4.2d) ensures that at each crossdock at most one customer is served at the same time. Furthermore, constraint (4.2e) requires that all customers are served before T_{end}. Note that this constraint can also be written as

$$x_{i,j,t} = 0, \quad j\in\mathcal{M},\ t\in\mathcal{T}_i^{\text{end}},$$

which boils down to exactly the same requirement. However, the notation used in (4.2e) reduces the number of constraints significantly. In addition, constraint (4.2f) makes sure that each customer is scheduled at a crossdock where he/she is allowed to be served.

Note that in the single-crossdock case (i.e., $M = 1$), the model could be simplified by removing all indices that indicate the crossdock. Hence, if we define

$$x_{i,t} = \begin{cases} 1, & \text{if customer } i \text{ is scheduled at time interval } t,\\ 0, & \text{otherwise,}\end{cases}$$

$$s_{i,t} = \begin{cases} 1, & \text{if customer } i \text{ is served during time interval } t,\\ 0, & \text{otherwise,}\end{cases}$$

and let $f(i,t)$ be a general function indicating the costs of scheduling customer i at time interval t, then the single-crossdock ILP is given by

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} f(i,t) x_{i,t} && && (4.3\text{a}) \\
\text{subject to} \quad & \sum_{t \in \mathcal{T}} x_{i,t} = 1, && i \in \mathcal{N}, && (4.3\text{b}) \\
& s_{i,t} = \sum_{s=\max\{1, t-D_i+1\}}^{t} x_{i,s}, && i \in \mathcal{N},\ t \in \mathcal{T}, && (4.3\text{c}) \\
& \sum_{i \in \mathcal{N}} s_{i,t} \leq 1, && t \in \mathcal{T}, && (4.3\text{d}) \\
& \sum_{t \in \mathcal{T}_i^{\text{end}}} x_{i,t} = 0, && i \in \mathcal{N}, && (4.3\text{e}) \\
& x_{i,t} \in \{0,1\}, && i \in \mathcal{N},\ t \in \mathcal{T}. && (4.3\text{f})
\end{aligned}
$$

The described ILPs can be solved by using a software package as for instance CPLEX or Gurobi. Several benchmarks have been conducted to compare the performance of these software packages, see for instance Koch et al. (2011), Meindl and Templ (2012), and Gurobi Optimization (2013). Without going into detail, we have chosen to use Gurobi for the implementation of the described ILP.

4.1.2 Cost function

The function $f(i,j,t)$ as shown in Equation (4.2a) is a general function indicating the costs of scheduling customer i at crossdock j at time interval t. Within this cost function many different factors can be taken into account. Take for instance the priority of a customer. By defining different costs for different types of customers, the customers' priority can be included in the model. The priority of a customer can for instance be determined by the monthly average reward that is obtained by the distribution center, or the frequency at which the customer visits the distribution center. In this subsection we illustrate two types of cost functions: a linear and quadratic one. Though, next to these examples many other types of cost functions are possible.

Next to the priorities of customers that can be taken into account, preferences with respect to the crossdock at which customers are scheduled can be taken into account. This can be done by specifying for each crossdock different costs. In some cases a certain crossdock will be preferred above another crossdock. Suppose that a certain customer needs to deliver goods that should end up in repository A. In order to avoid unnecessary transport of goods, it is preferred to let this customer deliver the goods directly at a crossdock by repository A. However, if it does not fit in the schedule to unload the truck directly at repository A, then a second option could be to unload the truck at another crossdock, and then transport the goods to repository A. This option is preferred over refusing the customer, but is not the first choice. Hence, in this case one could specify for the considered customer high costs for all crossdocks that are not located at repository A.

In the remainder part of this subsection we will illustrate two types of cost functions. First, we consider a *linear cost function* in which the costs increase linearly as a customer is scheduled further away from his/her preferred time interval. Let $c_{W,i,j}$ represent the waiting costs per time unit of scheduling customer i at crossdock j. Then, the priority of customer i can be taken into account by assigning a high value to $c_{W,i,j}$ if customer i has a high priority, and assigning a low value to $c_{W,i,j}$ if customer i has a low priority. Then, if we want to create a schedule in which customers are scheduled as closely as possible to their preferred time interval, we can define the linear cost function

$$\begin{aligned} f(i,j,t) = {} & c_{W,i,j}[a_i - \Delta(t-1)]\mathbb{1}\left\{t < \frac{a_i}{\Delta} + 1\right\} \\ & + c_{W,i,j}[\Delta(t-1) - b_i + \beta_i]\mathbb{1}\left\{t > \frac{b_i - \beta_i}{\Delta} + 1\right\}. \end{aligned} \tag{4.4}$$

By substituting this cost function into Equation (4.2a), the objective function can be written as

$$\begin{aligned} \text{minimize} \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} c_{W,i,j} & \left(\sum_{t=1}^{\lfloor a_i/\Delta \rfloor + 1} [a_i - \Delta(t-1)] x_{i,j,t} \right. \\ & \left. + \sum_{t=\lceil (b_i - \beta_i)/\Delta \rceil + 1}^{T} [\Delta(t-1) - b_i + \beta_i]\, x_{i,j,t} \right). \end{aligned} \tag{4.5}$$

Second, we consider a *quadratic cost function.* Such a function can be applied when it is desirable to avoid large deviations between the scheduled and preferred time of a customer. For instance, a schedule in which two customers both deviate one time interval from their preferred time interval is preferred above a schedule in which one customer deviates two time intervals from his/her preferred time interval. The quadratic cost function can be defined as

$$\begin{aligned} f(i,j,t) = {} & c_{W,i,j}[a_i - \Delta(t-1)]^2\mathbb{1}\left\{t < \frac{a_i}{\Delta} + 1\right\} \\ & + c_{W,i,j}[\Delta(t-1) - b_i + \beta_i]^2\mathbb{1}\left\{t > \frac{b_i - \beta_i}{\Delta} + 1\right\}. \end{aligned} \tag{4.6}$$

When this cost function is applied, the objective function (4.2a) can be written as

$$\begin{aligned} \text{minimize} \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} c_{W,i,j} & \left(\sum_{t=1}^{\lfloor a_i/\Delta \rfloor + 1} [a_i - \Delta(t-1)]^2 x_{i,j,t} \right. \\ & \left. + \sum_{t=\lceil (b_i - \beta_i)/\Delta \rceil + 1}^{T} [\Delta(t-1) - b_i + \beta_i]^2\, x_{i,j,t} \right). \end{aligned} \tag{4.7}$$

Another factor that affects the customer's priority could be the remaining time before the goods that need to be delivered are actually needed. Suppose that a customer needs to deliver goods which will be shipped by a boat that departs at 15:00. Then it is required to schedule that customer a considerable amount of time before 15:00, in order to keep some room to transport the goods from the truck to the boat, and carry out all necessary checks before shipping. In such a case it may be preferred to schedule that customer early in the morning, and consequently count lower costs for early time intervals. The cost functions as given in Equations (4.4) and (4.6) can easily be extended to cost functions with different costs per time interval. To this end, replace $c_{W,i,j}$ by $c_{W,i,j,t}$, where the latter represents the waiting costs per time unit of scheduling customer i at crossdock j at time interval t. For each time interval t this variable could have a different value, and thus in case of the given example, for that specific customer increasingly high costs could be counted as the time approaches 15:00. In Section 4.2.1 it will be explained how time interval limitations can be added to the model, which can be used to avoid that the considered customer is scheduled after 15:00.

4.2 Extensions and scenarios

In this section several extensions to the model as described in Section 4.1.1 will be presented. These extensions and scenarios are interesting from different points of view. The basis of all of these extensions is the most standard scenario in which customers will be scheduled as closely as possible to their required time interval, while priorities of different types of customers are taken into account. Such a schedule can be obtained by using cost function (4.4) or (4.6), and then solving the corresponding ILP as given in Equation (4.2). Here, no additional constraints are required.

In the following subsections, multiple extensions to the standard model will be described. For each extension, the additional constraints that are required will be given. In Section 4.3 these extensions will be illustrated by means of multiple examples.

4.2.1 Limited allowed time intervals

In practice it may be the case that some customers are not allowed to be scheduled at a certain time period during the day, due to for instance other obligations or appointments. Moreover, in line with the example given at the end of Subsection 4.1.2, it may also be the case that a specific customer is not allowed to be scheduled after say 15:00, due to a ship that leaves around this time with goods that should be delivered before. Hence, it may be useful to add a constraint that limits the set of time intervals at which a customer can be scheduled. Suppose that customer i is not able to be scheduled between time $[t_{1,i}, t_{2,i}]$, where $0 \leq t_{1,i} \leq t_{2,i} \leq T_{\text{end}}$. Then, the set of time intervals at which customer i is not allowed to be scheduled is given by

$$\mathcal{T}_i^{\text{not}} = \left\{ t \in \mathcal{T} : \frac{t_{1,i} - \beta_i}{\Delta} + 1 < t < \frac{t_{2,i}}{\Delta} + 1 \right\}. \tag{4.8}$$

For each customer $i \in \mathcal{N}$ for which such a time interval limitation applies, the following constraint could be added to the model:

$$\sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}_i^{\text{not}}} x_{i,j,t} = 0. \tag{4.9}$$

Note that when a specific customer i has multiple time intervals during which he/she is not allowed to be scheduled, then for each of these time intervals the set $\mathcal{T}_i^{\text{not}}$ as given in Equation (4.8) should be determined, and the corresponding constraint (4.9) has to be added to the model.

4.2.2 Predecessor requirements

A feature with which the model can be extended are requirements regarding the order in which customers should be scheduled. Such requirements are for instance applicable if a certain customer k needs to pick up goods at the distribution center which should be first delivered by customer i. In this case it is required to schedule customer i before customer k, which can be realized by adding the constraint

$$\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} tx_{i,j,t} < \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} tx_{k,j,t}. \tag{4.10}$$

Similarly, if there are multiple customer ordering constraints, then for each requirement in the form 'customer i needs to be scheduled before customer k', constraint (4.10) should be added to the model.

4.2.3 Minimum required interarrival times

In some cases it is desired to plan a certain amount of (empty) time after a customer's service time. For example, some time may be needed to transport all delivered goods to the right location within the distribution center, or some time may be needed to prepare or set up the crossdock for the next customer.

Suppose that it is desired to schedule at least $\Delta_i^{\min}$ time units after customer i. A schedule that takes this extra amount of time into account can be obtained by increasing the time a customer will be 'in service'. Thus, the number of time intervals D_i at which customer i will be in service becomes

$$D_i = \left\lceil \frac{\beta_i + \Delta_i^{\min}}{\Delta} \right\rceil. \tag{4.11}$$

Substituting Equation (4.11) into constraint (4.3c) and solving the resulting ILP gives a schedule in which minimum interarrival times are taken into account.

4.2.4 Maximal interarrival times

From a customer perspective it may be desirable to create a schedule with large interarrival times. When a customer is longer in service than planned, this may affect other customers that are scheduled at a later moment during the day. Hence, in order to minimize the expected waiting time of customers, one may prefer to create a schedule with large interarrival times. Besides, if some time is kept empty at the end of a schedule, the lateness will be reduced. Additionally, the presence of empty time during the day gives the possibility to include high-priority customers that arrive during the day in the schedule without making many mutations.

As mentioned in Chapter 2, empirical studies have shown that the distribution of the loading and unloading times of trucks is right-tailed, see for instance Kiesling and Walton (1995), Franz and Stolletz (2012), The Tioga Group (2013). When customers have different expected service times, then it makes sense to schedule more empty time after customers with a large service times than after customers with a small service times. We illustrate this by an example. Suppose that the service time B_i of customer i is Erlang-2 distributed with mean β_i. Denote with

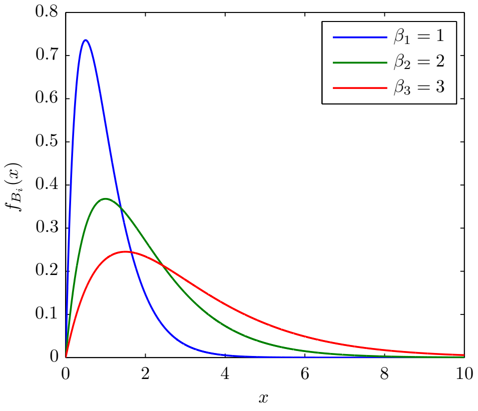


Figure 4.1: Erlang-2 probability density function for different means (β).

