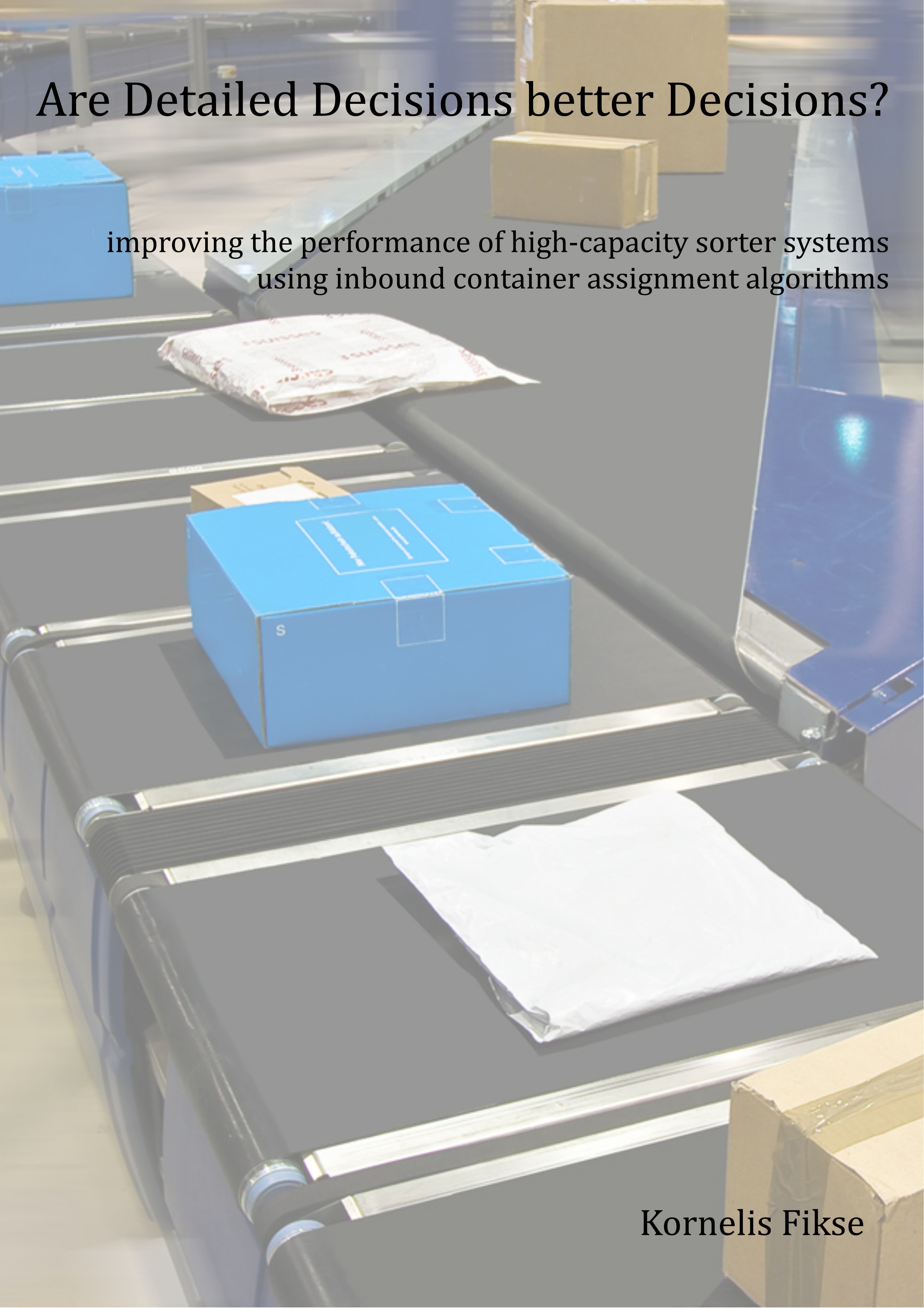


Are Detailed Decisions better Decisions?

improving the performance of high-capacity sorter systems using inbound container assignment algorithms

Kornelis Fikse

Are Detailed Decisions Better Decisions?

improving the performance of high-capacity sorter systems using inbound container assignment algorithms

Kornelis Fikse, MSc

Elburg, 15th November 2011

In fulfilment of the Master Degree
Industrial Engineering & Management, University of Twente, The Netherlands

Graduation Committee

Dr. ir. J.M.J. Schutten

Ir. G.J.A.M. Weijenberg

Ir. S.W.A. Haneyah, MSc

UNIVERSITY OF TWENTE.

VANDERLANDE INDUSTRIES

Summary

High-capacity sorter systems are used worldwide in the express parcel and aviation industry. The ability to sort thousands of items per hour enables express parcel companies to deliver items overnight at the same continent. Furthermore, they provide airports and airlines with the ability to sort the baggage items for thousands of passengers to dozens of flights within the three hours that are available for check in. Express parcel companies as well as airports frequently use a first-come-first-served (FCFS) policy when deciding which container should be unloaded on a sorter system infeed. However, often the contents of individual containers are known, and this information could be used when selecting a container. This thesis therefore focusses on how knowledge about the contents of containers could be used when selecting the next container to be unloaded, in order to improve the performance of high-capacity sorter systems.

The literature review shows that this topic has received little attention from researchers. There are a few papers that focus on the development of detailed and complicated (un)loading schedules, but these approaches are time consuming. Their use is therefore limited to offline scheduling, i.e. creating fixed unloading schemes. The dynamic load balancing algorithm (DLBA), however, is specifically developed for online inbound container scheduling at parcel and postal hubs. Each time an infeed becomes available, the approach selects that container from the queue that minimises the workload imbalance on the sorter. More specifically, the approach ensures that the number of items (i.e. workload) per destination on the sorter system is more or less equal.

Unfortunately, the DLBA has some shortcomings. It does, for instance, not take the internal transport times into account. Especially in larger sorter systems this could severely affect the selection decision being made. Therefore, an adapted version, the advanced dynamic load balancing algorithm (ADLBA), has been developed in this research. This algorithm estimates when items will arrive at the outfeed, and selects containers in such a way that the workload at each of the outfeeds is, at all times, more or less equal.

Furthermore, two extensions for the baggage handling industry have been developed. The priority extension acknowledges that some containers might have priority over others. Priority should, for instance, be given to a container that mainly contain items that have to make a flight that departs soon. The delayability extension, on the other hand, sends containers holding only items that currently cannot be sorted, because their destinations have not yet been assigned to an outfeed, to a remote container park. This enables the dispatcher to select only containers that can be unloaded and significantly reduces the number of items that are sent to the early baggage system (EBS). Additionally, this could reduce the workload at the main sorter, thereby improving the performance.

In order to determine the performance of each of the scheduling approaches, three simulation models and four simulation scenarios are developed. The simulation studies show that especially the DLBA is able to improve the performance of sorter systems, whereas the ADLBA performs equal to, or even worse than, current practice FCFS. In fact, results show that the DLBA is able to increase the throughput of sorter systems up to 4.5% for parcel and postal. Furthermore reductions in missort rate (number of items per thousand that arrive too late) of several permillage points, depending on the workload on the sorter system, are possible.

The results for the extensions are twofold. The priority extension proves to be of little use, for it reduces the missort rate only marginally. The delayability extension, on the other hand, performs extremely well and is able to reduce the missort rate by percentage points, again depending on the workload of the sorter system.

We therefore conclude that the workload balancing approach can be very interesting. The DLBA is able to improve the results for both the baggage handling and the parcel and postal industry. Because the ADLBA performs rather poorly, we conclude that a more detailed approach does not necessarily result in better scheduling decisions. The results for the extensions do, however, show that a suitable extension can result in an increasing performance as well.

Further, we recommend to continue research on workload balancing approaches, but focus should be on more general, less detailed, approaches. The results for the delayability extension show that elementary approaches developed for either of the industries can contribute significantly to the performance, but are easier to implement. Future research should therefore not fixate itself on finding a single solution for both trades, but aim at finding good solutions for either of them.

Contents

CHAPTER 1

Introduction

This chapter presents the problem that is the focal point of this thesis. Section 1.1 therefore introduces the company where the research has been performed and the material handling systems that are the subject of interest. Section 1.2 discusses the research problem in greater detail and explains why additional research could be beneficial for the material handling industry.

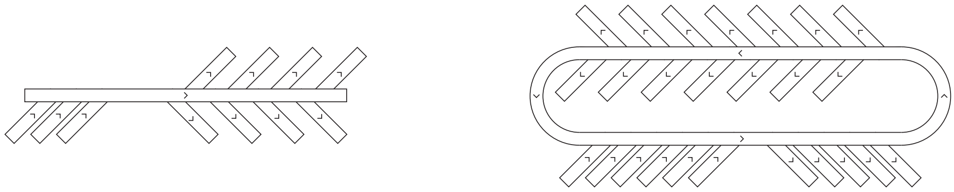
Section 1.3 focusses on the formulation of the primary and secondary research goal and Section 1.4 formulates and underpins the six research questions that should lead to the attainment of these research goals. Finally, Section 1.5 provides the outline of the remainder of this thesis.

1.1 Background

Vanderlande Industries (Vanderlande) is a Dutch company that *'provides automated material handling systems and services'* (Vanderlande Industries [VI], 2010). The company focusses on three different markets: baggage handling systems at airports, automated logistic systems in distribution centres, and sorting solutions in parcel and postal facilities (VI, 2010). These three markets are quite different indeed: baggage handling systems are provided by airports and used by airlines, which can be considered their customers;[1] distribution systems are used for *building* and moving pallets with goods, order-picking, or both; and parcel and postal systems are usually located at multiple locations in a network structure, serving many (small) customers.

This research focusses on the sorter systems that are currently provided by Vanderlande for the baggage handling and parcel and postal industries. We do, however, try to describe the sorter systems, business processes, and approaches in a generic way to ensure that this research and its results can

[1] In the USA, however, airlines also own the baggage handling systems. It is therefore not uncommon that there are multiple baggage sorting systems present at a single airport.



(a) Line Configuration (b) Loop Configuration

Figure 1.1: Physical Configuration of Sorter Systems

also be used for (and applied to) future sorter systems and problem settings that are not discussed in this thesis.

At Vanderlande, two different sorter systems are identified: line sorters and loop sorters. For this research, however, the difference between line and loop sorters is of minor importance, for we aim at developing one approach for both systems. Differences between physical layouts, however, can be important. We therefore distinguish sorter systems with a 'line configuration' (Figure 1.1a) from sorter systems with a 'loop configuration' (Figure 1.1b). Focus, however, is on sorters in loop configuration.

Currently, Vanderlande focusses on delivering the hardware, software, and maintenance services for sorter systems. More specifically, a 'door-to-door' policy is applied: what customers do before an item is placed on an *infeed* (conveyor that transports items onto the main sorter) or after an item has been retrieved from an *outfeed* (catchment conveyor after an item has been sorted) is outside Vanderlande's scope. However, given the fierce competition, interest has aroused for providing additional services to customers.

One of the services that is considered, is providing customers with tools to use sorter systems more efficiently. This service is especially interesting because it allows customers with existing equipment to increase throughput without installing expensive additional equipment. Customers that require a new installed base also benefit, because they get a system with performance figures that could otherwise only be achieved by installing more expensive and space taking equipment. Obviously these type of services can significantly improve the competitive position of Vanderlande: providing systems with similar performance but lower costs by showing your customers how to use the equipment efficiently is a selling point.

This research aims at developing one of the tools that could be used to make better use of the existing sorter capacity. However, a more specific and better defined problem is needed before an approach can be developed. Section 1.2 therefore describes the problem that is studied in this research.

1.2 Problem Description

Something that makes it difficult to define a single problem for this research, is that baggage handling and parcel and postal are two separate industries, each with their own business processes, equipment, and techniques. However, it is interesting to see that there are also similarities between the two industries.

Perhaps the most interesting similarity, at least for this research, is the assignment of inbound trailers, ramp carts, and Unit Load Devices (ULDs; a standard type of container used by airlines all over the world) to the infeeds of the sorter system. Parcel and postal as well as baggage handling companies use a kind of first-come-first-served (FCFS) policy when deciding when and where the conveyables[2] should be unloaded. In parcel and postal, the dispatcher will typically queue all arriving trailers. As soon as an unloading dock becomes available, the trailer that is first in line will be called for and sent to the unloading dock. In baggage handling, the tug driver (a tug is the tractor that pulls the ramp carts with loose baggage or the dollies on which the ULDs are loaded) drives towards the transfer baggage infeed area and joins the queue of a particular unloading *lateral* (a type of conveyor frequently used in baggage handling), which is usually selected using the existing queue length.

The problem with the FCFS approach is that it is a 'simple' approach. Although a lot is known about the contents of specific trailers a/or ULDs, this knowledge is not used when unloading the conveyables onto the sorter system. In fact, by using the FCFS approach, or any other non-workload-balancing approach, uncontrolled peak flows for a particular outfeed could arise, causing the outfeed conveyor to fill up completely. Although this in itself is not a very serious problem, one might even argue that if the outfeed or lateral is never full, too much money has been invested in these outfeeds, the consequences can be significant.

Unfortunately, those full outfeeds reduce the capacity (measured in sorted conveyables per hour) or at least increase material handling costs. When an outfeed is full, a sorter in line configuration will transport the conveyable to the outfeed for unsorted conveyables, which is a large catchment area at the end of the sorter system. The capacity of the sorter system is indirectly reduced, because the unsorted conveyables have to be loaded onto the beginning of the sorter system again for a second delivery attempt. The other solution is that a worker manually delivers the conveyable to the right outfeed, but this significantly increases material handling costs, as additional labourers have to be hired. In a sorter system in loop configuration, a full outfeed will result in *recirculation*, i.e. the conveyable is transported through the entire sorter system again before a second delivery attempt can be made. This reduces the sorter capacity directly, since a recirculating conveyable claims space that

[2] Conveyables is a container term for baggage, parcels, and other items that have to be sorted *and* can be carried on a sorter system.

otherwise could have been used for a 'new' conveyable.

Recirculation is therefore one of the reasons why there is a significant difference between *operational peak capacity* and *design capacity*. The operational peak capacity, which is typically defined by the client, is a theoretical construct that approximates the workload that the sorter system should be able to handle during peak hours. The design capacity is the theoretical capacity of the system, assuming a perfect flow of identical conveyables. Due to practicalities, such as merging difficulties and recirculation, the difference between operational peak and design capacity is about 15%. Reducing recirculation by balancing the workload could therefore increase the operational peak capacity on existing systems or reduce the required design capacity for future sorter systems.

The problems associated with recirculation, either directly or through manually loading conveyables onto the sorter system for the nth time, can go far beyond capacity reduction. For example in baggage handling, this could cause baggage, either baggage that has to recirculate or baggage that could not be loaded onto the sorter system due to recirculating baggage, to miss its flight. Delivering a single bag to its rightful owner when the bag has missed its flight can cost the airline up to 100 US$ in 2009 (SITA, 2010). Therefore, it makes sense that airlines and airports aim at delivering all baggage on time. Until recently, these problems did not exist in the parcel and postal business. In the past decades and especially in the past years, however, more and more focus has been given to reliable delivery. Customers demanding that a parcel is delivered anywhere in Europe within a day are not uncommon any more, and meeting these demands requires a very tight schedule for aeroplanes and trailers. It is therefore becoming more and more important to sort as many parcels as possible before the outfeed closes (*cutoff time*).

Summarised, both industries aim at sorting all conveyables before the corresponding outfeed closes. However, due to the aforementioned uncontrolled peak flows, in combination with cutoff times for aeroplanes and trailers, airports and parcel and postal handlers sometimes fail to achieve this goal. Therefore, the main problem this research focusses on, is the existence of these uncontrolled peak flows and how to prevent them. The next section explains what this research aims to achieve.

1.3 Research Goal

We already mentioned that a lot of knowledge is available with regard to the contents of trailers, ramp carts, and ULDs.[3] When, in the parcel and postal

[3] For convenience, we will use the term 'container' when referring to either trailers, ramp carts, or ULDs. If something is related to trailers, ramp carts, or ULDs specifically, we will use these terms instead.

industry, a trailer leaves the depot, the destination zipcode and also other information that is less useful for our research is known for each parcel inside it. Similarly, when ULDs are unloaded from an aeroplane, for each individual ULD the contents and its connecting flight are known. When in the baggage handling industry loose baggage is loaded onto ramp carts for transfer, the connecting flight for each of the bags is known, but information about which bag is on which cart is not available.

The suggestion is that we can use this knowledge to prevent uncontrolled peak flows at the outfeeds of the sorter system. In fact, if we have full knowledge (i.e. we know exactly when a container arrives and we know the contents exactly) we could device an unloading schedule (i.e. assign each container to a specific infeed at a specific time) such that the workload on the outfeeds never exceeds the threshold that would cause an outfeed to be full. Of course the capacity of a specific outfeed is dependent on the number of workers that remove the conveyables from the system and the time they need to process each individual conveyable. Unfortunately a situation with full knowledge as assumed here rarely occurs in practice.

In our opinion we could device an approach that determines the unloading schedule, even if not all information is available beforehand. One could for instance determine for each container that arrives at the transfer baggage infeeds which queue it should join, based on a workload-balancing principle. Other options are to device a schedule based on predictions and only defer from this schedule when trailers or aeroplanes are indeed significantly delayed. Generally speaking it should be possible to develop an approach that can be used to intelligently assign containers to infeeds. Therefore, we can define the following primary research goal:

> *Development of an approach that balances workload on the outfeeds of a sorter system, by assigning containers to infeeds using the conveyable destination data.*

However, when describing the baggage handling and parcel and postal business processes, but mainly for baggage handling, it became clear that there are many more decisions that can be made around the scheduling of containers. We could for instance decide to park ULDs and ramp carts temporarily on the apron, which would allow us to give priority to other ULDs and ramp carts. Furthermore we found that larger airports often have multiple transfer areas, which makes us face the decision to which transfer area we should refer specific ULDs a/or ramp carts, and, given that we have multiple areas where we can park ULDs a/or ramp carts, a similar problem arises in determining at which parking they should wait before unloading them onto a transfer lateral. Briefly, we can state that we can develop multiple extensions that can be applied to a specific industry, or sometimes even to both industries. We therefore define a secondary research goal, which is:

> *Development of extensions to the workload-balancing-approach, which could further improve material handling processes using information on conveyable destination and yard layout.*

With the research problem expounded earlier and the research goals described above, we have defined the focal point of this research: the scheduling of inbound operations at sorter systems. This thesis discusses current practice and newly developed techniques that are, one way or the other, related to this focal point. The research questions, defined in the next section, therefore aim at gaining knowledge about these subjects.

1.4 Research Questions

Research questions should structure the road that leads to the attainment of the research goal. That is, when all questions have been answered, the research goal should be achieved. Based on the subjects introduced in the previous sections, three main questions can be defined: 'what is current practice?', 'what approaches can be developed to tackle the issues associated with current practice?', and 'what is the performance of these scheduling approaches?'

The first main question 'what is current practice?' can be divided in three parts, for it focusses on three topics: business processes, literature, and existing scheduling approaches.

Discussing the business processes around sorter systems is important for two reasons. It gives insight in the system- and conveyable information that is available and provides a framework in which the scheduling approaches should function. Because this research aims at developing a scheduling approach that can be applied to any sorter system, a business process model which suits both industries is preferred. The research question focussing on these issues is: *'what business processes are present around the sorter systems in the different industries?'*

A literature review is important, because it provides valuable knowledge about approaches that have been developed by other researchers. Besides solutions for the problem at hand, which are obviously interesting, literature might also provide partial solutions which can be used as part of a new solution. In short: *'what knowledge can be obtained from existing literature?'*

The existing scheduling approaches, those that are currently applied at sorter systems as well as state-of-the-art techniques from literature, are the concluding piece of this umbrella question. The existing scheduling approaches provide a standard used to assess the newly developed techniques, for only scheduling approaches that outperform existing ones are, from an operational perspective, relevant. The question to be answered is: *'what are current practice and state-of-the-art scheduling approaches?'*

The second main question 'what approaches can be developed to tackle the issues associated with current practice?' focusses on the development of new scheduling techniques and can be divided into two parts.

The first part focusses on the development of a workload balancing approach that is capable of handling realistic scheduling problems. Therefore this part not only discusses issues that current approaches cannot cope with, it also provides solutions for these issues. The corresponding research question is: *'what approach can be developed to tackle the issues current scheduling techniques cannot cope with?'*

The focal point of the second part is the development of generic extensions, which enable scheduling approaches to deal with challenges that are typical for the baggage handling industry. Again, this part not only discusses the challenges, it also provides solutions to them. The research question that accompanies this part is: *'what generic extensions can be developed to handle industry specific challenges?'*

The previous questions provide existing scheduling approaches or focus on the development of new ones. The third main question therefore aims at determining which scheduling approach would be preferred. Subjects that are part of this question are the development of performance indicators and the methodology that is used to determine this performance. The last research question is thus: *'what is the performance of the different scheduling approaches?'*

Summarised, the following research questions are formulated:

1. What business processes are present around the sorter systems in the different industries?
2. What knowledge can be obtained from existing literature?
3. What are current practice and state-of-the-art scheduling approaches?
4. What approach can be developed to tackle the issues current scheduling techniques cannot cope with?
5. What generic extensions can be developed to handle industry specific challenges?
6. What is the performance of the different scheduling approaches?

1.5 Outline

The outline of the thesis is guided by the research questions defined above. It therefore starts with explaining the background of the problem, introducing various scheduling approaches, and, finally, testing these approaches to determine which one is most suitable.

Chapter 2 focusses on the business processes around sorter systems. The main goal is the development of a combined process model that describes both the baggage handling and parcel and postal industry. This model is necessary for the assessment of existing, and the development of new, scheduling approaches. Chapter 3 discusses literature, related to the scheduling problem at hand, from the baggage handling, the parcel and postal, as well as the distribution industry. This literature overview serves two purposes. First, it provides an overview of the state-of-the-art scheduling techniques from literature, and, second, it provides interesting methods that could be used in the development of a new scheduling approach. Chapter 4 concludes the first main question and describes the situation, regarding scheduling approaches, as it stands. It pays attention to three approaches, one that is frequently applied in reality, one theoretical solution, and one state-of-the-art approach from literature.

Chapter 5 focusses on the development of a new scheduling approach that tackles the drawbacks of the existing approaches. The purpose is to develop a workload balancing algorithm that is also capable of handling very large and complex sorter systems. Chapter 6 introduces two extensions to scheduling algorithms that could significantly improve the performance of these algorithms at baggage handling sorter systems.

Chapter 7 provides the methodology that is used to determine the performance of the approaches as well as the performance itself. Because many different approaches have to be tested, only the most remarkable results are presented in this chapter.

Finally, Chapter 8 provides the overall conclusions of this thesis and focusses on two subjects. First, it determines whether there is a preferred scheduling approach for specific situations, thereby also assessing the quality of the new scheduling approach and extensions. Second, it assesses whether the direction of the new approach is one that requires further research.

Chapter 2

Process Descriptions

Before developing a new approach for scheduling inbound containers at sorting facilities, we first have to identify the relevant business processes at these facilities. This chapter therefore provides an overview of the processes in and around different types of sorting facilities. Furthermore, it identifies 'common features' that enable us to develop one approach for different types of sorting facilities.

This chapter therefore starts by providing a brief overview of the types of customers that are served by Vanderlande (Section 2.1). This is important, for it is not unlikely that small local companies have other processes than large, international ones.

Next, Section 2.2 and Section 2.3 discuss the relevant business processes in baggage handling and parcel and postal respectively. The processes at distribution centres are not discussed.[1] Main goal is to provide an overview of the environment of a typical baggage handling and parcel and postal company.

Finally, Section 2.4 summarises the results from the previous sections and provides a compact business process model. This model can be used to describe both baggage handling and parcel and postal processes and serves as a normative process model for this research.

2.1 Customer Profile

The customers of the baggage handling department at Vanderlande are usually airports. Based on their size (measured in million passengers per annum; mppa), the location, and the type of passengers that they serve, four different types of airports are identified: small airports (less than 5 mppa), medium

[1] Baggage handling and parcel and postal facilities are designed to sort the incoming items as fast as possible to outfeeds, but distribution centres are usually designed to temporarily store items. According to most literature, this is an entirely different business process indeed.

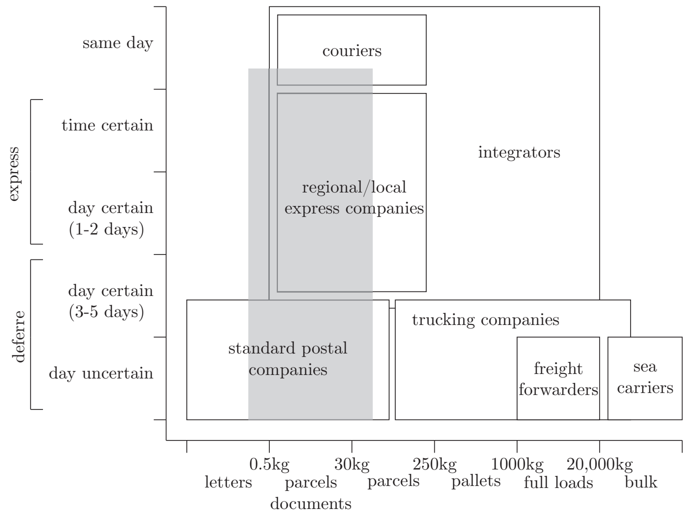


Figure 2.1: Overview of Parcel and Postal Customer Types

sized airports (5–10 mppa), large airports (10–25 mppa), and major transfer hubs (more than 25 mppa). The baggage handling department at Vanderlande, however, has not formulated a formal scope. That is, Vanderlande does not aim at serving a specific type of airport, but is trying to serve them all.

This, however, makes it difficult to determine the typical customer. It is obvious that a small, regional airport such as Eindhoven Airport requires a different system than a large transfer hub such as Amsterdam Airport Schiphol. This often results in a solution where a small, regional airport installs a sorter system that consists of one single sorter, whereas a large transfer hub installs a sorter system that consists of multiple sorters that are interconnected. Whilst discussing the baggage handling processes (Section 2.2), we point to the differences in processes between small, regional airports and large transfer hubs.

In contrast to the baggage handling market, the parcel and postal market is a much more diverse industry. Companies are typified based on three criteria: delivery time, delivery certainty, and weight of the items being shipped. Figure 2.1 provides an overview of the different customer types. The 'scope' of Vanderlande is indicated using a grey box. Generally speaking, we find that the focus is on items between 250g and 50kg and on all delivery options, except same day deliveries by couriers.

Customers of Vanderlande in the parcel and postal industry can thus be grouped into three categories: standard postal companies (e.g. De Post–La Poste and Estonian Post), local a/or regional express companies (e.g. Bartolini, GLS, and GeoPost) and large integrators (e.g. DHL, UPS, and FedEx). The solution to provide each of these categories with a suitable sorter system is similar to the one used in baggage handling, for also in parcel and postal we see that small companies use a sorter system that consists of one small sorter, whereas large integrators use multiple sorters that are interconnected. It is important to realise that many of these companies also ship items that are smaller or significantly larger than Vanderlande's sorter systems can handle. Whilst discussing the processes in Section 2.3, we explicitly mention where and how these items are separated from the others.

Also interesting is that many companies in the parcel and postal market do not own a one single sorter system. In fact, most of them have multiple sorting facilities, each equipped with its own sorter system. This is an important observation, because it could significantly influence the processes at, and transport between, the different sorting facilities. The combined process model developed in Section 2.4 should therefore be able to cope with the differences between these sorting facilities.