$f_{B_i}(\cdot)$ the probability density function (pdf) of the service time of customer i, having parameters $k = 2$ and $\mu_i = k/\beta_i$. Thus,

$$f_{B_i}(x) = \frac{\mu_i^k x^{k-1} e^{-\mu_i x}}{(k-1)!}.$$

Now, consider three different customers, having an average service time of $\beta_1 = 1$, $\beta_2 = 2$, and $\beta_3 = 3$ respectively. In Figure 4.1, for each customer the pdf of the service time is shown. As can be seen from this figure is that the pdf of customer 3 has a much heavier right-tail than the pdf of customer 1. In other words, large jobs are likely to have more delay than small jobs. Hence, in order to minimize the customer's waiting time it makes sense to schedule more empty time after customers with a large service time than after customers with a small service time.

Similarly, denote with $F_{B_i}^{-1}(\cdot)$ the inverse cumulative distribution function (cdf) of the service time of customer i. For some distributions the inverse cdf can be found analytically; the inverse cdf of the Erlang distribution can be found numerically via an iterative procedure. Continuing with the values of the example given in the previous paragraph, we have $F_{B_1}^{-1}(0.8) = 1.497$, $F_{B_2}^{-1}(0.8) = 2.994$, and $F_{B_3}^{-1}(0.8) = 4.491$. In other words, 80% of the customers with an average service time of 1 time unit is served within 1.497 time units, 80% of the customers with an average service time of 2 time units is served within 2.994 time units, whereas 80% of the customers with an average service time of 3 time units is served within 4.491 time units. For all three customers the relative increment $F_{B_i}^{-1}(y)/\beta_i$ is given by 1.497 when $y = 0.8$. This means that for the given values the delay of customers increase linearly with respect to the expected service time. These results were also found for different values for k, y, and β_i. In Figure 4.2 the inverse Erlang-2 cdf is shown for several parameter values.

When the service times B_i are exponentially distributed, which is similar to the Erlang distribution with $k = 1$, then the inverse cdf can be determined analytically. In this case, if $\mu_i = 1/\beta_i$, the cdf is given by

$$F_{B_i}(x) = 1 - e^{-\mu_i x},$$

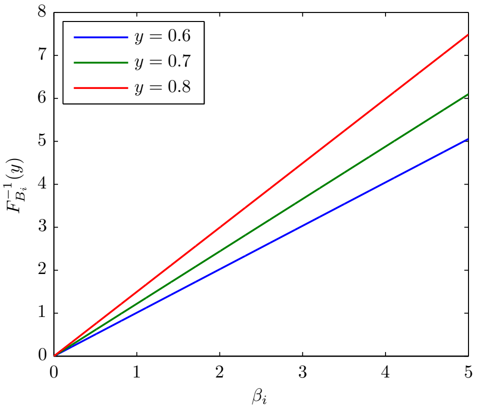


Figure 4.2: Inverse Erlang-2 cdf for different parameters.

and the corresponding inverse cdf equals

$$F_{B_i}^{-1}(x) = -\frac{\ln(1-x)}{\mu_i} = -\beta_i \ln(1-x).$$

The latter function is linear in β_i. Hence, it can be concluded that when the service times are exponentially distributed that the delay of service increases linearly with respect to the expected service time β_i. Since the Erlang-k distribution is equal to the distribution of the sum of k independent identically distributed exponential variables, it seems reasonable that the inverse Erlang cdf is also linear in β_i. After all, the sum of multiple linear functions is again linear. This reasoning corresponds to the numerical results as shown earlier in this subsection.

For several distributions in which the variance increases linearly with respect to the mean, a linear relationship has been found between the average service time and the service time delay. For instance, when the service times are normally distributed with mean β_i and standard deviation $\beta_i/\sqrt{k}$, which is equal to the standard deviation of the Erlang distribution, then for any $\beta_i > 0$ the relative increment $F_{B_i}^{-1}(0.8)/\beta_i$ is given by 1.595. However, this linear relationship does not hold for all parameter combinations and distributions. For example, when the standard deviation used within the normal distribution does not increase linearly with respect to the mean, then the relative increment is not linear. Similarly, when the service times follow a chi-square distribution, then the relative increment is not linear. However, for such distributions the relationship is often near-linear, and thus a linear increment will be a good approximation.

A schedule in which the delay of customers is being reduced can be obtained by scheduling the remaining (empty) time intervals in a proper way between jobs. This can be done by increasing the number of time intervals D_i in which a customer is in service. As shown in the previous paragraphs, in many service time distributions the delay increases linearly with respect to the job size. Hence, we will first describe how under this assumption a schedule with maximal interarrival times could be created. Consider a single-crossdock model, i.e., $M = 1$, and suppose that there are no time interval limitations or predecessor requirements. Then the maximum factor γ with which all service durations can be increased is given by

$$\gamma = \max\left\{\theta : \sum_{i\in\mathcal{N}} \left\lceil \frac{\theta\beta_i}{\Delta} \right\rceil \leq T\right\}. \tag{4.12}$$

The lower and upper bounds for γ are respectively given by

$$\gamma_{\mathrm{LB}} = \frac{T}{\sum_{i\in\mathcal{N}} \left\lceil \frac{\beta_i}{\Delta} \right\rceil},$$

and

$$\gamma_{\mathrm{UB}} = \frac{T}{\sum_{i\in\mathcal{N}} \frac{\beta_i}{\Delta}}.$$

Clearly, when the lower and upper bounds have the same value, then $\gamma = \gamma_{\mathrm{LB}} = \gamma_{\mathrm{UB}}$. If this is not the case, then γ can be found by using a numerical procedure. Once γ is determined, a schedule with maximal interarrival times can be obtained by solving the ILP as given in Equation (4.3), where for all $i \in \mathcal{N}$ the number of time intervals D_i in which customer i is 'in service' is replaced by

$$D_i = \left\lceil \frac{\gamma\beta_i}{\Delta} \right\rceil. \tag{4.13}$$

When there are any time interval limitations or predecessor requirements, or when the multi-crossdock model is applicable, then it is much more complicated to find the maximum factor with which the service times can be increased. In this case γ can be determined via an iterative procedure; an upper bound is given by

$$\gamma_{\mathrm{UB}} = \frac{MT}{\sum_{i\in\mathcal{N}} \frac{\beta_i}{\Delta}}.$$

This upper bound can be used as starting value within the model, by substituting this value in Equation (4.13) and next substituting D_i in constraint (4.2c). Then, γ can be found up to a certain precision by iteratively decreasing and increasing the value for an increasingly smaller step size.

If the delay in service does not increase linearly with respect to the average service time, then the number of time intervals D_i in which customer i is in service can be replaced by

$$D_i = \left\lceil \frac{F_{B_i}^{-1}(p)}{\Delta} \right\rceil, \tag{4.14}$$

where in the single-crossdock model without additional requirements

$$p = \max\left\{y : \sum_{i\in\mathcal{N}} \left\lceil \frac{F_{B_i}^{-1}(y)}{\Delta} \right\rceil \leq T\right\}.$$

When there are additional requirements, or when the multi-crossdock model is applicable, p can be determined via an iterative procedure. Notice that for the implementation the inverse cdf $F_{B_i}^{-1}(\cdot)$ should be known. However, the empirical inverse cdf can also be used within Equation (4.14).

4.2.5 Minimum makespan

A schedule with an early as possible end time can be obtained by minimizing the makespan. Such a schedule will reduce the operating or salary costs throughout the day, since the earliest possible end time is realized. Another application in which it could be preferred to minimize the makespan is when laying asphalt. This is a continuous process which should basically not be interrupted for a large time period in between. The required asphalt is supplied directly from the factory, but cannot be used hours after the production, due to temperature requirements. Hence, in this case it is preferred that the trucks continuously deliver asphalt, with a certain interarrival time. Such a schedule can be achieved by minimizing the makespan.

A schedule with a minimum makespan can be created in two steps. The first step is to determine the earliest time interval T_{earliest} at which the schedule can be finished; the second step is to solve the ILP model with an additional constraint. When there are no time interval limitations or customer predecessor requirements, then for the single-crossdock model we have

$$T_{\text{earliest}} = \sum_{i \in \mathcal{N}} D_i.$$

However, when there are any time interval limitations or customer predecessor requirements, or when the multi-crossdock model is applicable, then T_{earliest} can be obtained by solving an ILP. This ILP is based on (4.2), but additionally a dummy customer, say customer $N+1$, with a service time $\beta_{N+1} = 1$ is added to the model. Additionally, a constraint is added that obliges customer $N+1$ to be scheduled after all other customers. In order to ensure that the dummy customer can be scheduled, we add an extra time interval. Hence, define $\mathcal{N}_{\text{new}} := \{1, \ldots, N+1\}$ and $\mathcal{T}_{\text{new}} := \{1, \ldots, T+1\}$. Then, the earliest possible time interval T_{earliest} at which the schedule can be finished is the solution of the following ILP:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}_{\text{new}}} (t-1) x_{N+1,j,t} && && \text{(4.15a)}\\
\text{subject to} \quad & \sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}_{\text{new}}} t x_{i,j,t} < \sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}_{\text{new}}} t x_{N+1,j,t}, && i \in \mathcal{N}, && \text{(4.15b)}\\
& \sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}_i^{\text{end}}} x_{i,j,t} = 0, && i \in \mathcal{N}, && \text{(4.15c)}\\
& \sum_{t \in \mathcal{T}_{\text{new}}} x_{i,j,t} \le d_{i,j}, && i \in \mathcal{N},\ j \in \mathcal{M}, && \text{(4.15d)}\\
& \text{(4.2b)} - \text{(4.2d)},\ \text{(4.2g)}. && && \text{(4.15e)}
\end{aligned}$$

Here, in constraint (4.15e) the sets $\mathcal{N}$ and $\mathcal{T}$ are replaced by $\mathcal{N}_{\text{new}}$ and $\mathcal{T}_{\text{new}}$ respectively. Additionally, when there are any time interval limitations or requirements concerning the order in which customers are served, then these constraints (see Sections 4.2.1 and 4.2.2) can be added to the ILP.

Once the earliest possible end time is determined, the following constraint can be added to (4.2) in order to create a schedule with a minimum makespan:

$$\sum_{j \in \mathcal{M}} \sum_{t \in \mathcal{T}} (t + D_i - 1) x_{i,j,t} \le T_{\text{earliest}}, \quad i \in \mathcal{N}. \tag{4.16}$$

In the multi-crossdock model the solution of the corresponding ILP is one with an overall minimum makespan. Then, when preferred, for each machine the makespan can be further minimized by applying the same model on each machine separately. Thus, the set of jobs that should be scheduled at each machine is determined by the overall model; subsequently for each machine T_{earliest} is determined and a single-crossdock schedule is created. Of course, other combinations of reruns are also possible. Given that the schedule finishes at the overall earliest end time interval T_{earliest}, one could for instance also create a schedule with maximal interarrival times by applying the model given in Subsection 4.2.4, whereas constraint (4.16) is added to (4.2), or where the number of time intervals for which the schedule should be made T is replaced by T_{earliest}.

4.2.6 Minimum idle time

In this subsection we will describe how a schedule with a minimum idle time can be created. The idle time of a schedule is given by the time in which no customers are served while the schedule is in progress. Such a schedule could be useful in the same situations as that of a schedule with a minimal makespan might be used. However, in addition to the latter type of schedule, it is not required that a schedule with a minimum idle time starts at time 0. Therefore, the idle time within a schedule with a minimum idle time is always less than or equal to the idle time of a schedule with a minimum makespan. The advantage of a schedule with a minimum idle time is that customers are basically scheduled closer to their preferred time interval; i.e., the schedule has lower costs.