2.2 Baggage Handling Processes

Most people that travel by aeroplane perceive only a very limited part of the baggage handling process. They hand over their baggage[2] at the check-in desk, from where a system of conveyors takes over. If they take a close look they might be able to spot their baggage when it is loaded on the aeroplane. Otherwise they will not see it back until they arrive at their destination, where they can retrieve their baggage items at the baggage reclaim belt.

In reality, however, the process is a bit more complex, something that can best be explained using the overview from Figure 2.2. It is important to note that this is a high level overview, i.e., not all processes are shown in detail. Furthermore, the overview is not exhaustive, there are many airports that do not have such an extensive and complicated baggage handling system, whereas others have a system that is far more complex than the one shown. It is, however, necessary to first explain how the baggage arrived at the airport.

After an aeroplane has landed on its destination airport, staff starts unloading the baggage from the aeroplane's hold. If the aeroplane is a wide body aeroplane (i.e. an aeroplane with two or more aisles) baggage is stored in ULDs. Each ULD can hold up to 40 pieces of baggage, depending on the actual size of the baggage items and ULD, which size is dependent on the size of the aeroplane. Usually the baggage has been segregated at the airport of origin. Segregation means that baggage from business class passengers is in another

[2]The terms 'baggage' and 'luggage' are somewhat interchangeable: 'luggage' is the correct term in UK English and 'baggage' is more appropriate in US English. Although the term 'luggage' is preferred by the author, 'baggage' is the industry standard and is used throughout airports all over the world. We therefore decide to use 'baggage' instead of 'luggage'.

ULD than the baggage from tourist class passengers and that *transfer baggage* (i.e. baggage that has not yet reached its destination, but has to continue its journey on another aeroplane) is not mixed with baggage that has arrived on its destination (*reclaim baggage*). If the aeroplane is a narrow body aeroplane (i.e. an aeroplane with six seats or less per row and only one aisle) the baggage is usually loose in the hold (i.e. it is not stored in ULDs) and it is therefore difficult to segregate the baggage. In that case the staff that unloads the baggage is also responsible for separating the transfer and reclaim baggage.

At this point we have ramp carts and ULDs that contain either transfer or reclaim baggage and we can now continue with the overview from Figure 2.2. All the reclaim baggage is transported to and unloaded on dedicated unloading conveyors, which transport the baggage to the reclaim baggage belt in the arrivals terminal (1). This type of baggage is less interesting from an optimization point of view since there are hardly any penalties for delayed reclaim baggage, although especially business and first class passengers might get annoyed if their baggage takes too long to arrive at the reclaim belt. Besides, reclaim baggage hardly requires expensive sorter systems. We therefore decided not to investigate this baggage flow in detail.

In fact, the most interesting flow is the transfer baggage, which has to make its way through the sorting process onto another aeroplane. Because these baggage items have to go through the sorter, they are delayed and mixed with other items. This might cause them to be late for their connecting flight, even if their inbound flight was on time. It is important to note that for each inbound flight detailed information about the transfer baggage is available, i.e. for each bag the connecting flight and final destination are known. If the bags are stored in ULDs, the contents of each specific ULD are also known. There are however very few airports that scan (i.e. identify) the ULDs or individual baggage items when unloading them from an aeroplane. Therefore, the details of the baggage become available only after the baggage has been transported (2) to one of the transfer areas. The identification of ULDs, however, is necessary if we want to develop an unloading schedule that takes the contents of the ULDs into account. It is therefore important to realise that this identification can be implemented relatively easy by checking the identification numbers of the ULDs, when loading them onto the dollies.

A transfer area is defined as an area that contains multiple baggage loading conveyors, which can transport the transfer baggage into the sorter system (3). We decided not to show individual loading conveyors, because it would clutter Figure 2.2 and hardly adds any information. Furthermore, the travel times for individual loading conveyors inside one transfer area are more or less the same. The transfer baggage that is fed onto the loading conveyors is identified using barcode scanners and screened using X-ray machines to identify potential hazardous contents. Due to regulations by the International

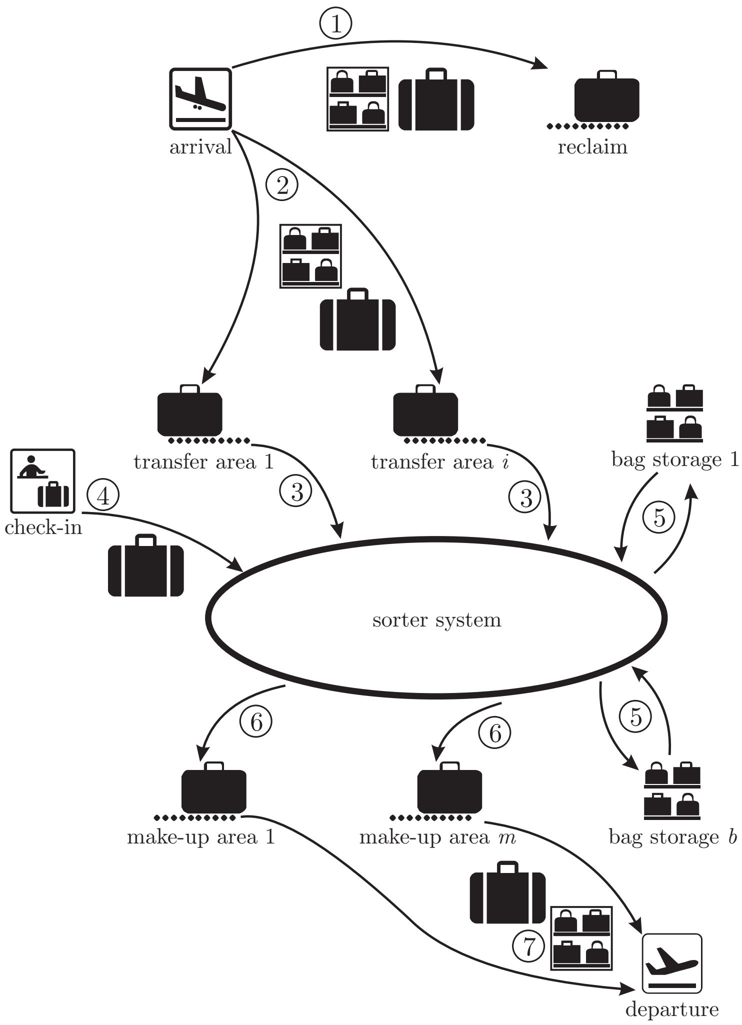


Figure 2.2: Overview of the Baggage Handling Process

Air Transport Association (IATA), all transfer baggage has to be screened, even if it has undergone a security screening at the airport of origin.

At this point the transfer baggage merges with the baggage that has been dropped off by passengers at the check-in desks (4). The check-in baggage is identified by its barcode and is screened using X-ray machines.[3] One important aspect of check-in baggage is that it is difficult to influence the arrival process of the baggage onto the sorter. By influencing the opening and closing times of check-in counters, reducing this uncertainty is possible. However,

[3]Although it is not necessary that hold baggage is screened immediately after check-in, it could for instance be scanned only just before it arrives at the make-up area, there are very few airports that allow unscreened baggage to continue their journey onto their baggage handling system.

one still needs a minimum amount of time to check-in all the passengers and airports are eager to check-in passengers as soon as possible. The earlier a passenger has dropped its baggage items, the more time he/she spends in the shops and catering facilities.

Now all the baggage has arrived in the sorter system, which might consist of different (separate) sorters which are connected using a conveyor system, Destination Coded Vehicles (DCVs), or the same ramp carts and tugs that are used to transport the baggage from and to the aeroplane. At this point in time, bags have been classified as being ‘cold’, ‘normal’, or ‘hot’. ‘Hot’ baggage is baggage that is supposed to be on an aeroplane that departs very soon. The exact definition differs per airport, but generally is between 45 and 20 minutes. ‘Normal’ baggage can be transported directly toward the make-up area, because an outfeed has already been assigned to its flight. Generally this assignment is done three to two hours before departure. Finally ‘cold’ baggage is baggage that is too early to be transported towards the make-up area, because no outfeed has been assigned to its flight yet. This ‘cold’ baggage is therefore transported to one of the baggage storage facilities (5) where it is released back onto the sorter system when the baggage gets the status ‘normal’. Note that not only ‘cold’ but also ‘normal’ baggage may be transferred to the baggage storage area. This is for instance done to store all first and business class baggage, which is then released from the baggage storage area only minutes before the make-up area assigned to this flight closes. This ensures that first and business class baggage is loaded onto the aeroplane last, and thus can be retrieved first when the aeroplane has arrived at its destination. Some baggage storage facilities consist of ordinary conveyor belts, which means that if one bag from the conveyor has to be sent towards the make-up area, all baggage that is downstream of this specific bag also re-enters the sorter system. Nowadays, however, Automatic Storage / Retrieval Systems (ASRSs) that can store and retrieve individual bags are becoming more and more popular.

When the make-up for a flight is ‘open’ one or more laterals or carousels are assigned to handle the baggage for this flight.[4] If a flight gets assigned more than one lateral, which is generally the case with wide body aeroplanes, those laterals are usually close together in order to ensure that further transport towards the aeroplane is efficient. The baggage arrives from the sorter system at one of the make-up areas (6).

When staff takes a bag off the lateral, it is scanned to see whether it is allowed on the aeroplane. That is, the passenger should have checked-in for this particular flight and the bag should have been cleared by security. This

[4]A lateral is a line conveyor which has only a limited capacity and usually can be staffed with only one employee. A carousel is loaded from the top and has a larger capacity, which also allows multiple employees to work at one carousel. In the remainder of this research we use the term lateral to indicate these conveyors in the make-up area, unless a more specific definition is needed.

scan also indicates that the bag has physically left the sorter system, but for the sorter control software, the bag already left the BHS when it was sorted to the correct outfeed. If a bag is loaded into a ULD, staff scans both the baggage item and the ULD so that for each ULD the contents are known. This information, together with passenger data, is used to generate the *manifest*. It is only then that the bags, either on a ramp cart or inside a ULD, are transported towards the aeroplane (7) which is parked at the gate or on a remote aeroplane stand.

Besides the processes discussed above and shown in Figure 2.2, there are also some processes that handle exceptions. A typical sorter system for instance has one or more laterals that are used for bags that are too late (i.e. the lateral that was used for make-up has already been closed), bags which can no longer be identified (e.g. because the label has been smudged or torn), or bags that are 'extremely hot'. The first have to be assigned to another flight to the same destination, the second have to be manually identified and fed into the sorter system, whereas the last could be rushed towards the aeroplane by manually delivering them 'at the tail'. Note that although these should be exceptions, performance criteria that dictate a 0.5‰ missort rate are not uncommon, the worldwide mishandled baggage rate, which contains all bags that have gone missing or did not arrive on their destination together with their owner, was still 11.4‰ in 2009 (SITA, 2010).

2.3 Parcel & Postal Processes

Similar to baggage handling, the parcel and postal process is much more complicated than most consumer clients perceive. Consumers hand over their parcel at the post office or a service point before 5pm and assume that the parcel is delivered the next working day. The majority of the parcels is shipped by business clients, who might have a better understanding of the complexity of the processes, but still demand a parcel to be delivered overnight.

Although in this research we refer to the parcel and postal industry, we mainly focus on the express parcel business. There are three important reasons for this. First, we see that the express parcel volume is increasingly significant, which is mainly driven by the increase in e-commerce. Second, the customers are becoming more demanding with respect to cost-effectiveness, reliability, and delivery times. Finally, we find that postal items (letters etc.) are usually considered to be too small to be carried on standard sorting systems. They therefore have their own, dedicated, sorting systems where they are merged into totes (tote boxes) which *can* be carried on the standard sorting systems. Because we also take the baggage handling process into account, we decided to focus solely on standard sorting systems, which excludes letters, documents, and other non-conveyables.[5]

We do however begin by explaining one of the major differences between the baggage handling process and the parcel and postal process. Where the baggage handling process takes place at a single location (i.e. a single airport) the parcel and postal process is usually distributed over a number of locations, which are 'connected' through a transport network. Although one might argue that, from an airliner point-of-view, the sorting facilities in the aviation industry are also interconnected and thus form a transport network, we do not share this view. In baggage handling, the processes at each of the sorting facilities are exactly the same, whereas in parcel and postal we find that the processes at the different sorting facilities can be significantly different and are highly dependent on the selected network layout. Appendix A introduces the three most important network layouts and explains the layout specific processes for each of them.

Despite the differences in business processes around the network layouts, the processes within sorting centres are actually quite similar. This does however require us to overlook processes that are typical for outbound depots such as weighing, measuring, and (re)labelling. We then find that the processes inside the different sorting centres (outbound depots, hubs, and inbound depots) can be represented using the overview in Figure 2.3.

The first step, which only occurs at air hubs, is the unloading of the ULDs from an aeroplane (1). The ULDs, trailers, or vans are assigned to an unload dock/infeed by the dispatcher (2). This could be done using the relatively simple first-come-first-served (FCFS) concept where a ULD, trailer, or van is assigned to the infeed that becomes available first, although more intelligent approaches are also possible.

At the infeed area, the conveyables are fed into the sorter system (4) whereas the smalls, non-conveyables, etc. are separated (3) and treated manually or using a dedicated sorter system. As mentioned earlier, we do not focus on the treatment of these 'special' parcels. The conveyables are identified using barcode scanners and are sent towards the assigned outfeed (5). If the assigned outfeed is full or other problems arise the parcel recirculates or is sent to the outfeed at the end for parcels that could not be sorted. Similar to the sorter in the baggage handling system, special outfeeds exist for parcels that request a destination that does no longer exist or for no-read errors.

At the outfeed the parcels are loaded into the van, trailer, or ULD (6) and the ULDs are loaded onto the aeroplane (7). It is, however, important to note that aeroplanes, and to a lesser extent trailers and vans, have a departure time that must be met. Parcels that have not been sorted before this cutoff

[5]Non-conveyables are items that cannot be transported on the selected sorter because they are too small, too large, too light, too heavy, too unstable, etc. Whether something is a non-conveyable is thus dependent on the sorter system at hand.