A schedule with a minimum idle time can be obtained in two steps. The first step is to determine the smallest number of time intervals T_{shortest} in which the schedule can be finished; the second step is to create a schedule by solving the ILP model with an additional constraint. When there are no time interval limitations or customer predecessor requirements (see Sections 4.2.1 and 4.2.2), then in case of the single-crossdock model we have

$$T_{\text{shortest}} = \sum_{i \in \mathcal{N}} D_i .$$

When there are any time interval limitations or customer predecessor requirements, or when the multi-crossdock model is applicable, then T_{shortest} can be determined by solving an ILP. This ILP is based on (4.2), but additionally two 'dummy' customers are added: one that is required to be scheduled before all other customers – say customer 0 – and one that is required to be scheduled after all other customers – say customer $N+1$. Both customers have a service time of $\beta_i = 1$ time unit. Furthermore, in order to be sure that both customers can be scheduled, we add both at the beginning and end of the schedule a time interval. Hence, define $\mathcal{N}_{\text{new}} = \{0, \ldots, N+1\}$ and $\mathcal{T}_{\text{new}} = \{0, \ldots, T+1\}$. Then, T_{shortest} is the solution of the following ILP:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}}(t-1)x_{N+1,j,t} - \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}} tx_{0,t} && && \text{(4.17a)}\\
\text{subject to} \quad & \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}} tx_{0,j,t} < \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}} tx_{i,j,t}, && i\in\mathcal{N}, && \text{(4.17b)}\\
& \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}} tx_{i,j,t} < \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_{\text{new}}} tx_{N+1,j,t}, && i\in\mathcal{N}, && \text{(4.17c)}\\
& \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}_i^{\text{end}}\cup 0} x_{i,j,t} = 0, && i\in\mathcal{N}, && \text{(4.17d)}\\
& \sum_{t\in\mathcal{T}_{\text{new}}} x_{i,j,t} \le d_{i,j}, && i\in\mathcal{N},\ j\in\mathcal{M}, && \text{(4.17e)}\\
& \text{(4.2b)} - \text{(4.2d)},\ \text{(4.2g)}. && && \text{(4.17f)}
\end{aligned}$$

Here, in constraint (4.17f) the sets $\mathcal{N}$ and $\mathcal{T}$ are replaced by $\mathcal{N}_{\text{new}}$ and $\mathcal{T}_{\text{new}}$ respectively. As holds for the schedule with minimum makespan, additional constraints arising from for instance time interval limitations or customer predecessors can be added to the ILP.

Once the smallest number of time intervals T_{shortest} in which all customer could be served is determined, a schedule with a minimum idle time could be created by adding the following constraint to (4.2):

$$\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}}(t+D_k-1)x_{k,j,t} - \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}}(t-1)x_{i,j,t} \le T_{\text{shortest}}, \quad i,k\in\mathcal{N} : i<k.$$

For the multi-crossdock model similar extensions are possible as for the schedule with a minimum makespan, see Subsection 4.2.5.

4.3 Examples

In this section the model with extensions as described in the previous sections will be illustrated. For simplicity, and in order to keep the notation and the number of presented schedules limited, we present examples of single-crossdock systems (i.e., $M = 1$). The legend corresponding to all schedules that will be given is shown in Figure 4.3. First in Subsection 4.3.1 different cost functions will be illustrated, whereas in Subsection 4.3.2 examples will be given of the extensions of the model as described in Section 4.2.

4.3.1 Different cost functions

In this example the difference between a linear and quadratic cost function will be illustrated. Consider a time period of $T = 10$ intervals, each of length $\Delta = 1$ quarter. Suppose that $N = 5$ customers need to be scheduled, each having a service time of $\beta_i = 2$ quarters. Let the preferred time intervals for customer 1 until N be given by $[0, 2]$, $[6, 8]$, $[4, 6]$, $[6, 8]$, $[8, 10]$ respectively, and let all customers have the same priority. Thus, $c_{W,i} = 1$, for all $i \in \mathcal{N}$. Solving the standard single-crossdock ILP, see Equation 4.2, with linear objective function (4.4) results in the schedule given

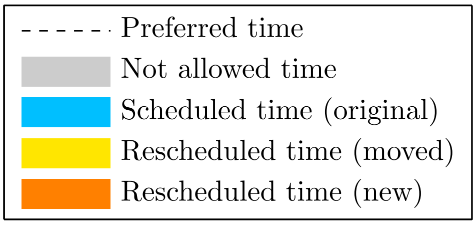
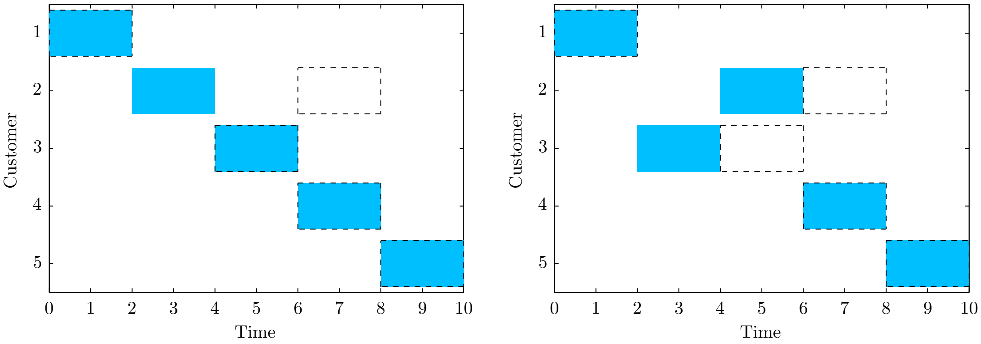


Figure 4.3: Legend belonging to the schedules.



(a) Linear cost function.

(b) Quadratic cost function.

Figure 4.4: Schedules illustrating different types of cost functions.

in Figure 4.4(a). The schedule corresponding with the quadratic cost function (4.6) is given in Figure 4.4(b).

From Figure 4.4 it can be seen that in the schedule in which the linear cost function is used just one customer is not scheduled within his/her preferred time interval. For this customer the difference between the scheduled and preferred time is 4 time intervals, which corresponds to an objective value of 4 for the whole schedule. In the schedule in which the quadratic cost function is used two customers are not scheduled within their preferred time interval. For each of these customers the difference between the scheduled and preferred time is 2 time intervals, which corresponds to an objective value of $2^2 + 2^2 = 8$ for the whole schedule. Notice that in the latter case the model will never end up with the schedule corresponding with linear cost function as given in Figure 4.4(a), since the costs of this schedule are $4^2 = 16$ in case of a quadratic cost function. Hence, a quadratic cost function avoids large deviations between the scheduled and preferred time of a customers.

4.3.2 Extensions and scenarios

In this subsection we will illustrate the extensions as given in Section 4.2. To this end, we first create a standard schedule, and then add multiple extensions. Consider a time period of $T = 10$ intervals, each of length $\Delta = 1$ quarter. Suppose that $N = 5$ customers need to be scheduled, each having a service time of $\beta_i = 1$ quarter, for all $i \in \mathcal{N}$. Let the preferred time intervals of customer 1 until N be given by $[0, 3]$, $[1, 6]$, $[2, 5]$, $[7, 10]$, and $[8, 10]$ respectively, and let all customers have the same priority. Thus, $c_{W,i} = 1$, for all $i \in \mathcal{N}$. Applying linear cost function (4.4) and solving the corresponding ILP results in the schedule shown in Figure 4.5(a). In the remaining part of this subsection this schedule will be called the 'standard schedule'.

Limited allowed time intervals Consider the standard schedule, but now suppose that customer 2 is not able to be scheduled between time $[2, 10]$, due to external reasons. Adding this restriction to the model and solving the corresponding ILP results in the schedule shown in Figure 4.5(b).

Predecessor requirements Consider the standard schedule, but now require that customer 3 can only be served if customer 4 is served before. Thus, customer 3 has predecessor 4. Adding this constraint and solving the corresponding ILP results in the schedule given in Figure 4.5(c).

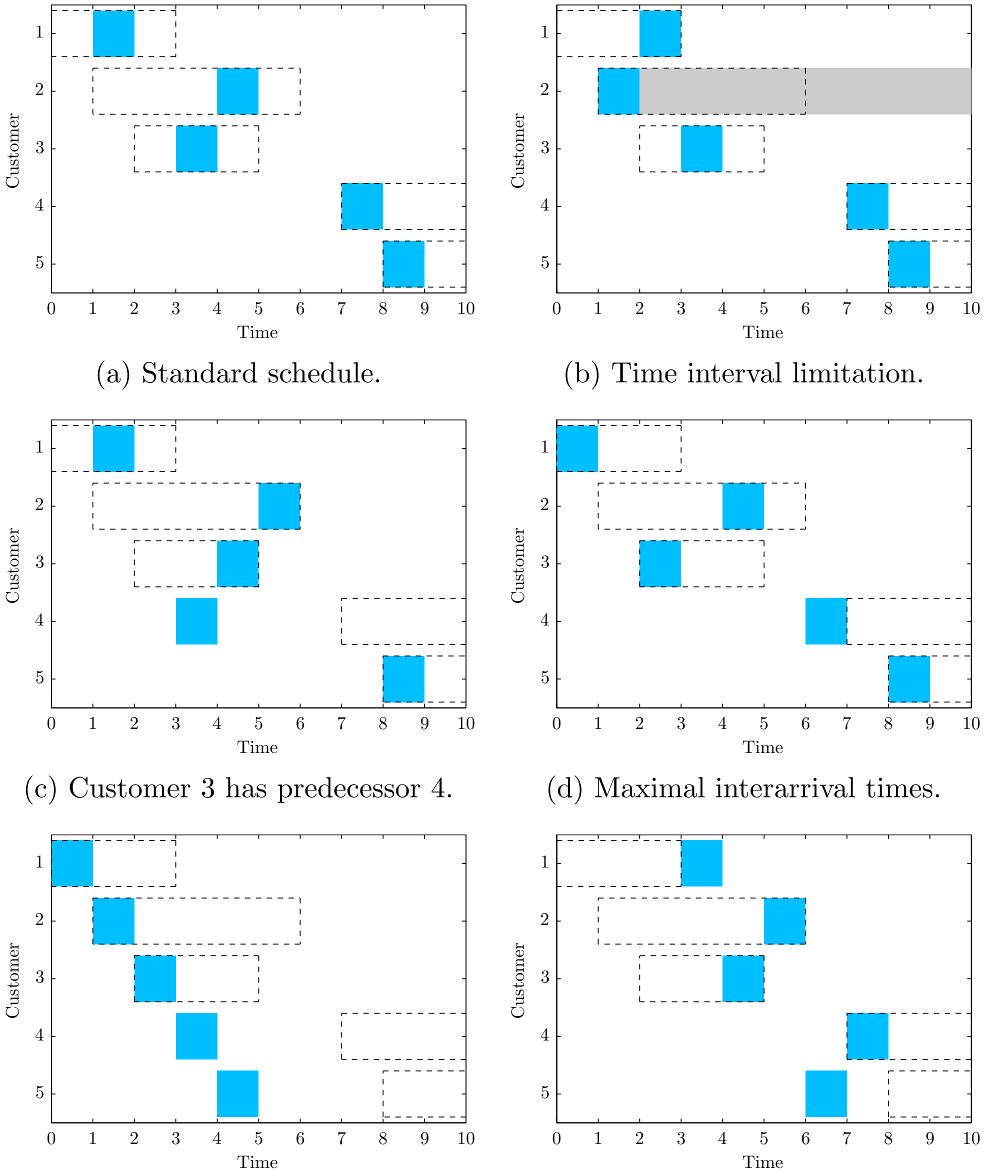
Maximal interarrival times Consider the standard schedule, but now require that each customer has an as large as possible interarrival time. The maximum factor with which the service times can be increased is given by $\gamma = 2$; the resulting schedule is shown in Figure 4.5(d).

Minimum makespan Consider the standard schedule, but now require that the makespan of the schedule is minimized. The earliest time interval at which the schedule can be finished is given by $T_{\text{earliest}} = 5$. A schedule with a minimal makespan, whereas all customers are scheduled as closely as possible to their preferred time interval is shown in Figure 4.5(e).

Minimum idle time Consider the standard schedule, but now suppose that we want to create a schedule with a minimum idle time. The smallest number of time intervals in which the schedule can be finished is given by $T_{\text{shortest}} = 5$. Adding this constraint and solving the ILP results in the schedule as given in Figure 4.5(f). Notice that the costs of this schedule is much lower than the costs of the schedule with a minimum makespan.

4.3.3 Different service times

In Section 4.2.4 we have discussed how a schedule should be created with maximal interarrival times when customers have different expected service times. Here we will give an example of the model in which it is assumed that the delay of service time increases linearly with respect to the expected service time. Consider a time period of $T = 10$ intervals, each of length $\Delta = 1$ quarter. Suppose that $N = 5$ customers need to be scheduled, where the expected service times of the customers are given by $\beta_1 = \beta_2 = 2$ quarters, and $\beta_3 = \beta_4 = \beta_5 = 1$ quarter. Let the preferred time intervals for customer 1, 2, and 3 be given by $[0, 5]$, and let the preferred time interval for customers 4 and 5 be $[10, 12]$. Assume that all customers have the same priority, i.e., $c_{W,i} = 1$, for all $i \in \mathcal{N}$. Solving the single-crossdock ILP with linear objective function (4.4) and D_i defined as in Equation (4.13), results in the schedule given in Figure 4.6(a). However, if we now assume that customer 1 has a high-priority, i.e., $c_{W,1} = 3$, then one will obtain the schedule as given in Figure 4.6(b).



(a) Standard schedule.

(b) Time interval limitation.

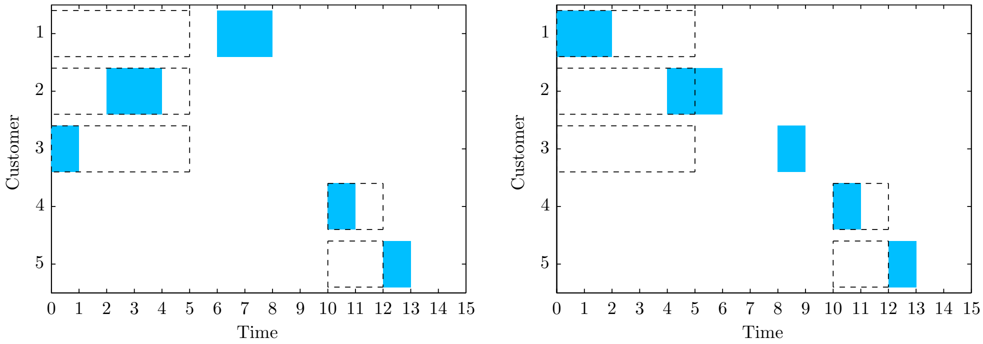
(c) Customer 3 has predecessor 4.

(d) Maximal interarrival times.

(e) Minimum makespan.

(f) Minimum idle time.

Figure 4.5: Schedules illustrating different requirements.



(a) Equal priorities.

(b) Customers 1 and 2 have a high priority.

Figure 4.6: Schedules illustrating customers with different service times and priorities.

It can be seen that in the latter schedule the high-priority customer is scheduled within his/her preferred time interval.

4.4 Rescheduling during the day

In this section we will discuss how unscheduled high-priority customers that arrive during the day can be added to a schedule that is already in process. If an important customer arrives, it may not be desirable to let this customer wait until the end of the day before he/she is being served. Specifically, when there is not enough room in the schedule to serve the newly arrived customer within a reasonable amount of time, it may be preferred to move some already scheduled to a later time interval. Depending on the priority of the (un)scheduled customers, the newly arrived customer should wait for a short or long time.

The remaining part of this section is organized as follows. First, in Subsection 4.4.1 we will describe the model that can be used for rescheduling during the day. Next, in Subsection 4.4.2 the cost function will be explained in more detail.

4.4.1 Model description

Consider the model as described in Section 4.1, but now suppose that during the day some new customers arrive. Denote with $\mathcal{N}_{\text{new}}$ the complete set of customers that should be served, consisting of scheduled and newly arrived customers. Suppose that we want to make a new schedule from time interval $s \in \mathcal{T}$ on. Seen from a customer-friendly perspective, it is preferred to create a schedule in which the movements of already scheduled customers to other time intervals is minimized. In general, it will be experienced as annoying by customers to move to another time interval if they were already assigned to a time interval. Moreover, for some customers it may even not be possible to go in service at an earlier moment during the day, due to travel time or other appointments.

Define $\mathcal{T}_s = \{1, \ldots, s-1\}$ as the set of time intervals for which the schedule cannot be changed anymore. Furthermore, let $g(i, j, t)$ be a general function indicating the costs of scheduling customer i at crossdock j at time interval t. One of the most important factors that may influence the cost function is the initial schedule. In Subsection 4.4.2 this cost function will be explained in more detail. Let $x^{\text{old}}_{i,j,t}$ be the value of $x_{i,j,t}$ within the original schedule, i.e., $x^{\text{old}}_{i,j,t} = 1$ if customer i was originally scheduled at crossdock j at time interval t; $x^{\text{old}}_{i,j,t} = 0$ otherwise. Then a new schedule in which all newly arrived customers are included while the costs are minimized can be obtained by solving the following ILP:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in\mathcal{N}_{\text{new}}}\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} g(i,j,t)x_{i,j,t} && && \text{(4.18a)}\\
\text{subject to} \quad & x_{i,j,t} = x^{\text{old}}_{i,j,t}, && i\in\mathcal{N},\ j\in\mathcal{M},\ t\in\mathcal{T}_s, && \text{(4.18b)}\\
& \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} t x_{i,j,t} \geq s, && i\in\mathcal{N}_{\text{new}}\setminus\mathcal{N}, && \text{(4.18c)}\\
& \text{(4.2b)} - \text{(4.2g)}. && && \text{(4.18d)}
\end{aligned}$$

Here in constraint (4.18d) the set $\mathcal{N}$ is replaced by $\mathcal{N}_{\text{new}}$. The extensions that are described in Section 4.2 can be added to this model in a similar way as they are added to (4.2). When it is desired that already scheduled customers are not scheduled at an earlier time interval, then the following constraint can be added to (4.18):

$$\sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} t x^{\text{old}}_{i,j,t} \leq \sum_{j\in\mathcal{M}}\sum_{t\in\mathcal{T}} t x_{i,j,t}, \quad i\in\mathcal{N}. \tag{4.19}$$

4.4.2 Cost function

The function $g(i,j,t)$ as given in (4.18a) is a general function indicating the costs of (re)scheduling customer i at crossdock j at time interval t. As mentioned above, from a customer-friendly perspective it may be desirable to create a new schedule in such a way that the movements of already scheduled customers is minimized. Denote with t^{old}_i the originally scheduled time interval of customer i, and let α_R be the weight of rescheduling. For instance, when considering two equal-priority customers, $\alpha_R = 2$ means that the costs of moving an already scheduled customer one time unit later is two times as high as the costs of scheduling a newly arrived customer one time unit after his/her preferred time interval.

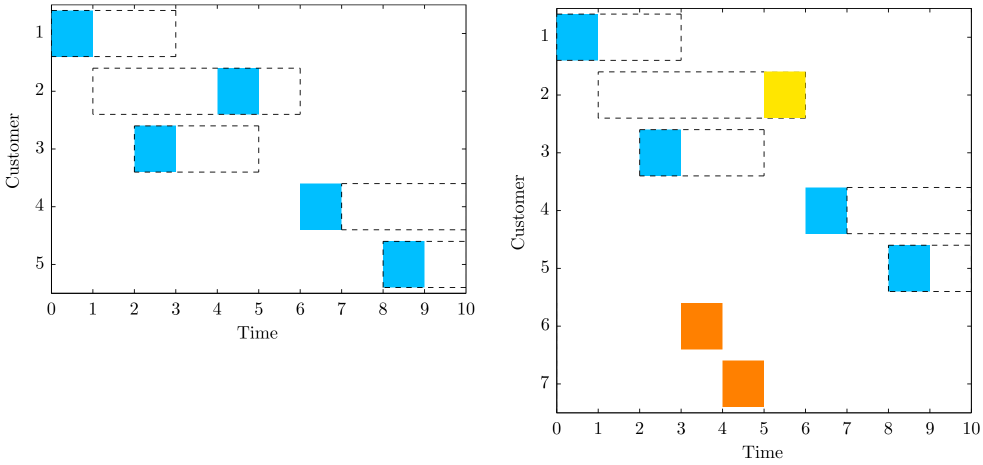
A linear cost function that can be used to minimize the movements of customers while taking the waiting time of the new customer into account can be given by

$$g(i,j,t) = \begin{cases} \alpha_R c_{W,i,j}\Delta \left|t - t^{\text{old}}_i\right|, & i\in\mathcal{N},\ j\in\mathcal{M},\\ c_{W,i,j}\Delta|t-s|, & i\in\mathcal{N}_{\text{new}}\setminus\mathcal{N},\ j\in\mathcal{M}. \end{cases} \tag{4.20}$$

Similarly, one could define the quadratic cost function

$$g(i,j,t) = \begin{cases} \alpha_R c_{W,i,j}\Delta \left|t - t^{\text{old}}_i\right|^2, & i\in\mathcal{N},\ j\in\mathcal{M},\\ c_{W,i,j}\Delta|t-s|^2, & i\in\mathcal{N}_{\text{new}}\setminus\mathcal{N},\ j\in\mathcal{M}. \end{cases} \tag{4.21}$$

Furthermore, preferences with respect to the time interval in which customers will be scheduled could be taken into account by replacing $c_{W,i,j}$ by $c_{W,i,j,t}$, where the latter represents the waiting costs per time unit of scheduling customer i at crossdock j during time interval t. Then, for each time interval t different costs could be counted.



(a) Original schedule.

(b) New schedule with two high-priority customers.

Figure 4.7: Schedules illustrating the incorporation of new arrivals during the day.

4.5 Example

We will illustrate the rescheduling model as given in the previous section by an example. Consider the standard schedule with maximum interarrival times, as described in Subsection 4.2. For completeness, this schedule is given in Figure 4.7(a). Now suppose that two new high-priority customers arrive at time 3. Let the waiting costs of both customers be $c_{W,i} = 3$. Then solving (4.18) with cost function (4.20) with moving weight $\alpha_R = 2$ results in the schedule as shown in Figure 4.7(b). From this figure it can be seen that one originally scheduled customer is moved to another time interval; the two newly arrived customers are scheduled somewhere in between.

Chapter 5

Performance Measures

Several different measures exist that can be used to indicate the performance of a schedule or algorithm. In this chapter some performance measures will be given; these can be used for both the schedules that are created via the appointment scheduling model (see Chapter 3) and the schedules that are created via the job shop scheduling model (see Chapter 4). Since the appointment scheduling model and job shop scheduling model differ in many different ways, we will do not present a comparison between both models. Doing so will lead to an unequal, distorted comparison. For instance, within the job shop scheduling model many different extensions can be taken into account, which cannot be included within the appointment scheduling model.

Next to the differences between the models, there are many different objectives that cannot all be quantified clearly, often due to a lack of practical and empirical evidence. Below we list some of these objectives and consequently performance measures. Most of them will be described in more detail in the subsequent sections in this chapter.