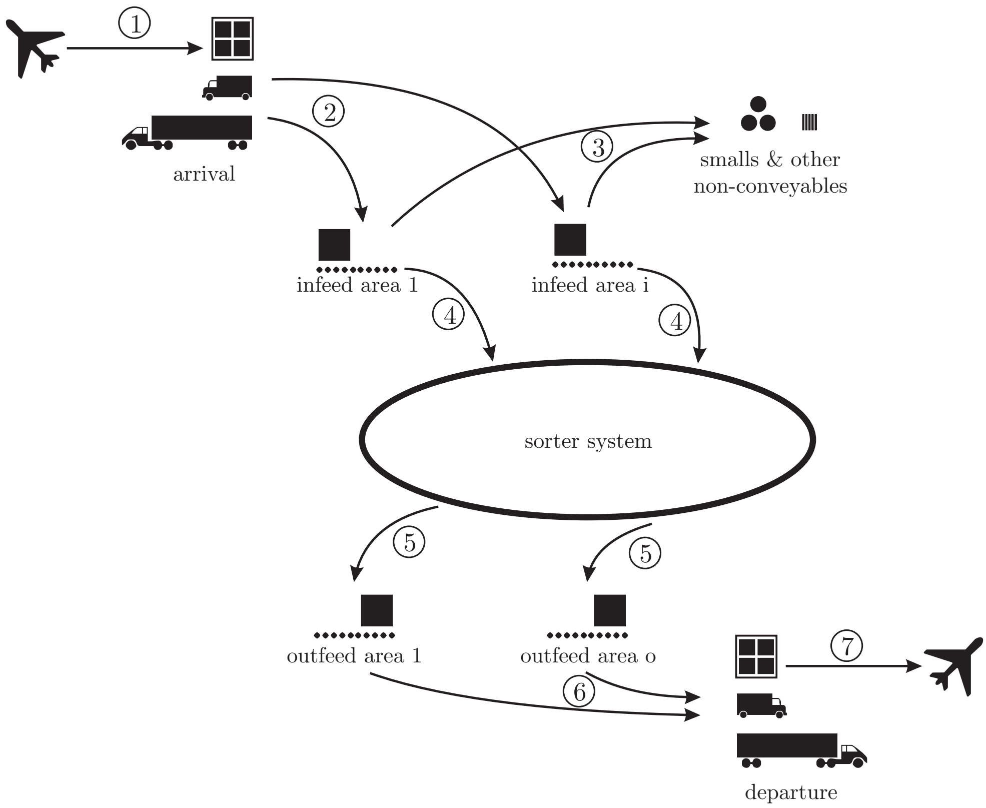


Figure 2.3: Overview of the Parcel & Postal Sorting Process

time will be delayed by a day. Depending on the customer these delays can be really expensive, since many express parcel companies promise overnight delivery.

We thus find that, although the processes for the different network layouts are quite different, it is not difficult to find a common denominator amongst them. Furthermore we see that, if we aggregate to a level that is sufficiently high, the processes within each of the depots and hubs can be modelled similarly.

2.4 Combined Process Model

The two processes from Figures 2.2 and 2.3 have many similarities, but also some differences can be identified. Before presenting the combined process model, we have to determine how to cope with these differences. Generally speaking, three different approaches can be used: disregarding, converting, or incorporating. When a difference is disregarded, it is not incorporated in the model at all. In fact, it indicates that the difference is so insignificant that a model holds without. A difference that is converted is also not directly incorporated in the model. However, in this case an alternative technique is applied to include the effects of the difference. Finally when a difference is incorporated, the possibility of this difference is added to the model. As a result, this part of the model may or may not be used, depending on the problem at hand.

First, in baggage handling systems there is a flow of reclaim baggage (1, Figure 2.2) which is not present in the parcel and postal industry. Or, more generally, in baggage handling there is a flow of items that do not have to be processed by the sorter system, whereas in parcel and postal all items have to be processed. We decided to disregard this difference, because the reclaim baggage is segregated from the transfer baggage, as stated in Section 2.2. As a result, we only model the flow of goods that have to be sorted (2, Figure 2.2 and Figure 2.3).

Second, the unloading and loading of ULDs from and in an aeroplane (1 and 7, Figure 2.3) is something that is not separately modelled in baggage handling. Reason is that in baggage handling all items have to be (un)loaded in two steps,[6] whereas in parcel and postal only interhub air transport requires the two step approach. That is, in parcel and postal most items are (un)loaded directly from / onto the trucks, which is a one step approach. In this case, we decided to apply the converting approach. Instead of explicitly modelling this (un)loading of ULDs, we only model the (un)loading of items from and in the ULDs. The perceived behaviour of air transport, all ULDs transported by an aeroplane arrive at the same time and are required to be loaded before departure, can be modelled quite easily, by assigning the same start up and cutoff times to all ULDs that are (un)loaded from / in a specific aeroplane.

Third, in baggage handling systems there is an uncontrollable inflow of items to the sorter, originating from the check-in desks (4, Figure 2.2). These check-in desks are obviously missing in parcel and postal sorter systems, but, more importantly, there is no similar flow of items present. The uncontrollable inflow could, however, severely affect the performance of the sorter system and it is therefore undesirable to disregard or convert this difference. We therefore decided to incorporate this difference in the process model. By setting the inflow equal to zero, a parcel and postal sorter system can be mimicked quite easily.

Last, the baggage handling systems often provide temporary storage for ‘cold’ baggage, baggage segregation, etc. (5, Figure 2.2). In parcel and postal these temporary storage facilities are extremely rare, if they exist at all. The need for temporary storage in baggage handling is because one make-up conveyor (*outfeed*) is assigned to multiple flights during the day, whereas in parcel

[6] When loading, items are loaded in ULDs and the ULDs are loaded in the aeroplane. Similarly, when unloading ULDs are first unloaded from the aeroplane, after which the baggage items are unloaded from the ULDs.

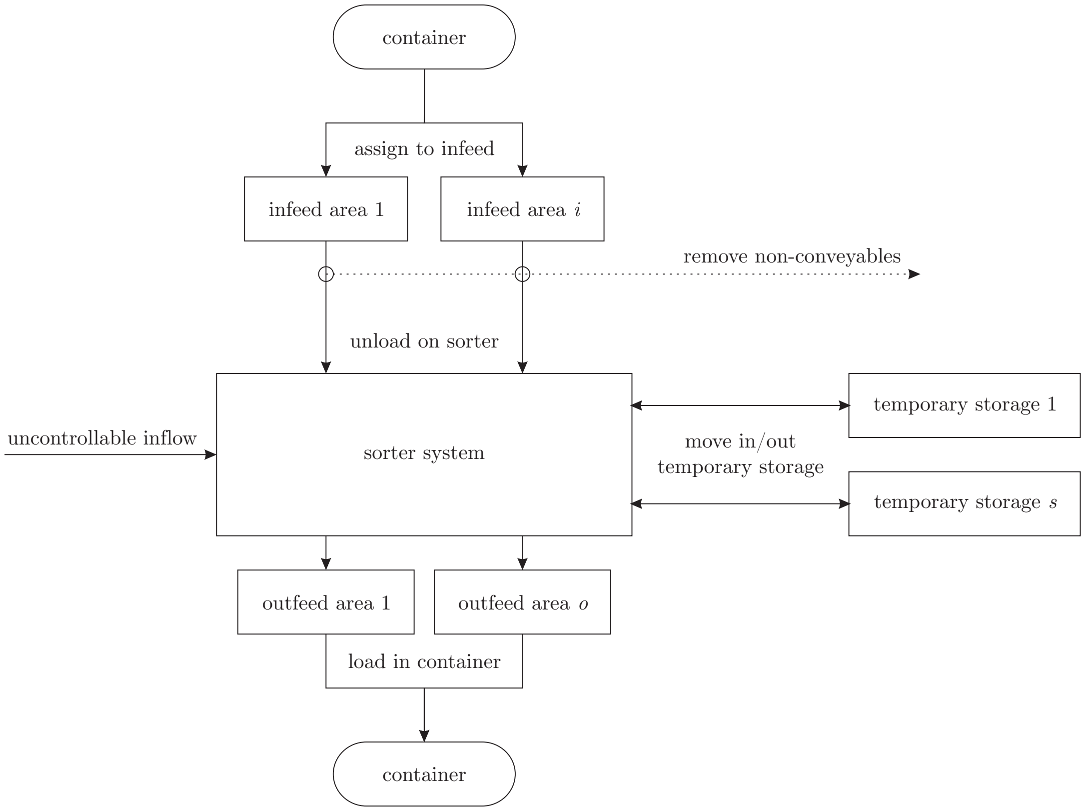


Figure 2.4: Combined Process Model

and postal an outfeed is usually assigned to a single destination during the entire shift. As a result, items in parcel and postal can always be assigned to their outfeeds, while in baggage handling this is not always the case. Because items that are not yet assigned to an outfeed can clutter the sorters, due to recirculation, there is a strong need for temporary storage facilities. For exactly the same reasons it is important to model the temporary storage facilities, for otherwise the simulation model might clutter. The temporary storage facility is therefore incorporated in the combined model. Parcel and postal sorter systems can then be modelled, quite easily, by assuming a zero capacity storage.

Figure 2.4 shows the combined process model, including the adaptations listed above. When a container arrives, it is assigned to an infeed conveyor in one of the infeed areas. The non-conveyables, if present, are removed from the container during the unload process and the conveyables are loaded in the sorter system. An uncontrollable inflow of conveyables may be present, these items are fed directly in the sorter system. Furthermore one or more temporary storage facilities may be present. These storage facilities can be used to store items that cannot yet be delivered or are requested to arrive at a later time. Finally the items arrive at one of the outfeed conveyors in one of the outfeed areas, after which they are loaded into a container.

It is interesting to see which approaches and techniques have already been developed to solve similar problems. Chapter 3 therefore provides an overview of literature that focusses on inbound container assignment, and other, related, problems. Special attention is paid to the algorithms and approaches that are used to solve these problems.

Chapter 3

Literature Review

Existing literature is always a good starting point when developing a new theory, algorithm, or methodology. This chapter therefore provides an overview of literature related to the problem described in Section 1.2. In addition, focus is on literature that describes processes that can be compared to the ones described in Chapter 2.

Chapter 1 defined a rather abstract and broad research question: *'what knowledge can be obtained from existing literature?'* However, in order to provide clear answers, more specific research questions are required. Therefore this introduction formulates two objectives that should be attained by the end of this literature review.

The first objective is to determine whether similar research has been done before, and, more importantly, what solutions these studies applied a/or developed. Algorithms and approaches developed by other researchers might aid in developing a new scheduling/assignment algorithm for the problem at hand. The corresponding research question is: what solution(s) does literature provide for similar problems?

The second objective is to find building blocks from approaches and algorithms that could easily be integrated in other existing or newly developed approaches. These 'scraps of code' may, for instance, contain easy container selection algorithms or workload balancing techniques. Using these 'extensions' or 'components', the probability of re-inventing the wheel is reduced. Summarised, the research question is: what extensions and components can be found in literature?

Instead of sequentially discussing the research questions defined above, this chapter is organised in a more thematic way. Section 3.1 focusses on literature related to a specific problem, which has been defined a couple of years ago. This problem is selected as the start of this chapter, because its main features are similar to the problem addressed in this research.

The problem that Section 3.1 introduces, is actually a problem from the parcel and postal industry. Section 3.2 therefore discusses other relevant literature from this industry, although literature from the distribution industry is also included. After all, inbound trailer assignment problems in parcel and postal do not, at least not by definition, differ much from inbound trailer assignment problems in distribution.

Section 3.3 discusses relevant literature from the baggage handling industry. It is important to review literature from this industry, because some problems, e.g. the sequential assignment of different destinations to one outfeed, are typical for the aviation industry.

Section 3.4 concludes this chapter, and therefore answers the research questions defined above.

3.1 Parcel Hub Scheduling Problem

The 'parcel hub scheduling problem' (PHSP), introduced by McWilliams, Stanfield, and Geiger (2005), is one of the first problems that focusses solely on the scheduling of inbound trailers. In this paper, but also in later research, they use a fairly simple parcel sorting hub with three unloading docks and 9 loading docks (see Figure 3.1). Each unloading dock ($U1, U2, U3$) feeds the parcels onto a presorter (or primary sorter; $P1, P2, P3$), which directs the parcels to one of the three mainline conveyors. Near the loading docks ($L1, \ldots, L9$), the parcels on each mainline conveyor are sorted into three different chutes by the secondary sorters ($S1, S2, S3$), after which the parcels are loaded in a waiting trailer.

The attempt of McWilliams et al. (2005) consists of an overly simplified system in which they try to minimise the *makespan* of the sorting process. That is, they want to finish the sorting process as quickly as possible. Because these types of scheduling problems are very hard to solve, they use a simulation-based scheduling algorithm (SBSA), which is based on a genetic algorithm (GA), to solve the problem. They show that their approach is superior to the arbitrary scheduling (ARB) approach, which, according to McWilliams et al. (2005), was the industry standard at that time. Note that in the same year, McWilliams (2005) showed that similar results could be achieved using iterative local search or simulated annealing techniques.

One of the drawbacks of the approach of McWilliams et al. (2005) is that only equal batch sizes (i.e. full-truckloads; FTLs) are considered for the inbound trailers. In a later paper, McWilliams, Stanfield, and Geiger (2008) therefore relax this assumption and show that the size of the batches should be considered when developing scheduling algorithms for these problems. Unfortunately, their new approach requires much more computational effort. They

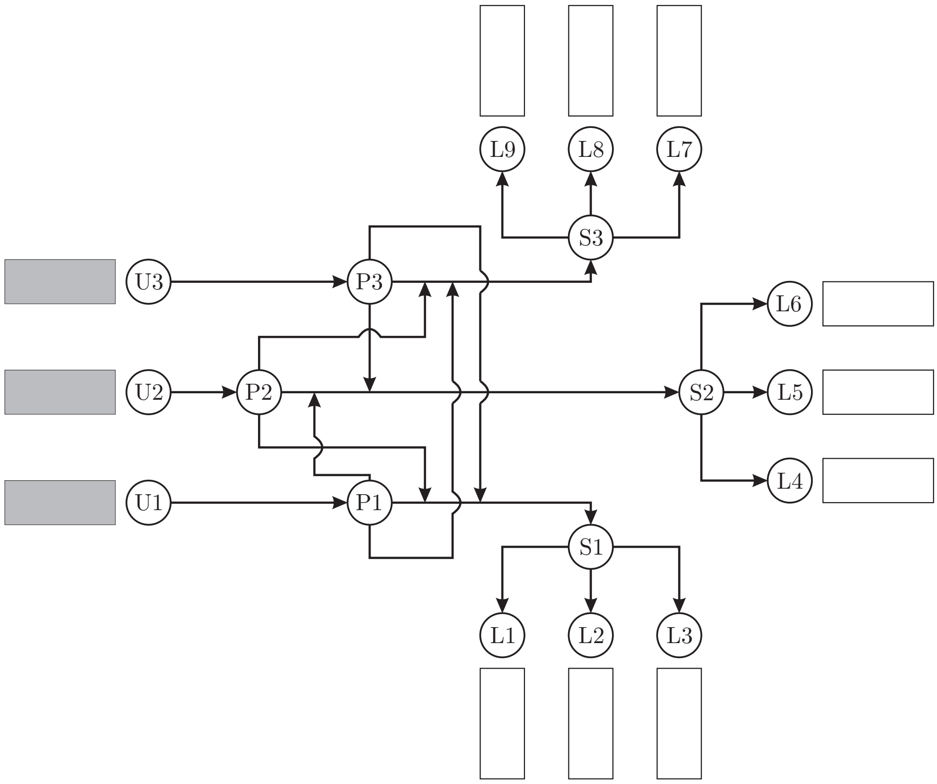


Figure 3.1: Simple Parcel Consolidation Terminal (McWilliams et al., 2005)

therefore advise to continue their research in order to find faster algorithms, which could thus be applied more easily in reality.

One of the problems that can be associated with finding solutions that minimise the makespan, is that this can result in unequal workloads at the secondary sorters and loading docks. McWilliams (2009a) therefore continued the research, aiming for an approach to balance the workload on the loading docks. In order to solve small problems to optimality, he formulated a binary minimax programming model. However, he recognised that for larger and more realistic problems, solving to optimality would take too long. McWilliams therefore resorted to using a genetic algorithm. Obviously this new approach requires much less time, because it does not need the computationally expensive simulations. Furthermore, it proved to outperform the SBSA and ARB approaches used in McWilliams et al. (2005). Major drawback of this approach, however, is that due to the minimax problem there may exist many optimal solutions in a very large non-convex solution space. The question that arises is whether genetic algorithms are indeed the preferred solution approach or that better approaches might exist. McWilliams (2010) shows in further research that iterative approaches, such as simulated annealing and local search, provide solutions that are on average 6% better than the solutions provided by the genetic algorithm, although for large problems more time is needed to find these solutions.

Recently, McWilliams (2009b) developed a relatively simple dynamic load balancing algorithm (DLBA), which appears to be a promising approach. Where the other algorithms require information on all trailers in a particular shift, this algorithm only requires knowledge of the trailers that are waiting to be assigned to an unloading dock. In that sense, it is already a relaxation of his assumption that all unloading trailers are available at the beginning of the shift, although he does not formulate it that way. He finds that this extremely simple algorithm performs much better than random assignments (makespan reduction of 15%). Furthermore, it appears that this DLBA is better (makespan reduction of 8%) in large complex problems than the approach from McWilliams (2010), although the latter approach performs better in smaller problem instances (makespan reduction of 2%). It is, however, important to note that McWilliams (2009b, but also in previous work) defines a number of restrictive assumptions,[1] which limit the practical application of the DLBA. It is somewhat remarkable that McWilliams does not report the performance with respect to load balancing, although this was his primary objective. The most likely explanation for him failing to do so, is that McWilliams argues that makespan and load balancing are interdependent and optimisation of makespan therefore results in an optimally balanced workload.

If we summarise the results from the work that has been done by McWilliams et al. (2005, 2008) and McWilliams (2009a, 2009b, 2010) on the subject of the parcel hub scheduling problem, we conclude that a reduction in makespan by 15% (compared to arbitrary scheduling) appears to be feasible. In fact, this result is achieved using the DLBA, a very simple approach.

Besides the work that has been done by McWilliams et al. there has been a lot of research on the subject of scheduling parcel and postal processes. Other literature might not only provide us with another point of view on the matter, it could also give directions for 'better' scheduling algorithms. It would especially be interesting to find solutions for the restrictive assumptions that have been made by McWilliams. This chapter therefore continues with a review of relevant literature from the parcel and postal industry.

3.2 Parcel & Postal Literature Review

Probably one of the most studied subjects in the distribution, parcel, and postal (DPP) field is the concept of crossdocking. A crossdock is generally

[1] E.g. concerning the arrival and departure processes and (un)loading speed. A full overview can be found on page 36.

defined as a facility in which inbound trailers are unloaded, their contents immediately sorted and shipped to outbound trailers. One of the most important features of a crossdock is the lack of intermediate storage space, which significantly reduces the required floorspace. This, combined with the ability to consolidate less-than-truckloads (LTLs), which reduces transport costs, makes that crossdocks are quite popular. Wal-Mart, for instance, uses the crossdocking concept and has thus eliminated all inventory points but one, the shop itself (McInerney & White, 1995).

Cohen and Keren (2009) state that: *'any receiving and shipping system that uses a conveyor as a major freight moving tool ceases to be a crossdock'*, which suggests that we cannot use the techniques that are developed for crossdocking in our problem. This, however, is not necessarily true. Cohen and Keren probably wrote this with a crossdocking facility in mind where the doors can be assigned to either inbound or outbound trailers. In that case their definition holds, since a conveyor system forces the crossdocking facility to have dedicated unloading and loading docks.[2] Furthermore they might have a point, since in a crossdock the freight is transported directly from an unload dock towards a load dock using e.g. forklifts, whereas in a conveyor based crossdock freight would have to travel on the conveyor system until it is sent towards the correct outfeed. In that sense the scheduling algorithms used might indeed be *'totally different'* (Cohen & Keren, 2009), which is also the reason why the algorithm provided by Cohen and Keren is not suitable for our problem.

Interesting is that, although a crossdock is defined as a no-inventory sorting facility, there is quite a lot of literature that explicitly uses temporary storage. Li, Low, Shakeri, and Lim (2009) consider the situation in which the floorspace in the centre of the facility is used to store products that cannot be loaded yet, in fact they assume infinite storage space. Although the problem that is discussed by Li et al. is different from the problem we have at hand, in their problem each inbound trailer is also an outbound trailer that has to be loaded directly after it has been unloaded, they use an interesting heuristic, which is based on the well-known parallel uniform scheduling problem.

The work by Yu and Egbelu (2008) is perhaps even more interesting, since it explicitly focusses on how one should cope with the possibilities of limited intermediate storage.[3] Yu and Egbelu aim at scheduling the inbound and outbound operations of a crossdock in such a way that the makespan of the operation is minimised. They provide both a mathematical model to solve the scheduling problem to optimality and a quite extensive heuristic algorithm. The mathematical model is used to determine the optimal solution for relatively simple problems, which can then be used to assess the performance of

[2] In theory it might be possible to have an infeed and outfeed for every door, but this would be an expensive and impractical solution.

[3] Where limited refers to the fact that all goods that are stored during a shift, should also be retrieved in the same shift, i.e. at the end of a shift the crossdock is empty.

the heuristic. Yu and Egbelu show that their heuristic performs best if both outbound and inbound trailers are selected based on the amount of temporary storage space that is required. Unfortunately, similar to the DLBA by McWilliams (2009b), the approach by Yu and Egbelu requires quite a number of restrictive assumptions. That is, all trailers are assumed to be available at the start of the operation, there is infinite storage space, the unloading sequence of products from an inbound trailer can be determined, etc. It shows, again, that although the ideas behind existing scheduling algorithms are good, there still are a number of assumptions that limit the use of these algorithms in practice.

A scheduling algorithm that has been tested using a realistic case is the Bin and Rack Assignment Model (BRAM) by McAree, Bodin, and Ball (2002). This algorithm was specifically designed for air terminals where inbound ULDs are assigned to bins to be broken into individual pallets. These pallets are then sorted and transported using forklifts and built up to ULDs again at rack locations. The main goal of McAree et al. is to minimize the operational cost, which consists of total distance travelled and the number of forklifts that is necessary for the operation. Because the BRAM is too complex to solve, even for small instances, they developed a new algorithm that finds a solution by iteratively solving the Bin Assignment Model (BAM) and Rack Assignment Model (RAM), both of which are mixed integer programs (MIPs). McAree et al. noticed that for large and realistic problems the BAM and RAM algorithm still required too much computational effort. They therefore developed the Aggregated BRAM and, subsequently, ABAM and ARAM algorithm. Here bins are aggregated into superbins and in a similar way racks are aggregated into superracks. This significantly reduces the solution space of the problem. The newly developed ABAM and ARAM algorithm were applied to different facility layouts that were under consideration by FedEx (McAree, Bodin, Ball, & Segars, 2006). McAree et al. (2006) showed that there can be a significant difference in total distance travelled and the required number of forklifts between different layouts. For us, the approach by McAree et al. (2002, 2006) is especially interesting because it shows that dividing an extremely complicated problem into two much less complicated subproblems, enables us to solve complicated problems with millions of variables and hundreds of thousands of constraints in a reasonable amount of time.[4]

Gue (1999) tackles, although inadvertently, another problem found in the solutions by McWilliams (2009b) and Yu and Egbelu (2008). Gue aims at determining which doors should be unload docks and which should be load docks

[4] A 'reasonable amount of time' is a relative concept and is strongly dependent on the purpose of the algorithm. McAree et al. (2006) find solutions for the different layouts in 8–2152 minutes, which for large scale investment decisions is actually quite fast (but would be too slow for operational scheduling decisions).

in a true crossdock facility,[5] using a somewhat unconventional algorithm for selecting trailers from the queue. As Gue states: *'the most obvious scheduling rule is to assign to an open strip door that trailer in the queue having the lowest travel cost (...) for the open door'*, which is more or less what both McWilliams and Yu and Egbelu do. These approaches however do have a major drawback, as recognised by Gue: *'these rules could cause particularly high-cost trailers to be stuck in the queue for a long period of time'*. He therefore proposes an alternative selection algorithm. First one should determine for each inbound trailer i the unload docks that would yield the kth lowest 'cost'. The inbound trailers are sorted based on arrival time in a waiting list. If an unload dock j becomes available, the dispatcher scans the waiting list and selects the first trailer for which unload dock j is the cheapest solution (i.e. unload dock j has the 1st lowest cost). If no such trailer exists, the dispatcher scans the queue again and selects the first trailer for which unload dock j is the second cheapest solution, etc. until a trailer has been selected. This simple algorithm prevents that high-cost trailers are not selected because it does not select the cheapest solution for the idle unload dock, but selects the trailer that has been waiting the longest for which the idle unload dock is the cheapest solution.

It is important to realise that all previously mentioned literature focusses on dock-door or similar assignment problems. The work by Werners and Wülfing (2010) is especially interesting because they consider a more complicated sorter system. In their model of a Deutsche Post parcel sorting centre, each parcel is unloaded at an unloading dock, sorted into a chute and then assigned to a loading dock. Especially the last two steps are interesting, since the number of chutes is larger than the number of loading docks. This can be quite realistic since loading the trailer requires only a fraction of the time that is required for the entire sorting process and using this layout one would make better use of the available space. Werners and Wülfing aim at minimizing the total transport effort, i.e. they want to reduce the total distance travelled on the (pre)sorters. In order to solve this large complex problem they hierarchically decompose the problem into two subproblems. The top level assigns destinations and tours (each tour consists of the parcels for different destinations) to specific areas in the sorting centre in such a way that the workload is balanced and the total distance travelled is minimized. On the lower level each destination is assigned to a specific chute and the tours are assigned to a specific dock door, both limited by the aggregated area the destination and tour was assigned to earlier. Werners and Wülfing show that their approach ensures a balanced workload over the different areas in the sorting centre, whilst providing robust solutions. Werners and Wülfing, how-

[5] A *true* crossdock is defined by Gue (1999) as a crossdock in which all doors can be either unload or load docks. During operations, however, the assignment of doors is fixed.

ever, do not discuss the unloading process, they focus solely on scheduling the outbound processes.

Summarised, literature in the parcel and postal field provides some interesting leads for further research. Yu and Egbelu (2008) provide an interesting approach for coping with the possibility of intermediate storage, which could be useful for modelling bag storage solutions. The trailer selection mechanism developed by Gue (1999) could be a very useful alternative to the trailer selection algorithm developed by McWilliams (2009b). Furthermore, the hierarchical decomposition proposed by both McAree et al. (2006) and Werners and Wülfing (2010) could be a good approach to simplify the optimisation problems that are inevitably part of finding an 'optimal' solution.

It is also important to review literature that has been written in the other field of interest, baggage handling. Particularly commonalities and contraries are interesting, since they give a good indication where synergy can or cannot be found. It also is highly likely that, due to continuously changing schedules, literature in baggage handling is much more focussed on dynamic algorithms for short-term operational planning, whereas parcel and postal literature appeared to focus mainly on long-term tactical and strategic planning.

3.3 Baggage Handling Literature Review

One of the earlier authors that provide an overview of literature in the field of airport operational management is Tošić (1992). Interesting is that most literature, in his overview, focusses on terminal building design and modelling. Literature that deals with baggage handling is actually quite rare, and focusses primarily on reducing the waiting queues for passengers both at check-in desks and reclaim belts.