- The expected waiting time of customers. This includes both the expected waiting time of scheduled and unscheduled customers. For the latter type of customers it should be known in which way they are served: as soon as possible, within a certain time range, at the end of the day, or something else. By defining a service discipline and/or service level, the expected service time of both scheduled and unscheduled customers can be quantified. In Section 5.1 an explanation will be given of how the expected waiting time of a known schedule can be determined.

- The expected lateness of a schedule. Probably it is not desirable that still a lot of work needs to be done at the planned end time of the schedule. Lateness can be avoided by keeping some time intervals empty at the end of the day. In Section 5.2 the lateness of a schedule will be discussed.

- The expected idle time of crossdocks. Sometimes it is preferred to minimize the idle time of crossdocks. This can be realized by scheduling all customers as closely as possible after each other. In Section 5.2 calculations for the expected idle time of a schedule will be given.

- The total costs of a schedule. This includes the deviation between the preferred and scheduled time of customers, different priorities of customers, and all other preferences and restrictions. Once a schedule is made, the costs can be calculated via the formula as given in Equation (4.4) or (4.6). When the job shop scheduling model is applied, the costs of the schedule equals the objective value of the ILP. The costs of a schedule strongly depends on the used settings and input parameters, such as the preferred time intervals, priorities, etc.

- Run time of the algorithm and scalability. Depending on the way in which an algorithm is used, it is preferred that it comes with a solution within a reasonable amount of time. Regularly, the run time of an algorithm increases heavily as the scale at which it is applied increases. In Section 5.3 more information about this topic can be found.

- The robustness and extensibility of the model. Regularly, a solution is preferred that is insensitive to small changes and which can be extended easily with additional requirements or preferences. For more information about this subject we refer to Section 5.4.

Most of the objectives as described above are conflicting with each other. For instance, it is contradictory to minimize both the waiting time of customers and the idle time of the crossdock at the same. Particularly, a schedule with minimal waiting times will have large interarrival times, whereas a schedule with a minimal idle time will have small interarrival times. Similarly, when the costs of the schedule are minimized, then customers may have large waiting times. Hence, depending on the preferences of the user of the scheduling algorithm a tradeoff can be made between the objectives. Moreover, one could optimize a (selected) combination of objectives.

5.1 Expected waiting time

As introduced earlier, schedule x is be defined by the values $x_{i,t}$, for $i \in \mathcal{N}$ and $t \in \mathcal{T}$. We have $x = (n_1, \ldots, n_T)$, where $n_t = \sum_{i \in \mathcal{N}} x_{i,t}$. Now, let $B_{i,t}$ and $W_{i,t}$ be random variables representing the service time and waiting time of the ith scheduled customer at time interval t respectively. Denote with $f_{B_{i,t}}(\cdot)$ and $f_{W_{i,t}}(\cdot)$ the corresponding probability density functions. Without loss of generality we assume that $n_1 > 0$. If there are no customers scheduled at the first time interval of schedule x, then ignore all the empty time intervals at the beginning of the schedule and let the schedule start at the first time interval for which it holds that $n_t > 0$. Additionally, define for $t > 1$,

$$\bar{t} = \underset{s \in \{1,\ldots,t-1\}}{\arg\max} \{n_s > 0 | n_1 > 0\},$$

which can be interpreted as the last time interval before t at which any customer is scheduled.

In this section we consider a continuous service and waiting time distribution. Nevertheless, similar calculations for a discrete time service and waiting time distribution are given in Appendix A. The probability density function of the waiting time $f_{W_{i,t}}(\cdot)$ can be defined as follows. The first scheduled customer has no waiting time. Hence,

$$f_{W_{1,1}}(0) = 1,$$

and for $y > 0$,

$$f_{W_{1,1}}(y) = 0.$$

All subsequent customers do have a positive waiting time probability density. The probability density of the event that the first scheduled customer at any time interval $t \in \{2, \ldots, T\}$ has no waiting time is given by

$$\begin{aligned} f_{W_{1,t}}(0) &= F_{W_{n_{\bar{t}},\bar{t}}+B_{n_{\bar{t}},\bar{t}}}(\Delta[t-\bar{t}]) \\ &= \int_0^{\Delta[t-\bar{t}]} f_{W_{n_{\bar{t}},\bar{t}}+B_{n_{\bar{t}},\bar{t}}}(y)\,\mathrm{d}y \\ &= \int_0^{\Delta[t-\bar{t}]} \left[f_{W_{n_{\bar{t}},\bar{t}}}(0) f_{B_{n_{\bar{t}},\bar{t}}}(y) + \int_0^y f_{W_{n_{\bar{t}},\bar{t}}}(z) f_{B_{n_{\bar{t}},\bar{t}}}(y-z)\,\mathrm{d}z \right] \mathrm{d}y. \end{aligned}$$

For $t \in \{2, \ldots, T\}$ and $y > 0$ we have

$$\begin{aligned} f_{W_{1,t}}(y) &= f_{W_{n_{\bar{t}},\bar{t}}+B_{n_{\bar{t}},\bar{t}}}(\Delta[t-\bar{t}]+y) \\ &= f_{W_{n_{\bar{t}},\bar{t}}}(0) f_{B_{n_{\bar{t}},\bar{t}}}(\Delta[t-\bar{t}]+y) \\ &\quad + \int_0^{\Delta[t-\bar{t}]+y} f_{W_{n_{\bar{t}},\bar{t}}}(z) f_{B_{n_{\bar{t}},\bar{t}}}(\Delta[t-\bar{t}]+y-z)\,\mathrm{d}z. \end{aligned}$$

For any customer $i \in \{2, \ldots, n_t\}$ the waiting time pdf depends on the waiting time and service time of the previously scheduled customer at that time interval. Thus, for $y \geq 0$,

$$\begin{aligned} f_{W_{i,t}}(y) &= f_{W_{i-1,t}+B_{i-1,t}}(y) \\ &= f_{W_{i-1,t}}(0) f_{B_{i-1,t}}(y) + \int_0^y f_{W_{i-1,t}}(z) f_{B_{i-1,t}}(y-z)\,\mathrm{d}z. \end{aligned}$$

By using the probability density function $f_{W_{i,t}}(\cdot)$ as defined above, the expected waiting time of the ith customer scheduled at time interval t is given by

$$\mathbb{E}W_{i,t} = \int_0^\infty y f_{W_{i,t}}(y)\,\mathrm{d}y. \tag{5.1}$$

Then, the expected waiting time $W(x)$ of schedule x is defined by the sum of all these expectations. Thus,

$$\begin{aligned} W(x) &= \sum_{t=1}^{T} \sum_{i=1}^{n_t} \mathbb{E}W_{i,t} \\ &= \sum_{t=1}^{T} \sum_{i=1}^{n_t} \int_0^\infty y f_{W_{i,t}}(y)\,\mathrm{d}y. \end{aligned} \tag{5.2}$$

5.2 Expected lateness and idle time

The expected lateness $L(x)$ of schedule x corresponds to the expected amount of work that is present at the crossdock at the end of time interval T. This can be determined by scheduling a fictitious customer at time interval $T+1$, and then determining his/her expected waiting time. Hence,

$$\begin{aligned} L(x) &= \mathbb{E}W_{1,T+1} \\ &= \int_0^\infty y f_{W_{1,T+1}}(y)\,\mathrm{d}y. \end{aligned}$$

The idle time $I(x)$ of schedule x is given by the time in which no customers are served at the crossdock while the schedule is still in progress. Therefore, the idle time is given by the duration of the schedule minus the total service time of customers. Hence, if $\mathcal{N}_{\text{new}}$ represents the complete set of customers that should be served, then

$$I(x) = \Delta T + L(x) - \sum_{i \in \mathcal{N}_{\text{new}}} \beta_i.$$

5.3 Run time and scalability

Depending on the way in which a model or algorithm is used, it is preferred that is comes to a solution within a reasonable amount of time. This is especially important for algorithms that are used real-time. Additionally, even though when the model is applied on a larger scale, the run time should be reasonable.

There is a quite large difference in the run time of both the appointment scheduling model and the job shop scheduling model. Within the appointment scheduling model a local search procedure is used, whose search space increases exponentially as the number of time intervals increases. Roughly, it took already one hour to create a schedule with $T = 30$ time intervals and $N = 10$ customers. Even though when the appointment scheduling model can be extended with different service times, multiple crossdocks, and all other extensions, then the search space becomes so large that the run time becomes much more longer than one hour. Additionally, even though when these extensions can be added, there are many different combinations of input parameters and restrictions; consequently the algorithm should be re-executed each time that a new schedule should be made. Especially when the algorithm is used for rescheduling during the day, then such long run times are not acceptable. For comparison, the job shop scheduling model – in which the many extensions and preferences can be taken into account – finds usually in less than one second a schedule with $T = 30$ time intervals and $N = 10$ customers.

5.4 Robustness and extensibility

Generally, a model is preferred that is insensitive for small changes and which can be extended easily with some features. In our case, especially the extensibility of the model is important. From practice there may arise additional requirements or preferences which are desirable to take into account. Clearly, the job shop scheduling model can be extended much more easily than the appointment scheduling model.

The job shop scheduling model can be extended or changed by defining a different cost function or by adding an extra constraint to the ILP. Unfortunately, such extensions or changes cannot be easily added to the appointment scheduling model. The cost function that is used within this model should be multimodular; otherwise the algorithm is not ensured to converge to an optimum. This property cannot be easily proven. Even though when the cost function is multimodular, the proof covers multiple pages.

5.5 Examples

In this section we will give two different examples. First, in Subsection 5.5.1 the influence of different service times on the waiting time of customers is illustrated. Second, in Subsection 5.5.2 differences between the solution of the appointment scheduling algorithm and job shop scheduling model will be presented via an example.

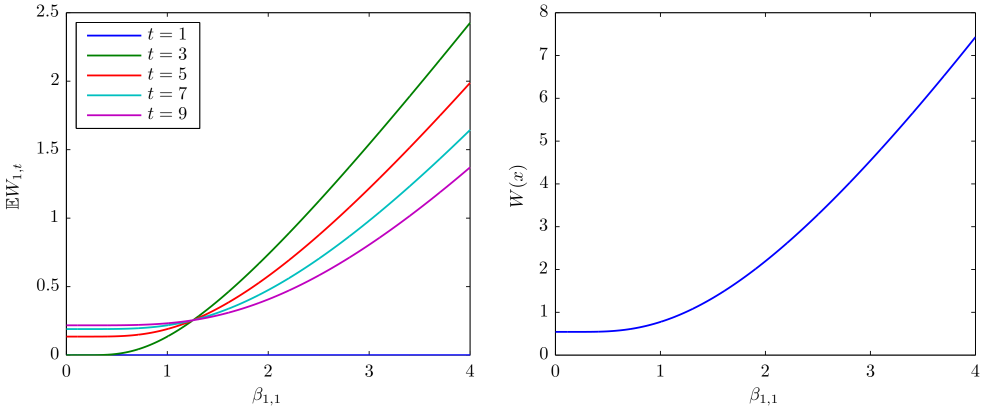
5.5.1 Influence of different service times

It has been found that for many different parameter values the appointment scheduling algorithm turns out in an evenly-spaced schedule. For instance, suppose we want to schedule $N = 5$ customers at one single crossdock during $T = 10$ time intervals, each of length $\Delta = 1$ quarter. Let the service times of all customers be exponentially distributed with an average of $\beta_i = 1$ quarter, for all $i \in \mathcal{N}$. Furthermore, let $\alpha_W = 3$, $\alpha_I = 0$, and $\alpha_L = 1$. Then the resulting appointment schedule is given by $x = (1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$.