Tošić (1992), however, also refers to Robusté and Daganzo (1992) who focus on *'presorting strategies for containerized transfer baggage that would decrease handling costs'* (Tošić, 1992).[6] In fact Robusté and Daganzo provide an extensive overview of the possibilities of presorting strategies, whilst aiming at minimising baggage handling costs. They therefore model the baggage handling process in detail, by specifying for each strategy the number of moves (for each bag, staff member, container, etc.) and determining the resulting costs of this strategy. The final conclusion by Robusté and Daganzo is that airlines could achieve significant cost reductions if they would segregate the baggage for the larger destinations at the origin airport. In fact, it is this work

[6] The doubt that speaks from this quote is probably due to the fact that Tošić has not been able to study the work properly, given that his paper was published in the same year Robusté and Daganzo published their results.

that allows us to intelligently assign ULDs to infeeds, because they propose a model in which the baggage for different destinations is not homogeneously distributed over the different ULDs.

The subject that more recently received quite a lot of attention, is the outbound assignment problem, i.e. assigning outfeeds (make-up areas) to specific flights. Abdelghany, Abdelghany, and Narasimhan (2006), for instance, propose a relatively simple algorithm for dynamic pier (make-up area) assignment in a congested airport. From all available piers, they select the pier that matches the requirements for the flight under consideration, where they prefer piers that have already been used. The latter ensures that only a minimum number of piers is used, which reduces the movement of staff from pier to pier. Furthermore they use information on the location of the different piers and plane stands, thus enabling them to select a pier that is close to the expected location of the plane. Abdelghany et al. also show that if more than one flight is assigned to a specific pier, the performance, in this case distribution of idle time over the different piers, improves. The most interesting part about the work of Abdelghany et al. is that it is an extremely simple dynamic algorithm, which can easily be adapted for different objectives or more difficult operational processes.

Frey, Artigues, Kolisch, and Lopez (2010) apply a much more mathematical approach for a 'baggage handling system scheduling problem'. They consider a baggage handling facility with an early-bag storage system, which catches all bags when the flight has not yet been opened. Furthermore they have multiple make-up carousels each with a number of workstations and a number of parking spaces for ramp carts. The main goal is to balance the workload over the different carousels and workstations by determining at what time the bag storage system should start sending bags to the carousels. Because assigning flights to workstations and carousels *and* determining when to send the bags from the bag storage is too complicated to solve, a problem with eight flights and one carousel with eight workstations takes an hour to solve, they decided to decompose the problem into smaller subproblems. First, they assign each flight to a carousel (circulation) in order to minimise the maximum workload per carousel, assuming that baggage is retrieved from the bag storage as soon as the flight opens. Next, a scheduling problem is solved in which each workstation is assigned to one or more flights and where the actual start of retrieving bags from the baggage storage is determined. If a solution violates the storage or workstation capacity, additional constraints are added to the problem until a feasible solution has been found. Finally, the remaining workstations are assigned to flights by solving a minimum cost flow problem, after which a final scheduling problem is solved. Given the runtime of the algorithm by Frey et al., we conclude that this approach is more useful for tactical planning (e.g. creating a schedule for the hub of a major airline that has the same flights everyday) then for operational planning, which should take last minute changes into account.

Cheng (1997) focusses on the gate assignment problem instead. In literature, we find two common assignment techniques: parallel, in which all trailers are assigned to a dock or flights are assigned to a carousel; and sequential where trailers or flights are assigned one by one. Cheng decided to apply a different approach: he divides the entire planning horizon in t different time buckets and for all the planes arriving within one time bucket he applies parallel assignment. This partial parallel assignment method therefore comes down to a simple assignment problem, which can easily be solved using Mack's method (Mack, 1969) or the Hungarian method (Kuhn, 1955; Munkres, 1957). Furthermore, he shows that the problem can easily be adapted to incorporate expert knowledge such as *'assigning aircraft X to gate Y is better in situation Z'* (Cheng, 1997).

Albeit not entirely related to the scheduling and assignment literature, an interesting approach to determine the 'urgency' of a container can be found in the work by Hallenborg (2007). Although he focusses on how to use agent based scheduling in DCV baggage handling facilities, the urgency equation for a bag could easily be applied to determine the urgency of a container containing multiple bags. Generally speaking Hallenborg determines a fixed value U_t at which point a bag becomes urgent and U_{max} is the time until the make-up lateral closes. Now if the total travel time until the DCV arrives at its destination is t, the urgency u of the bag is determined using the following equation:

$$u = \begin{cases} \frac{1}{t^2} & t < U_t \\ \frac{1}{(U_{max}-U_t)^2}\left(-t^2 + 2U_t \cdot t - U_t^2\right) & t \geq U_t \end{cases} \tag{3.1}$$

The result is a bag that has urgency -1 at time U_{max}. When the time until cutoff decreases, the urgency increases with a decreasing rate until urgency is 0 at time U_t. From that time on the bag becomes a rush bag and is considered to be urgent, the urgency increases with increasing rate until it is infinity when the make-up area closes, i.e. when $t = 0$. This approach thus provides an easy solution to determine which containers need to be unloaded first in order to ensure that their contents are indeed sorted before the outfeeds close.

Summarised, baggage handling literature does not only confirm the results from the parcel and postal literature, it also provides additional approaches. Similar to what we found in parcel and postal literature, we see that both Cheng (1997) and Frey et al. (2010) use decomposition to find solutions for complicated problems. It is important to note that we can identify two types of decomposition, Cheng applies a temporal decomposition, whereas Frey et al. uses a hierarchical one. Another important, and interesting, difference is that

Cheng uses a quite simple algorithm to solve the gate assignment problem. Frey et al., on the other hand, uses a very complicated and extensive algorithm. Similar to Cheng, Abdelghany et al. (2006) use a very simple algorithm for the pier assignment problem. What may be the most interesting conclusion to be derived from Abdelghany et al. and Cheng, is that a simple algorithm might already improve performance significantly. Finally, we may conclude that the urgency equation of Hallenborg (2007), especially in combination with the trailer selection mechanism by Gue (1999), shows that attention should be paid to how the urgency of a container is determined.

3.4 Conclusion

Using the results from the literature review, we can now answer the two research questions defined earlier:

1. What solution(s) does literature provide for similar problems?
2. What extensions and components can be found in literature?

From the literature review we conclude that there is not an instant solution to solve the problem at hand, which is hardly a surprise. Literature does, however, provide two interesting approaches that can be used to solve similar problems.

The parcel hub scheduling problem (McWilliams et al., 2005), for instance, can be considered a simplified instance of the problem that is the centre of our research. Solutions to the PHSP could thus be used as a starting point for approaches that may solve our problem. Especially the Dynamic Load Balancing Algorithm by McWilliams (2009b) appears to be an interesting and useful approach for the development of a more generic load balancing algorithm. Not only is the DLBA a relatively simple and fast approach, McWilliams (2009b) also showed that the algorithm can result in a 15% reduction in makespan when compared to an arbitrary scheduling approach. Unfortunately, the DLBA does have its drawbacks. That is, some of the restrictive assumptions severely reduce the applicability of the approach in realistic problems.

The baggage handling system scheduling problem (Frey et al., 2010) is considered to be a problem similar to the one we have at hand. Although Frey et al. aim at scheduling the make-up areas and the time at which the baggage storage system starts retrieving baggage, the problem could easily be converted into a scheduling problem for inbound containers. Unfortunately, this approach has two major drawbacks. First, the assumption that full knowledge, which is required for scheduling and assignment problems, is available; and second, the runtime of the algorithm, which is too long for dynamic operational planning.

We may therefore conclude that the DLBA by McWilliams (2009b) is a better, more suitable, approach for the problem at hand than the approach by Frey et al. (2010). The DLBA is therefore selected as the 'state-of-the-art' approach, and serves as a foundation for the development of new scheduling techniques.

As it appears, literature provides only a limited number of solution approaches for similar, but simplified, problems. Fortunately, literature frequently contains useful extensions a/or components. We would therefore like to emphasize four useful extensions and components that can easily be integrated in other approaches:

1. The 'intermediate storage solution' by Yu and Egbelu (2008). Although the full model is quite complicated, the results of their research are actually quite clear: select those inbound/outbound trailers that require the least intermediate storage.
2. The 'trailer selection algorithm' by Gue (1999). Where many researchers choose the, admittedly intuitive, approach of selecting the best trailer for the available door, he decided to select the trailer for which the available door scores best combined with FCFS. As a result, trailers that wait for a very long time before being served are history.
3. The 'pier staff reduction approach' by Abdelghany et al. (2006). By simply giving priority to piers (make-up belts) that have been used before when selecting piers for departing flights, they reduce the required staff and reduce the need to move to different piers continuously.
4. The 'urgency equation' by Hallenborg (2007). It is not so much this specific equation that is interesting, but the more general concept of introducing an intelligent method to determine which containers are urgent. Perhaps even more important, how to deal with an urgent container?

The next chapter focusses on three existing scheduling approaches that serve as the 'current position' for inbound container problems. Besides the current practice, which has already been introduced in previous chapters, also a academic solution and the 'state-of-the-art' from literature, the dynamic load balancing algorithm by McWilliams (2009b), are discussed.

Chapter **4**

Current Practice & State-of-the-Art

This chapter discusses three important scheduling approaches for inbound operations at sorter systems: arbitrary scheduling, first-come-first-served, and the dynamic load balancing algorithm. Each section focusses on one of the scheduling approaches and successively deals with the underlying principles, the algorithm itself, and implementation issues. To preserve conciseness, this chapter provides only the flowcharts of the approaches. The corresponding pseudocodes can be found in Appendix B.

Section 4.1 introduces arbitrary scheduling (ARB), which is an academic solution to the scheduling problem. It is therefore not so much an algorithm that would be applied in reality, but serves as a benchmark to check whether other approaches are better than a random technique.

Section 4.2 discusses the current practice in both the baggage handling and parcel and postal industry, where the first container that arrives is the first to be unloaded. This approach is especially interesting, because approaches that are unable to outperform first-come-first-served (FCFS) are unlikely to be used in reality.

Finally, Section 4.3 focusses on the dynamic load balancing algorithm (DLBA) by McWilliams (2009b). Besides discussing the concepts behind the approach and algorithm itself, also the parcel hub scheduling problem (PHSP), which is the foundation of the DLBA, is discussed. The DLBA serves as the state-of-the-art approach from literature, as it is one of the most recent dynamic optimisation approaches.

4.1 Arbitrary Selection

The arbitrary scheduling approach (ARB) is neither current practice nor state-of-the-art. It is even very unlikely that such an approach would ever be adop-

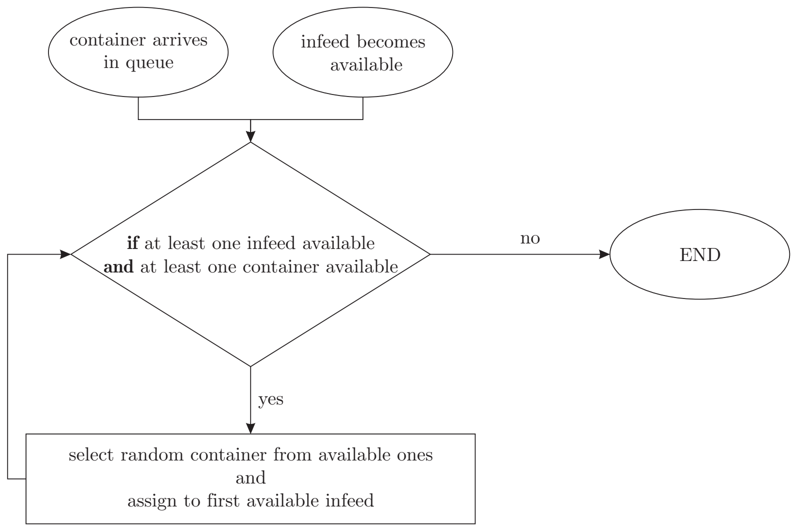


Figure 4.1: Flowchart for Arbitrary Scheduling

ted in reality, as the resulting assignment of containers to infeeds would not be intuitive. One might therefore argue that including this approach in the research is of little use. The arbitrary scheduling approach, however, is very interesting from a benchmarking perspective. Because arbitrary scheduling randomly selects a container for an available infeed, it can be considered a 'dumb' approach. All approaches that use some intelligence when selecting containers should therefore outperform this approach. Approaches that are unable to outperform arbitrary scheduling are thus considered to be unsuitable. After all, selecting a random container would be a better approach.

Algorithm

The arbitrary scheduling algorithm (Figure 4.1) is a very simple approach. It is designed in such a way that there are two events that trigger the algorithm: the arrival of a container in the queue and an infeed that finishes unloading a container. In both cases the algorithm first checks whether two prerequisites are met: there should be at least one container available and at least one idle infeed. Although these checks might be superfluous, they allow a single algorithm to handle both triggers. If both prerequisites are met, the algorithm selects a random container from the queue of waiting containers and assigns this container to the first available infeed. That is, starting with the infeed that is closest to the outfeeds, the algorithm sequentially visits each infeed. If an infeed is available, the container is assigned. The algorithm does not search for the best infeed for the selected container, even if multiple infeeds are available, for each infeed is as good as another.

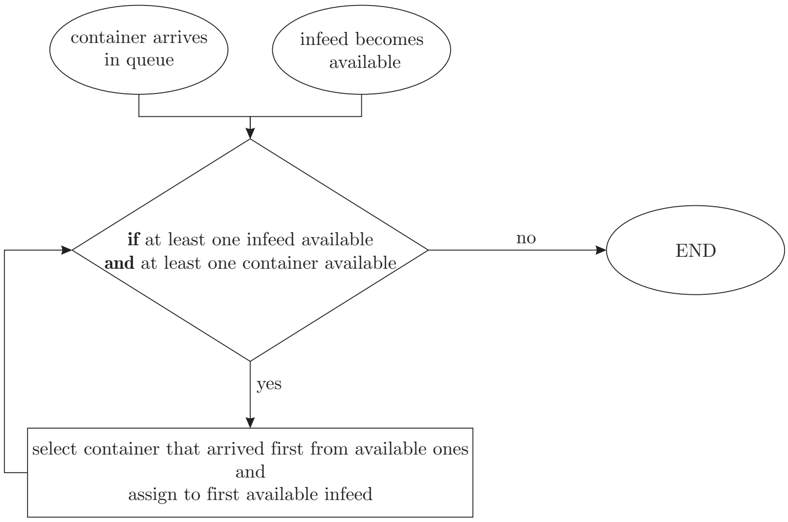


Figure 4.2: Flowchart for First-Come-First-Served Scheduling

4.2 First-Come-First-Served

The first-come-first-served scheduling algorithm corresponds to current practice, both in parcel and postal and baggage handling. In parcel and postal, the dispatcher will usually send the first trailer in the queue to the unloading dock that becomes available next. Because none of the trailers has priority, this approach is experienced as fair, the truck driver that is waiting the longest is the next to unload. In baggage handling a similar procedure is used. Tug drivers pull their set of dollies or ramp carts to the transfer infeed where the fewest dollies are waiting. Although there is no dispatcher that tells the tug drivers where to unload, they create the FCFS schedule themselves. It is therefore clear that this is an interesting scheduling approach to study, for if a newly developed scheduling approach does not perform better than FCFS, it is of little use.