However, when the first scheduled customer does not have an average service time of 1 quarter, but higher, then it can be seen that this has a high impact on the expected waiting time of subsequent customers. Denote with $\beta_{1,1}$ the average service time of the first customer. Then, for each customer the expected waiting time $\mathbb{E}W_{i,t}$ can be calculated according to Equation (5.1). For different values of $\beta_{1,1}$ these expectations are shown in Figure 5.1(a). The expected waiting time of the complete schedule, calculated according to Equation (5.2), is shown in Figure 5.1(b). From these figures it can be seen that the average service time of the first customer has a large impact on the waiting time of all subsequent customers. Despite the empty time intervals between customers, even the last scheduled customer can expect a considerably higher waiting time as the service time of the first scheduled customer increases.

5.5.2 Comparison of solutions

We will illustrate the differences between the solution of the appointment scheduling algorithm and job shop scheduling model by an example. Suppose that $N = 4$ customers need to be scheduled at one single crossdock during $T = 10$ time intervals, each of length $\Delta = 1$ quarter. Suppose there is one high-priority customer having an average service time of $\beta_{1,1} = 2$ quarters who prefers to be served at the beginning of the schedule. Let the average service times of all other customers be 1 quarter, and assume that the service times are exponentially distributed. Solving the appointment scheduling model with $\alpha_W = 3$, $\alpha_I = 0$, and $\alpha_L = 1$ gives schedule



(a) Expected waiting times $\mathbb{E}W_{1,t}$ for each customer scheduled at time interval t.

(b) Expected waiting time $W(x)$ of schedule x.

Figure 5.1: Expected waiting times of schedule $x = (1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$ for different values of $\beta_{1,1}$, the average service time of the customer scheduled at time interval $t = 1$.

$x = (1, 0, 1, 0, 0, 1, 0, 0, 1, 0)$, where the high-priority customer is scheduled at time interval $t = 1$. Calculating for each customer the expected waiting time by means of Equation (5.1) gives $\mathbb{E}W_{1,1} = 0$, $\mathbb{E}W_{1,3} = 0.736$, $\mathbb{E}W_{1,6} = 0.341$, and $\mathbb{E}W_{1,9} = 0.183$. The expected waiting time of the complete schedule is given by $W(x) = 1.260$ and the expected lateness equals $L(x) = 0.238$.

Next, we apply the job shop scheduling model with same parameter values, whereas we require that customers have large interarrival times. Solving the model gives $x = (1, 0, 0, 0, 1, 0, 1, 0, 1, 0)$, where the high-priority customer is scheduled at time interval $t = 1$. Calculating for each customer the expected waiting time according to Equation (5.1) gives $\mathbb{E}W_{1,1} = 0$, $\mathbb{E}W_{1,5} = 0.271$, $\mathbb{E}W_{1,7} = 0.298$, and $\mathbb{E}W_{1,9} = 0.295$. The expected waiting time of the complete schedule is given by $W(x) = 0.864$ and the expected lateness equals $L(x) = 0.287$. It can be seen is that the expected waiting time of the job shop schedule is much lower that those of the appointment schedule, whereas the lateness is just slightly higher.

Chapter 6

Routing Policies

Many distribution centers have to deal with a limited number of parking places. Due to these capacity limitations, truck traffic regularly arises on the access routes towards the distribution center. As mentioned in Chapter 1, the truck traffic – and consequently the waiting time of truck drivers and the amount of air pollution – will be reduced in two ways. First, the waiting time of truck drivers will be minimized by scheduling the trucks at the distribution center (see Chapters 3 and 4). When truck drivers know in advance at which time they will be served at the distribution center, then they can plan at which time they should depart such that they arrive on time, of course with a reasonable travel time margin. However, in practice not all customers will be scheduled in advance, and additionally, there might be customers that are scheduled, but who arrive a long time in advance. This may for instance be truck drivers that come from abroad, and who prefer to take a break before continuing work.

The efficient use of the available parking places at the distribution center will not only improve the flow of trucks, but it will also result in a well compliance of the schedule. By using the parking places at the distribution center just for customers that are scheduled within a short amount of time, customers are likely to be on time at the distribution center for their scheduled time interval. In this chapter we investigate how (temporary) parking areas near the distribution center can be used in an efficient way. Here, the aim is to find a routing policy that minimizes a certain cost function. This cost function can for instance be based on the travel time or travel distance that is needed for each type of arrivals to move to a parking area. When the travel times of trucks is minimized, the amount of air pollution that is caused by those trucks will consequently also be reduced.

This chapter is organized as follows. First, in Section 6.1 we give a general model description of the routing process of trucks towards the distribution center via a parking area. Next, in Section 6.2 we outline how this routing process can be modeled as a Markov decision process (MDP). MDPs can be used to find under certain assumptions an optimal policy that results in minimal costs. Unfortunately, the presented MDP becomes intractable quite rapidly, especially in case of large systems. Moreover, in Section 6.2 we argue why we do not use a MDP to find a routing policy. Alternatively, in Section 6.3 we present a simulation model that can be used to investigate the performance of different routing policies. The advantage of such a simulation model is that few assumptions are required, and that the model is

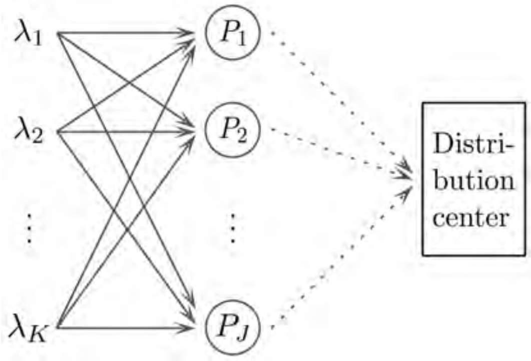


Figure 6.1: The routing process of trucks to a distribution center via a parking area.

flexible, in the sense that it can be adjusted or extended easily. Based on examples, we illustrate the performance of different routing policies.

6.1 Model description

Consider a distribution center with K different access routes or arrival streams, and J different parking areas. Let parking area j consists of N_j parking places. Assume that on access route i trucks arrive according to Poisson process with rate λ_i per time unit. Hence, per time unit, on average λ_i trucks arrive via access route i. Note that any time unit can be chosen – such as minutes, quarters, or hours – as long as it is used consistently. Additionally, let $c_{i,j}$ be the costs of routing a truck that arrived on access route i to the distribution center via parking area j. These costs can for instance represent the travel time or travel distance that corresponds to that specific route. For the convenience of truck drivers, trucks will only be routed to one single parking area. Hence, once a truck driver has been routed to a parking area, he/she will not be rerouted to another parking area. Furthermore, denote with P_j parking area j. In Figure 6.1 the routing process is shown graphically.

In order to develop a routing policy, the distribution of the residence time of trucks at the parking area should be known. Unfortunately we have little insight in the routing process of trucks. Hence, we cannot clearly determine how the residence times are distributed and how they should be included into the model. Additionally, the actual residence time of trucks at a parking area is influenced by multiple factors, including the way in which trucks are routed, the moments at which trucks are routed to a parking area, the scheduled time of customers at the distribution center, the travel time between the parking areas and the distribution center, etc. Hence, in the remaining part of this chapter we just make some assumptions.

Let R be a random variable representing the time between the arrival of a truck at a parking area and the time before he/she is going into service at the distribution center. In the model that we present in this chapter we assume that the distribution of R is known. Initially, we assume that R is exponentially distributed with rate μ. Furthermore, define r_j as the time that is needed to move from parking area j to the distribution center. It is likely that the parking areas are located near the distribution center; hence, the variation in trip duration between the parking area and the distribution center will be small. Therefore, if we assume that a certain travel time margin is included into r_j, it is reasonable to assume that r_j is deterministic. Then, the residence time of a certain customer at parking area j is given by $\max\{R - r_j, 0\}$, whose expected value equals

$$\begin{aligned}\mathbb{E}\max\{R - r_j, 0\} &= \int_{r_j}^{\infty} x f_R(x - r_j)\,\mathrm{d}x \\ &= \int_{r_j}^{\infty} x\mu e^{-\mu(x-r_j)}\,\mathrm{d}x \\ &= \frac{1 + \mu r_j}{\mu}.\end{aligned}$$

Note that when the time required to move from parking area j to the distribution center is similar for all $j \in \{1, \ldots, J\}$, then one can state $r_j = 0$ for all $j \in \{1, \ldots, J\}$. Then, R represents the residence time of any truck at a parking area.

6.2 Markov decision process

As mentioned in Section 6.1 many different factors do influence the routing process. There are several approaches that can be used to investigate which routing policy results in low costs. One of the possibilities is to model the routing process as a *Markov decision process* (MDP). To this end, we assume that $r_j = 0$ for all $j \in \{1, \ldots, \}$, and we consider the simple (but possibly unrealistic) case in which all customers that arrive at a parking area have an exponentially distributed residence time. Thus, we assume that R is exponentially distributed with rate μ. In this case, the state space can be defined as $\mathcal{X} = \{0, \ldots, N_1\} \times \{0, \ldots, N_2\} \times \cdots \times \{0, \ldots, N_J\}$, where a state is denoted by $(n_1, n_2, \ldots, n_J)$. Here, n_j represents the number of parking places that are occupied at parking area j. Additionally, the action space can be defined as $\mathcal{A} = \{1, \ldots, K\} \times \{1, \ldots, J\}$, where action (i, j) means that if a customer of type i arrives, he/she is forwarded to parking area j. Then, one could define a cost function in the form $c(x, a, y)$, representing the direct costs of going from state x to y when action a is chosen, for $x, y \in \mathcal{X}$ and $a \in \mathcal{A}$. This cost function is based on the values $c_{i,j}$.

In fact, one single parking area can be seen as a single-server queue with a limited number of parking places. Suppose that at a certain moment in time there are n truck drivers present at a specific parking area. Then, the service time distribution (or the time before the first customer leaves the parking area) can be seen as the minimum of n exponentially distributed random variables with rate μ, which is again exponentially distributed with rate $n\mu$. Notice that his model has state-dependent service times. However, due to the curse of dimensionality this model becomes intractable quite rapidly. For instance, if we have $J = 5$ different parking areas each with $N_j = 20$ parking places, then the number of different states is $21^5 \approx 4.1$ million. Such a large state space requires a lot of computational effort and memory.

In addition to the computational requirements that are needed to solve the formulated MDP, several assumptions are made which may be in practice not realistic. Unfortunately, MDPs can hardly be adjusted or extended with additional requirements. It may for instance be more realistic to assume that there are different types of truck drivers, having different service time distributions. Truck drivers that arrive

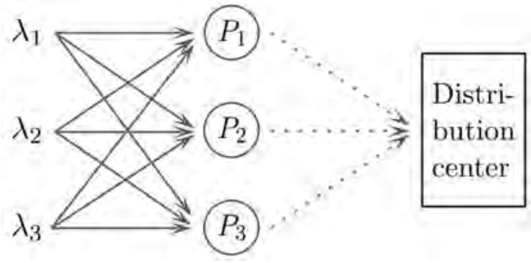


Figure 6.2: Example routing process of trucks to a distribution center via a parking area.

from abroad may for instance have a larger expected residence time at the parking area than truck drivers that arrive from the neighborhood and which are just a bit too early. Different types of customers with different residence time distributions can be taken into account by adding this information to the state space. However, in this case the state space becomes so large that it will be computationally infeasible to derive an optimal routing policy. Another possibility is not to register how many customers of each type have been arrived at each parking area, but to assume that at each parking area the residence time follows a general distribution. However, in order to include this in the model, the time should be added to the state space. Unfortunately, this will result in the same computational problems as mentioned before. Similarly, there are more requirements that cannot be easily added to the model without causing computational problems.

6.3 Simulation

In this section we present a simulation model that can be used to investigate the performance of different routing policies. The advantage of simulation is that few assumptions have to be made, and that practically any feature can be included in the model. In the following paragraphs we illustrate a simulation model with $K = 3$ access routes and $J = 3$ different parking areas, see Figure 6.2.