Another reason why FCFS is interesting, is that for problems where the processing times of individual containers are identical and the goal is to minimise the maximum waiting time of containers, it provides optimal schedules. Appendix C shows why FCFS is optimal for these problems.

Algorithm

The first-come-first-served scheduling algorithm is a simple approach and actually quite similar to the arbitrary scheduling algorithm. Both the algorithm flowchart (Figure 4.2) and pseudocode of FCFS are very much alike those of the arbitrary scheduling approach. It does, for instance, use the same triggers as arbitrary scheduling. After checking whether there is at least one container and one infeed available, the container that is first in queue is assigned to the first available infeed. Using the container that is first in queue instead of an arbitrary one, is in fact the only difference with arbitrary scheduling.

4.3 Dynamic Load Balancing Algorithm

The road to the development of the Dynamic Load Balancing Algorithm (DLBA) by McWilliams (2009b) has already been touched upon in Section 3.1. The algorithm, designed to solve the Parcel Hub Scheduling Problem (PHSP) by McWilliams et al. (2005), aims at balancing the workload over the different outfeeds. The underlying principle is that by providing each outfeed with the same amount of work, the probability of an outfeed being overloaded is minimised. Based on the literature review, we find that this is the state-of-the-art approach for generic inbound scheduling problems. To determine what benefits could be reached using state-of-the-art techniques, compared to the theoretical arbitrary scheduling and currently practised FCFS scheduling approaches, this research includes the DLBA.

It is important to mention the nine restrictive assumptions that have been formulated by McWilliams (2009b) when developing the PHSP and, to a lesser extend, the DLBA:

1. All unload docks are identical and have constant service rates. A trailer can thus be processed by any unload dock;
2. All load docks are identical and have constant service rates. A trailer can thus be processed by any load dock;
3. Parcels are transported from unload docks to load docks by means of a fixed network of conveyors;
4. All inbound trailers are available at the beginning of the transfer operation (this assumption is somewhat relaxed in the DLBA);
5. All outbound trailers are available at the beginning of the transfer operation;
6. Empty inbound trailers are instantaneously replaced with full inbound trailers;
7. Full outbound trailers are instantaneously replaced with empty outbound trailers;
8. All inbound and outbound trailers have equal priority; and
9. No trailer is pre-empted once unloading or loading has started.

These assumptions limit the problems to which the DLBA can be applied. The assumption that all inbound and outbound trailers (containers) have equal priority, for instance, does not hold for the baggage handling industry. There, it is clear that some outbound flights have priority over others, as they are due to leave earlier. Some inbound containers have priority, because they contain baggage items that are destined for a flight that leaves soon, whereas others can easily wait for a few hours. Assumptions that might not hold in reality, but are also assumed in this research, are the identical and constant service rates of unload and load docks and the availability of all outbound trailers at the beginning of the sorting shift.

Summarised, there is only one assumption that does not hold for the scheduling problems used in this research (equal priority) and there is one assumption that is already relaxed by McWilliams (availability of inbound trailers). The DLBA, therefore, seems a suitable and promising scheduling algorithm and should certainly be examined in this research.

Formal Problem Description

Unfortunately McWilliams (2009b) failed to provide a formal problem definition for the PHSP when he developed his DLBA. However, one year later McWilliams (2010) provided a integer linear program for the PHSP, when examining different solution approaches. Although the problem definition differs slightly from the problem that is solved by the DLBA, the proposed ILP is based on temporal decomposition, it still provides good insight in the problem that McWilliams was trying to solve. The formal problem definition of McWilliams (2010) is shown below, and uses the notation from Table 4.1.

$$\text{minimise} \qquad LD_{max} = \sum_{t \in T} LD_t \tag{4.1}$$

$$\text{subject to} \qquad \sum_{c \in C} \sum_{s \in S_c} x_{cst} \leq |U| \qquad \forall\, t \in T \tag{4.2}$$

$$\sum_{t \in T} x_{cst} = 1 \qquad \forall\, c \in C,\ \forall\, s \in S_c \tag{4.3}$$

$$|S_c| \cdot x_{c1t} - \sum_{s \in S_c} x_{cs(t+s-1)} = 0 \qquad \forall\, c \in C,\ \forall\, t \in T \tag{4.4}$$

$$\sum_{c \in C} \sum_{s \in S_c} \left(\frac{f_{cl}}{|S_c|} x_{cst} \right) \leq LD_t \qquad \forall\, l \in L,\ \forall\, t \in T \tag{4.5}$$

$$x_{cst} \in \mathbb{B} \qquad \forall\, c \in C,\ \forall\, s \in S_c,\ \forall\, t \in T \tag{4.6}$$

First, it is important to realise that the ILP by McWilliams (2010) minimises the total maximum workload per time bucket t (4.1). As a result the workload is balanced over the different outfeeds within one time bucket, but workload unbalance between time buckets is allowed. Second, the different constraints should be identified. Constraint 4.2 ensures that at any one time t no more

U	set of unload docks, $(u \in U)$
L	set of load docks, $(l \in L)$
T	set of time buckets, $(t \in T)$
C	set of inbound containers, $(c \in C)$
$\|S_c\|$	number of time buckets (segments) required to unload container c, $(S_c = \{1, \ldots, \|S_c\|\},\ s \in S_c)$
f_{cl}	number of parcels in container c destined for load dock l
x_{cst}	1 if container c, segment s, is assigned to time bucket t, 0 otherwise
LD_t	maximum workload at time bucket t, for any load dock $l \in L$
LD_{max}	summated maximum workload

Table 4.1: Notation for the FPD of the PHSP by McWilliams (2010)

than $|U|$ unload docks are used ($|U|$ represents the total number of unload docks). Next, constraint 4.3 stipulates that each container segment[1] $s \in S_c$ should be assigned completely to one time bucket, although the formulation of this specific constraint still allows a container segment to be split and assigned to multiple time buckets. Constraint 4.4 prohibits the pre-emption of unloading a container and constraint 4.5 determines for each time bucket t the maximum workload over all the load docks. Finally, all decision variables are defined as binary values (constraint 4.6), which ensures that a container segment ($s \in S_c$) is assigned to one time bucket indeed. This final constraint is thus a necessary addition to constraint 4.3 and ensures that the problem is indeed an ILP. Interesting is also that there generally is a feasible solution to this problem, as long as enough capacity is present. That is, the total number of required time buckets $\sum_{c \in C} |S_c|$ should not be more than the infeed capacity $|T||U|$. In practice, however, additional capacity (i.e. time buckets) would be required as not all containers arrive at $t = 0$. The capacity per load dock does not affect the feasibility of solutions, for this ILP does not model outfeed capacity.

McWilliams (2010) acknowledges that the number of time buckets $|T|$ is an implicit discrete decision variable that is limited by two boundaries, $t_{min} \leq |T| \leq t_{max}$. The boundaries are somewhat theoretical and can be determined using equation 4.7. The lower limit t_{min} assumes that the set of containers C can be divided into $|U|$ subsets, where each subset needs t_{min} time buckets to be processed. Given that such a division usually is non-existent,

[1] A container is divided into $|S_c|$ segments of equal size. In one time bucket each unload dock can unload exactly one container segment.

the value of t_{min} is usually an underestimated value. The upper limit t_{max} assumes that all containers need as many time buckets as the container that requires the most, which is a reasonable estimation if (and only if) the workload is indeed evenly distributed over the containers. If there is one container that requires an exceptional amount of time buckets, it would significantly complicate solving the ILP, because the number of constraints and decision variables is severely influenced by the number of time buckets.

$$t_{min} = \left\lceil \frac{\sum_{c \in C} |S_c|}{|U|} \right\rceil \qquad t_{max} = \left\lceil \frac{|C| \cdot \max\{|S_c|\}}{|U|} \right\rceil \tag{4.7}$$

Generally speaking, if the parcels are evenly distributed over the different containers $t_{min} \approx t_{max}$, which thus determines the optimal number of time buckets. In other situations a decision has to be made: either to reduce computational effort, by selecting $T \approx t_{min}$, or to improve the quality of the resulting schedule, by selecting $T \approx t_{max}$.

A drawback of the goal function as defined by McWilliams (2010) is that it only balances the workload within one time bucket. In fact, the ILP implicitly allows the maximum workload to change over time. Although this is a very practical schedule for automated sorters, for all outfeeds receive an even amount of parcels, thus minimising the risk of 'outfeed full' signals, it might be a very impractical schedule for the staff that has to remove the items from the outfeeds. Given the fact that this goal function might result in a schedule where the workload for each load dock cycles from low to high, back to low, etc. a very flexible or very large workforce would be required to process all the items in time. From this point of view, a goal function that balances the workload over the different time buckets might be preferred, i.e. the total amount of work in the system is more or less the same per time bucket, but may differ from load dock to load dock. Such a schedule would enable the company to have a fixed number of staff, fewer than in the previous situation, which can then be assigned to the load dock with the highest workload. Obviously the drawback of this approach is that it might 'overflow' the outfeed for a specific load dock, thus resulting in recirculation and reduced sorter capacity. Ultimately, a goal function that aims at both balancing the workload over time and balancing the workload over the different load docks is preferred. Such an approach would maximise the use of the sorter system, whilst minimising the required number of staff.[2]

Summarised, the PHSP is a complicated problem which size and feasible solutions are strongly dependent on the number of time buckets that is selected.

[2]The next chapter introduces an altered version of the DLBA that takes this dual workload balancing, both over the outfeeds and in time, into account.

This makes the PHSP a complex problem, in the sense that finding a feasible solution is already a difficult task. In addition, the selected goal function might not be suitable for sorter systems where load docks rely on manual labour to remove items from the conveyors.

Algorithm

The complexity that is caused by the number of time buckets used to solve the ILP to optimality is evaded by the DLBA. Instead of fixing the number of time buckets, the DLBA uses dynamic time buckets. Each time a container can be assigned to an infeed, i.e. when at least one container is available and there is at least one idle infeed, the algorithm starts a 'time bucket' and determines which container would balance the workload over the outfeeds the most. In fact, the best container is selected by calculating the workload imbalance for each container in the queue and selecting the container with the lowest workload imbalance. Equations 4.8–4.11 show how the workload imbalance (f_j) should be calculated, using the notation from Table 4.2. As the DLBA was originally designed for a parcel consolidation terminal, the terminology used is slightly different from the standard used in this thesis. One can, however, substitute unload dock, load dock, and trailer with infeed, outfeed, and container respectively, without loss of generality.

$$a_i = b_i + \sum_{u \in U} h_{iu} \tag{4.8}$$

$$c_{ij} = g_{ij} + a_i \tag{4.9}$$

$$c_j = \max_{i \in L} \{c_{ij}\} \tag{4.10}$$

$$f_j = |L| \cdot c_j - \sum_{i \in L} c_{ij} \tag{4.11}$$

The algorithm starts by determining the number of items for each of the outfeeds in the system in equation 4.8. Next, equation 4.9 determines for each of the containers in the queue what the total amount of items in the system would be if this container would be unloaded at the idle infeed. For each of the containers in the queue equation 4.10 determines the worst case, i.e. the outfeed with the largest workload. Finally, equation 4.11 determines the workload imbalance for each of the containers in the queue, by calculating the difference between the actual workload for all the outfeeds and the workload if all outfeeds would have the maximum workload as determined by equation 4.10.

Implementing this algorithm exactly as described above would result in a complicated and computational expensive algorithm. Each time the algorithm is called for, it has to cycle through all the items that are present on the sorter system and in the docked containers to determine a_i (equation 4.8).

U	set of unload docks
L	set of load docks
J	set of inbound trailers in the waiting queue
g_{ij}	number of parcels for load dock i in trailer j
h_{iu}	number of parcels for load dock i remaining in the trailer currently assigned to unload dock u
b_i	number of parcels for load dock i flowing over the material handling system
a_i	number of parcels for load dock i in the entire system
c_{ij}	number of parcels in the system destined for load dock i if trailer j is assigned to the idle unload dock
c_j	maximum workload over the set of load docks if trailer j is assigned to the idle unload dock
f_j	workload imbalance when trailer j is assigned to the idle unload dock
D	set of destinations
x_d	1 if destination d is assigned to a load dock, 0 otherwise

Table 4.2: Notation for DLBA equations (McWilliams, 2009b)

Therefore a slightly different approach is chosen to determine the value of a_i, which uses mostly elementary operations. This approach is based on the fact that as soon as a container is assigned to an infeed to be unloaded, the value of a_i increases by 1 for each item in this container destined for outfeed i. Similarly, when an item leaves the system at outfeed i, the value of a_i decreases by 1. In BHS scenarios, however, each outfeed is assigned to two destinations consecutively, which requires an adaptation of the approach. Instead of determining $a_i \ \forall\ i \in L$ the value of $a_i \ \forall\ i \in D$ is calculated, where D is the set of destinations. In order to be able to determine which destinations are assigned to an outfeed the variable x_d is used, $x_d = 0$ if destination d is not assigned to an outfeed and $x_d = 1$ if d is assigned to an outfeed. The DLBA then needs to balance the workload over all destinations d for which $x_d = 1$. The workload for the other destinations is either no longer existent because the outfeed has already closed for that destination, or is not yet relevant because these items should be transported to the EBS.