Suppose that parking area 1 and 3 both consist of $N_1 = N_3 = 10$ parking places, and let the number of available parking places at parking area 2 be given by $N_2 = 5$. Let the arrival rates for each access route be given by $\lambda_1 = \lambda_2 = \lambda_3 = 2$, where we use 'quarters' as time unit. This means that at each access route on average 2 customers arrive per quarter. Additionally, let $r_j = 0$, for all $j \in \{1, 2, 3\}$, and suppose that the residence time R of trucks at a parking area is exponentially distributed with a mean of 4 quarters. Hence, on average, customers have to wait 1 hour before going into service. Let $c_{i,j}$ represents the travel time that is needed for a truck that arrived at access route i to move to parking area j. Define the cost function matrix of this example as

$$C_1 = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 2 & 2 \\ 4 & 2 & 1 \end{pmatrix}.$$

Below, for the presented example we illustrate the performance of different routing policies. Note that it is optimal to route each customer to the parking area with the lowest costs when there are no capacity limitations. Hence, during off-peak hours, such a routing policy can be used. However, during peak hours, or when the number of available parking places is quite limited, a different routing policy can reduce the costs significantly.

We define three different policies. In all three policies trucks that arrive from access route 1 and 3 are routed similarly. Here, both types of arrivals are routed to the parking area with the lowest costs, given that a parking place is available. Specifically, we state:

- Trucks arriving from access route 1 are routed to P_1. If P_1 is full, then they are routed to P_2. If P_2 is full, then they are routed to P_3.
- Trucks arriving from access route 3 are routed to P_3. If P_3 is full, then they are routed to P_2. If P_2 is full, then they are routed to P_1.

The three policies that are defined differ from each other by the routing of the trucks that arrive from access route 2. We define the three routing policies as follows:

- Policy 1. Trucks arriving from access route 2 are routed to P_2. If P_2 is full, then they are either routed to P_1 or P_3. Here, the parking area is chosen with the largest number of available parking places.
- Policy 2. Trucks arriving from access route 2 are either routed to P_1 or P_3. Here, the parking area is chosen with the largest number of available parking places. When both parking areas are full, then the trucks are routed to P_2.
- Policy 3. Trucks arriving from access route 2 are routed to P_2 if the number of available parking places is larger than n, for $n \in \{1, \dots, N_2 - 1\}$. Otherwise, the trucks are either routed to P_1 or P_3. In this case, the parking area is chosen with the largest number of available parking places. When both parking areas are full, then the trucks are routed to P_2.

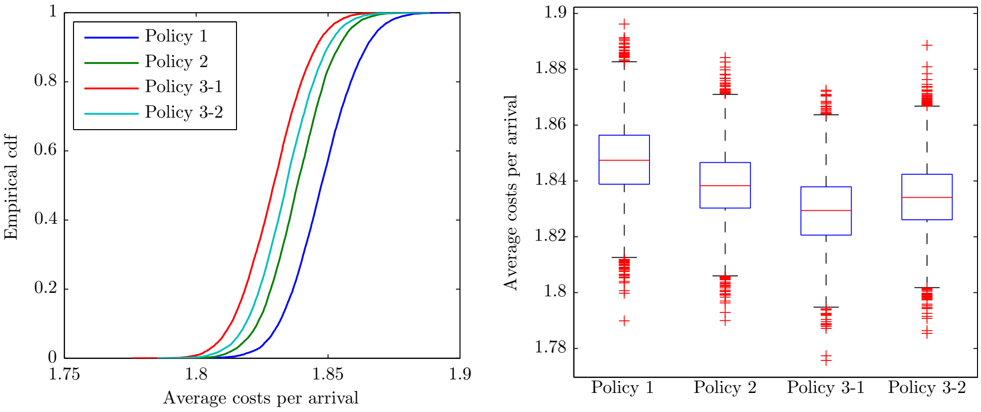
Here, policy 3 can be interpreted as that at parking area 2 n parking places are reserved for customers arriving from access route 1 or 3 that see their nearest parking are full. Moreover, for each policy it holds that when all parking places are occupied, that a newly arriving truck will be routed to the parking area with the lowest costs.

We evaluated the performance of the different routing policies as follows. First, in each simulation we generated 1,000 arrivals as ‘warm up’ for the system, since we are mostly interested in the performance of each routing policy during busy hours. Next, we simulated in total 10,000 new arrivals and listed for each arrival the costs that were made, given a certain routing policy. At the end of the simulation we calculated the average costs per arrival. For each routing policy, we repeated this procedure 10,000 times. Within the simulation model we assumed that the parking area to which a customer is routed is determined just before the arrival at the access route, and that once the parking area is determined, a parking place will be reserved for the newly arrived customer from that moment on.

By simulating the routing process of trucks for different policies, we obtain the statistics for the cost function as given in Table 6.1. Here, ‘policy 3-n’ is a short notation for policy 3 in which the trucks that arrive at access route 2 are initially routed to P_2 if the number of available parking places is larger than n. In Figure

Policy	Mean	Variance
1	1.8477	$0.1688 \cdot 10^{-3}$
2	1.8385	$0.1459 \cdot 10^{-3}$
3-1	1.8293	$0.1572 \cdot 10^{-3}$
3-2	1.8342	$0.1499 \cdot 10^{-3}$

Table 6.1: Statistics of the average costs per arrival for different routing policies.



(a) Empirical cdf.

(b) Boxplots.

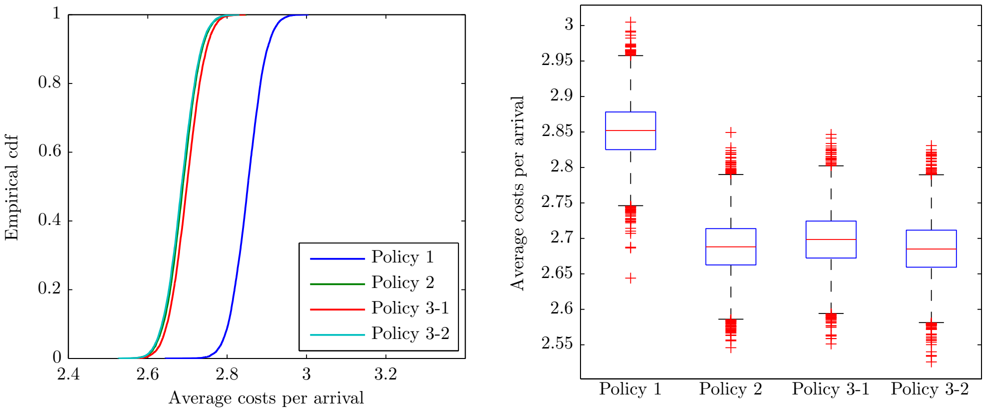
Figure 6.3: Graphs representing the average costs of per arrival for different routing policies, using cost matrix C_1.

6.3(a) the empirical cumulative distribution function (cdf) of the average costs per arrival is shown. In this figure differences between the performance of the different routing policies can be seen. Clearly, policy 1 is not optimal, whereas this policy is optimal when there are no capacity limitations. For this specific example, policy 3-2 results in the lowest overal costs. This policy can be interpreted as that at P_2 1 parking place is reserved for either customers of type 1 that see P_1 full, or customers of type 3 that see P_3 full. The differences in the performance of the routing policies are also confirmed by the boxplots as given in Figure 6.3(b). Notice however that the boxplots still overlap. Hence, for this example we cannot conclude with high confidence that for each combination of policies one policy is significantly better than another.

Moreover, if we evaluate the performance of the different routing policies for a different cost function matrix, such as

$$C_2 = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 10 \\ 2 & 2 & 2 \\ 10 & 2 & 1 \end{pmatrix},$$

then the differences between the policies become much clearer. In Figure 6.4 the simulation results for the average costs per arrival using cost matrix C_2 are shown graphically. Figure 6.4(b) clearly shows that policy 1 is not optimal, as the boxes do not overlap. Similarly, the simulation model can be used for different cost matrices



(a) Empirical cdf.

(b) Boxplots.

Figure 6.4: Graphs representing the average costs of per arrival for different routing policies, using cost matrix C_2.

and different routing processes in order to investigate the performance of different routing policies.

Chapter 7

Conclusions and Further Research

In this chapter the conclusions that can be drawn from this research will be presented and several directions for further research will be given in Sections 7.1 and 7.2 respectively.

7.1 Conclusions

In this thesis we presented two different models that can be used to schedule trucks at a distribution center: a model based on outpatient appointment scheduling, and a model based on job shop scheduling. Under certain circumstances the appointment scheduling model converges to an optimum, in the sense that a weighted sum of the customer's waiting time and the lateness or the idle time of the schedule is minimized, whereas (as second objective) customers are scheduled as closely as possible to their preferred time interval. It has been found that for many different parameter values the evenly-spaces schedule is optimal. Nevertheless, in practice the weights that should be used for the customer's waiting time, the crossdock's idle time and the lateness of the schedule cannot be clearly defined.

Empirical studies have shown that trucks usually have different service time distributions. However, when the assumptions that are made in the appointment scheduling model – such as equally distributed service times – do not hold, then the resulting schedule can be quite bad. We have shown that the length of the expected service time does have a large impact on the expected waiting time of all subsequently scheduled customers. Moreover, the appointment scheduling model does have a large run time (multiple hours), and can hardly be adjusted or extended with additional requirements that are preferred to take into account.

As an alternative to the appointment scheduling model, we presented a multi-crossdock job shop scheduling model. In this model many different extensions and additional requirements can be taken into account. For multiple (service time) distributions a linear relationship has been found between the average service time and the service time delay. We have illustrated how this relationship can be used in order to minimize the waiting time of customers, when customers have different service time distributions. In addition, we have shown how (high-priority) customers that arrive during the day can be included into a schedule that is already in process.

The presented job shop scheduling model results in a schedule with low costs, has a short run time (a few seconds), can be applied to large distribution centers, and can easily be extended with additional requirements or preferences.

Next, we developed a simulation model that can be used to investigate in which way trucks should be routed to parking areas in order to minimize a certain cost function, such as the travel time or travel distance. When there are no capacity limitations, then it is optimal to route trucks to the parking area with the lowest costs. We have illustrated that when there are capacity limitations, that this policy is not optimal in all cases. In some cases a policy in which a certain number of parking places is 'reserved' for specific types of customers results in much lower overall costs.

7.2 Further research

In this section we outline different directions for further research. In this report we presented different models that can be used to schedule trucks at a distribution center. Unfortunately, at several points there is a lack of practical insight. Therefore, as a next step, we recommend to obtain more understanding and insight. Additionally, there is a lack of empirical data. The results we presented with respect to the loading and unloading times of trucks are based on empirical studies from literature. When there is any data available, one could get more insight in the loading and unloading process of trucks via a data analysis. The same holds for the routing process of trucks to the distribution center, for which practical insight and empirical evidence lacks. Below we list several topics for further research which can mostly be conducted once there is more practical and empirical insight.

- Insight in the arrival process of trucks and the distribution of the loading and unloading times of trucks.

- The preferences and requirements that should be taken into account in the scheduling model. In this thesis we presented a scheduling model that is widely applicable, from different perspectives. However, there may be additional requirements or preferences we do not know about, but which are relevant to take into account.

- A validation of the scheduling model using real data, or by means of a real test case. This can for instance be done by creating schedules on the basis of real data, and then analyzing the performance of the schedule via simulation.

- Insight in the routing process of trucks to the distribution center, and insight in the residence time of trucks at the parking areas. Once there is more information available, this can be included into the simulation model. Additionally, a direction for further research is to develop an algorithm that finds the optimal routing policy. However, in order to do this, more information should be gathered.

Depending on the results of the topics listed above, the developed scheduling and routing models can further be adjusted or extended.

Notation

a_i	Start of time interval in which customer i prefers to be scheduled.
b_i	End of time interval in which customer i prefers to be scheduled.
B_i	Random variable representing the service time of customer i.
B_s	Random variable representing the service time of scheduled customers.
B_u	Random variable representing the service time of unscheduled customers.
$c_{i,j}$	Costs per unit of time of routing a truck that arrived from access route i to parking area j.
$c_{W,i}$	Waiting costs per time unit for customer i.
$c_{W,i,t}$	Waiting costs per time unit for customer i during time interval t.
$C(x)$	Costs of schedule x.
D_i	Number of time intervals in which customer i will be in service.
$f(i,t)$	Costs of scheduling customer i at time interval t.
$f(i,j,t)$	Costs of scheduling customer i at crossdock j at time interval t.
$F_{B_i}^{-1}(\cdot)$	Inverse cumulative distribution function of the service time of customer i.
$g(i,j,t)$	Costs of (re)scheduling customer i at crossdock j at time interval t.
$I(x)$	Expected idle time of schedule x.
$L(x)$	Expected lateness of schedule x.
M	Number of crossdocks.
$\mathcal{M}$	Set of crossdocks at which customers can be scheduled.
n_t	Number of customers scheduled at time interval t.
$n_{j,t}$	Number of customers scheduled at crossdock j at time interval t.
N	Number of customers.
N_j	Number of parking places at parking area j.
$\mathcal{N}$	Set of customers that need to be scheduled.
$\mathcal{N}_{\text{new}}$	Set of customers that need to be scheduled including new customers, being either dummy or unscheduled customers.
$\mathbb{N}_0$	Set of non-negative numbers $\{0, 1, 2, \ldots\}$.
$\mathbb{N}_1$	Set of positive numbers $\{1, 2, 3, \ldots\}$.
P_j	Parking area j.
r_j	The time that is needed to move from parking area j to the distribution center.
R	Random variable representing the time between the arrival of a truck at a parking area and the time before he/she is going into service at the distribution center.
s	Time interval from which on a reschedule should be made.
$s_{i,t}$	Variable indicating whether customer i is served during time interval t ($s_{i,t} = 1$) or not ($s_{i,t} = 0$).

$s_{i,j,t}$	Variable indicating whether customer i is served at crossdock j during time interval t ($s_{i,j,t} = 1$) or not ($s_{i,j,t} = 0$).
S_i	Random variable representing the number of time units of work that arrives at the beginning of any time interval, given that i customers are scheduled to arrive.
$\bar{t}$	The last time interval before t on which any customer is scheduled.
$t_{1,i}$	Start of time interval in which customer i is not allowed to be scheduled.
$t_{2,i}$	End of time interval in which customer i is not allowed to be scheduled.
t_i^{old}	Time interval at which customer i was originally scheduled.
T	Number of time intervals.
T_{earliest}	Earliest time interval at which the schedule could be finished.
T_{end}	Time at which the schedule should be finished.
T_{shortest}	The smallest number of time intervals in which the schedule could be finished.
$\mathcal{T}$	Set of time intervals on which customers can be scheduled.
$\mathcal{T}_i^{\text{end}}$	Set of time intervals at which customer i is not allowed to be scheduled in order to finish the schedule before T_{end}.
$\mathcal{T}_i^{\text{not}}$	Set of time intervals at which customer i is not allowed to be scheduled.
$\mathcal{T}_{\text{new}}$	Set of time intervals on which customers can be scheduled including additional time intervals at the beginning or end.
$\mathcal{T}_s$	Set of time intervals smaller than s on which the schedule is not allowed to change.
$W(x)$	Expected waiting time of schedule x.
$W_{i,t}$	Random variable representing the waiting time of the ith scheduled customer at time interval t.
x	Vector representing the schedule $(n_1, \ldots, n_T)$.
$x_{i,t}$	Variable indicating whether customer i is scheduled at time interval t ($x_{i,t} = 1$) or not ($x_{i,t} = 0$).
$x_{i,t}^{\text{old}}$	Variable indicating the original (old) value of $x_{i,t}$.
$x_{i,j,t}$	Variable indicating whether customer i is scheduled at crossdock j at time interval t ($x_{i,j,t} = 1$) or not ($x_{i,j,t} = 0$).
X_t^-	Random variable representing the number of time units of work in the system just before any arrivals at time interval t.
X_t^+	Random variable representing the number of time units of work in the system just after any arrivals at time interval t.
Y	Random variable representing the number of unscheduled high-priority customers that arrive at the beginning of any time interval.
α_I	Weight of the expected idle time $I(x)$.
α_L	Weight of the expected lateness $L(x)$.
α_R	Weight of rescheduling.
α_W	Weight of the expected waiting time $W(x)$.
β_i	Mean service time of customer i.
β_s	Mean service time of scheduled customers.
β_u	Mean service time of unscheduled high-priority customers.
Δ	Length of a time interval.
Δ_i^{min}	Minimum number of time units that need to be scheduled after customer i.
γ	Maximum factor with which the service durations of all customers can be increased such that the schedule is finished before T_{end}.

γ_{LB}	Lower bound for γ.
γ_{UB}	Upper bound for γ.
λ	Arrival rate of unscheduled high-priority customers per time interval.
λ_i	Arrival rate of trucks coming from access route i.

Bibliography

M. Angalakudati, S. Balwani, J. Calzada, B. Chatterjee, G. Perakis, N. Raad, and J. Uichanco. Business analytics for flexible resource allocation under random emergencies. *Management Science*, pages 1–22, 2014. Published online in Articles in Advance.

K. Bübül and P. Kaminsky. A linear programming-based method for job shop scheduling care services. *Journal of Scheduling*, 16:161–183, 2013.

P.M. Vanden Bosch and D.C. Dietz. Minimizing expected waiting in a medical appointment system. *IIE Transactions*, 32(9):841–848, 2000.

P.M. Vanden Bosch and D.C. Dietz. Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research*, 4(1):15–25, 2001.

P.M. Vanden Bosch, D.C. Dietz, and J.R. Simeoni. Scheduling customer arrivals to a stochastic service system. *Naval Research Logistics*, 46(5):549–559, 1999.

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

Dutch National Air Quality Cooperation Programme (NSL). Monitoring nsl 2013, June 2014. `https://www.nsl-monitoring.nl/viewer/`.

European Union. Directive 2008/50/EC of the European parliament and of the council on ambient air quality and cleaner air for Europe. *Official Journal of the European Union*, June 2008.

W.N.A. Wan Ahmad Fatthi, A. Shuib, and R.M. Dom. Estimating unloading time at cross docking centre by using fuzzy logic. *Research Journal of Business Management*, 7(1):1–14, 2013.

A. Franz and R. Stolletz. Performance analysis of slot-based appointment scheduling for truck handling operations at an air cargo terminal. University of Mannheim, Germany, 2012.

U. Gehring et al. Traffic-related air pollution and the development of asthma and allergies during the first 8 years of life. *American Journal of Respiratory and Critical Care Medicine*, 181:596–603, 2010.

Gurobi Optimization. Gurobi 5.6 performance benchmarks, 2013. `http://www.gurobi.com/pdf/Benchmarks.pdf`.

M. Jerrett et al. A cohort study of traffic-related air pollution and mortality in Toronto, Ontario, Canada. *Environmental Health Perspectives*, 117:772–777, 2009.

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

M.K. Kiesling and C.M. Walton. Loading/unloading operations and vehicle queuing processes at container ports. Research report, Center for Transportation Research, The University of Texas at Austin, 1995.

N. Künzli et al. Public-health impact of outdoor and traffic-related air pollution: a European assessment. *The Lancet*, 356:795–801, 2000.

T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R.E. Bixby, E. Danna, G. Gamrath, A.M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D.E. Steffy, and K. Wolter. Miplib 2010. *Mathematical Programming Computation*, 3(2):1–61, 2011.

P.M. Koeleman and G.M. Koole. Optimal outpatient appointment scheduling with emergency arrivals and general service times. *IIE Transactions on Healthcare Systems Engineering*, 2:14–30, 2012.

B. Meindl and M. Templ. Analysis of commercial and free and open source solvers for linear optimization problems, March 2012. Forschungsbericht, Technische Universität Wien.

National Institute for Public Health and the Environment (RIVM). Grootschalige concentratie- en depositiekaarten Nederland (GCN en GDN), June 2014. `http://geodata.rivm.nl/gcn/`.

F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity.* Springer, 2010.

P.L. Nguyen, R. Hoogerbrugge, and F. van Arkel. Evaluation of the representativeness of the dutch national air quality monitoring network. Technical report, National Institute for Public Health and the Environment (RIVM), 2009.

Inc. The Tioga Group. Port metro Vancouver truck turn time study: Analysis, results and recommendations, June 2013. `http://tiogagroup.com/docs/PortMetroVancouverTruckTurnTimeStudy2013.pdf`.

J.D. Welch and N.T.J Bailey. Appointment systems in hospital outpatient departments. *The Lancet*, 259:1105–1108, 1952.

A.M.S. Zalzala and P.J. Fleming. *Genetic Algorithms in Engineering Systems.* The Institution of Electrical Engineers, 1997.

Appendix A

Discrete-time Performance Measures

In this chapter it will be shown how the expected waiting time of a schedule x can be calculated when the service times and waiting times follow a discrete distribution. Here, a similar notation will be used as in Section 5.1.

The first scheduled customer has no waiting time. Hence,

$$\mathbb{P}(W_{1,1} = 0) = 1,$$

and for $j \in \mathbb{N}_1$

$$\mathbb{P}(W_{1,1} = j) = 0.$$

All subsequent customers do have a positive waiting time probability. The probability that the first customer scheduled at any time interval $t \in \{2, \ldots, T\}$ has no waiting time is given by

$$\begin{aligned}\mathbb{P}(W_{1,t} = 0) &= \mathbb{P}(W_{n_{\bar{t}},\bar{t}} + B_{n_{\bar{t}},\bar{t}} \leq \Delta[t - \bar{t}])\\ &= \sum_{k=0}^{\Delta[t-\bar{t}]} \mathbb{P}(W_{n_{\bar{t}},\bar{t}} + B_{n_{\bar{t}},\bar{t}} = k)\\ &= \sum_{k=0}^{\Delta[t-\bar{t}]} \sum_{m=0}^{k} \mathbb{P}(W_{n_{\bar{t}},\bar{t}} = m)\mathbb{P}(B_{n_{\bar{t}},\bar{t}} = k - m).\end{aligned}$$

For $t \in \{2, \ldots, T\}$ and $j \in \mathbb{N}_1$ we have

$$\begin{aligned}\mathbb{P}(W_{1,t} = j) &= \mathbb{P}(W_{n_{\bar{t}},\bar{t}} + B_{n_{\bar{t}},\bar{t}} = \Delta[t - \bar{t}] + j)\\ &= \sum_{k=0}^{\Delta[t-\bar{t}]+j} \mathbb{P}(W_{n_{\bar{t}},\bar{t}} = k)\mathbb{P}(B_{n_{\bar{t}},\bar{t}} = \Delta[t - \bar{t}] + j - k).\end{aligned}$$

For any customer $i \in \{2, \ldots, n_t\}$ the probability of waiting depends on the waiting time and service time of the previous scheduled customer at that time interval. Thus, for $j \in \mathbb{N}_0$,

$$\begin{aligned}\mathbb{P}(W_{i,t} = j) &= \mathbb{P}(W_{i-1,t} + B_{i-1,t} = j)\\ &= \sum_{k=0}^{j} \mathbb{P}(W_{i-1,t} = k)\mathbb{P}(B_{i-1,t} = j - k).\end{aligned}$$

By using the waiting time probability distribution as defined above, the expected waiting time of the ith customer scheduled at time interval t is given by

$$\mathbb{E}W_{i,t} = \sum_{j=0}^{\infty} j\mathbb{P}(W_{i,t} = j).$$

Then, the expected waiting time $W(x)$ of schedule x is defined by the sum of all these expectations. Thus,

$$\begin{aligned} W(x) &= \sum_{t=1}^{T} \sum_{i=1}^{n_t} \mathbb{E}W_{i,t} \\ &= \sum_{t=1}^{T} \sum_{i=1}^{n_t} \sum_{j=0}^{\infty} j\mathbb{P}(W_{i,t} = j). \end{aligned}$$