The uncontrollable inflow of baggage from the check-in desks and EBS can be integrated relatively simple in the DLBA, using 'imaginary containers'. As soon as the barcode label for a check-in baggage item is printed, it is assumed to have entered a container with infinite capacity and a FCFS queueing regime. This imaginary container acts as a buffer between the actual check-in procedure and the sorter system. When the bag is scanned to determine the corresponding outfeed, the bag is unloaded from the imaginary container and enters the sorter system. A slightly different approach can be used to model the EBS. The EBS can be regarded as a separate set of imaginary containers, one for each destination. When a bag enters the EBS, it is assumed to enter the container for its destination. At the start up time for a specific destination its corresponding container starts unloading, at which point the EBS releases the bags to the sorter system. This shows that the existing PHSP and DLBA are capable of modelling uncontrollable inflows. In reality, a more intelligent and complex EBS release mechanism might be used, but the development of such a workload balancing EBS system is outside the scope of this research.

Figure 4.3 shows the flowchart for the DLBA algorithm, and focusses on determining which container is the best solution. The algorithm is triggered when either a container arrives in the queue or when an infeed becomes available. In both cases the algorithm first checks whether an infeed is available, if not, the algorithm ends immediately. If an infeed is available, the algorithm determines how many containers there are waiting. If none are present the algorithm ends and if there is only one available this container is assigned to the first available infeed. There is no need to determine the size of the workload imbalance, because the available container is to be assigned to the available infeed anyway and calculating the workload imbalance could only delay the decision making process. If more than one container is waiting, the algorithm determines the values of c_{ij}, c_j and f_j respectively, for each of the containers in the queue. In order to ensure that only those destinations that are assigned to an outfeed are considered in determining c_j and f_j, equations 4.10 and 4.11 could be rewritten to:

$$c_j = \max_{i \in D | x_d = 1} \{c_{ij}\} \tag{4.12}$$

$$f_j = |L| \cdot c_j - \sum_{i \in D | x_d = 1} c_{ij} \tag{4.13}$$

Finally, the container with the lowest value of f_j, i.e. the container with the lowest workload imbalance, is assigned to the first available infeed and the algorithm returns to check whether there is another infeed available. In the baggage handling industry it is quite common that multiple containers arrive simultaneously, for they are pulled by the same tug. To prevent several instances of the scheduling algorithm to be running simultaneously, which would occur if each container triggered the scheduling algorithm, an arriving container only triggers the scheduling algorithm if no other instance is running yet.

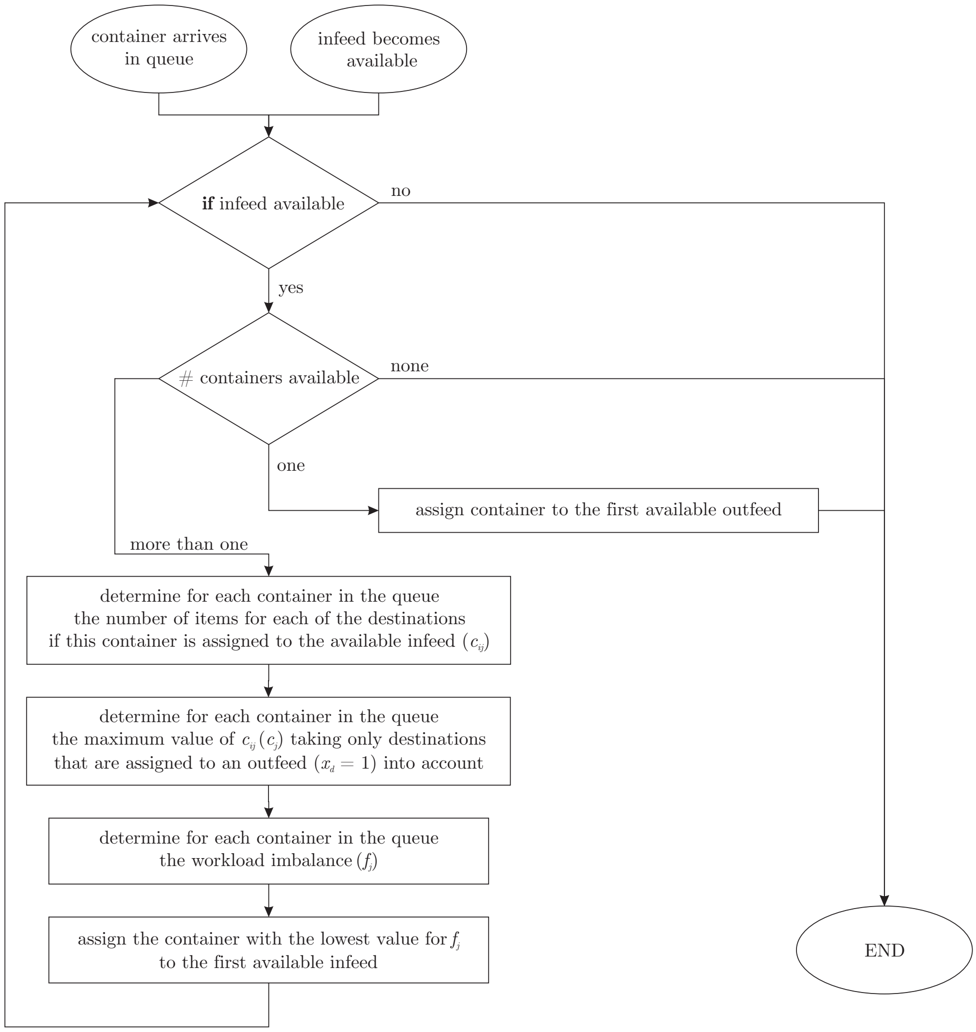


Figure 4.3: Flowchart for Dynamic Load Balancing Algorithm

Unfortunately, the DLBA also has some drawbacks, some of which have been mentioned in this chapter, that limit the applicability of the approach to realistic sorter systems. The next chapter therefore discusses the drawbacks and limitations of the DLBA in detail and focusses on the development of a new dynamic scheduling approach.

CHAPTER 5

Advanced Dynamic Load Balancing Algorithm

This chapter aims at the development of a new, improved, scheduling algorithm for high capacity sorters. The Dynamic Load Balancing Algorithm (DLBA) from the previous chapter is used as the foundation of this new scheduling approach, which will be referred to as advanced dynamic load balancing algorithm (ADLBA).

Section 5.1 starts by explaining the major drawback of the original DLBA, thereby justifying the development of a more detailed scheduling approach. Section 5.2 discusses how the drawback identified in Section 5.1 can be incorporated in a scheduling approach. Focus, however, is on how the resulting problems can be tackled, by adapting the objective of the scheduling approach.

Next, Section 5.3 introduces a formal problem that describes the type of problems that could be solved using the newly developed approach. Interesting is that only minor changes to the formal problem description (FPD) of the original DLBA are required to create a more detailed linear program.

Finally, Section 5.4 explains the resulting scheduling algorithm in detail. Because the new scheduling algorithm should be implementable at existing sorter systems, special attention is paid to how information systems can be updated to keep track of the required information and how information can be retrieved from these systems.

5.1 Justification

Because this research aims at the development of an online scheduling approach, the DLBA by McWilliams (2009b) is selected as the groundwork for this research. The DLBA, however, has one major drawback that hinders its application in realistic situations.

The PHSP defined by McWilliams et al. (2005) and the DLBA developed by McWilliams (2009b) assume that all internal transport times are zero. As a result, a parcel that is unloaded from a truck on an infeed is immediately loaded onto another truck at an outfeed. Note that, because load balancing focusses only on the relative arrival of parcels at the outfeed and does not consider the exact time at which a parcel arrives at the outfeed, the assumption of zero internal transport times is no stronger restriction than equal and fixed internal transport times. Furthermore, this restriction might be a valid simplification for small sorter systems such as the small parcel consolidation centre that McWilliams et al. use in their research (Figure 3.1), but does not necessarily hold for all sorter systems. Especially in large sorter systems with loop configuration, multiple infeed areas, presorters, and crossovers, internal transport times can differ significantly.

If the internal travel time to a specific outfeed is the same for any of the infeeds, and if this assumption holds for all outfeeds, the DLBA provides the best solution from a load balancing perspective. However, if this assumption does not hold, applying the DLBA could result in an unbalanced solution. The difference between these two situations can best be illustrated using the example from Figure 5.1. Consider a very simple scheduling problem with a planning horizon $T = 4$, that is, four time periods are available to unload all containers. Furthermore two containers are present, each containing two items for destination a and two for destination b, which have to be unloaded on either of the two unloading docks (1 and 2). One of the best solutions, according to the DLBA, is to unload container 1 at unload dock 1 at time $t = 1$ and container 2 at unload dock 2 at time $t = 3$. This would be a sound unloading strategy, as no container swap has to take place at an unloading dock. Now consider two sorter systems with different internal travel times ($t_{unloaddock,destination}$). The first sorter system satisfies the assumption made above, $t_{1a} = t_{2a} = 1$ and $t_{1b} = t_{2b} = 2$. The second sorter system does not satisfy the assumption, $t_{1a} = 2$, $t_{2a} = 1$, $t_{1b} = 3$, and $t_{2b} = 2$. Although the unload schedule is valid in both cases, the results show clearly that the workload for the second sorter system is not balanced any more. This shows that differences in internal transport times cannot be neglected. It is therefore that we aim at providing a solution for this issue.

The time that is required for unloading a container also influences the decision, for the unbalanced workload only occurs at the start and end of a *batch* of items (a set of items from the same container and with the same destination). As a result, if, due to the size of the batch, unloading a container requires much more time than the internal transport of items, one might decide to neglect the internal transport times. This supports the statement made before, that ignoring internal transport times might be valid for small sorter systems, especially in the parcel and postal industry. They usually unload trailers in 30 − 60 minutes, whereas the internal transport times are only minutes. In baggage handling, where containers can usually be unloaded in less than 10 minutes, neglecting the internal transport times is rarely sensible.

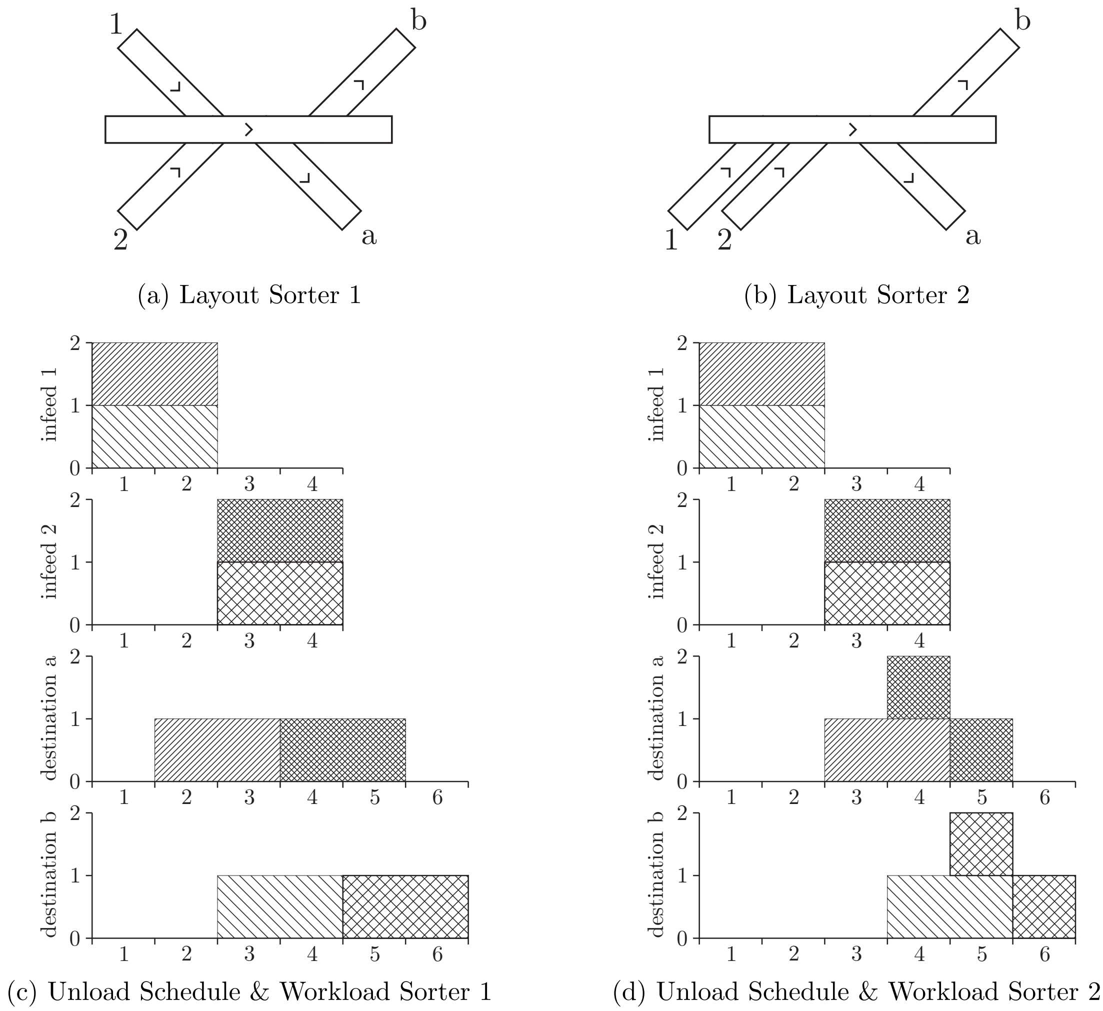


Figure 5.1: Effects of Internal Transport Times on Workload Balancing

Summarised, in order to be applicable to large sorters with significant differences in travel time, a successful scheduling approach should take internal travel times into account. This chapter focusses on the development of such an approach.

5.2 Concepts & Considerations

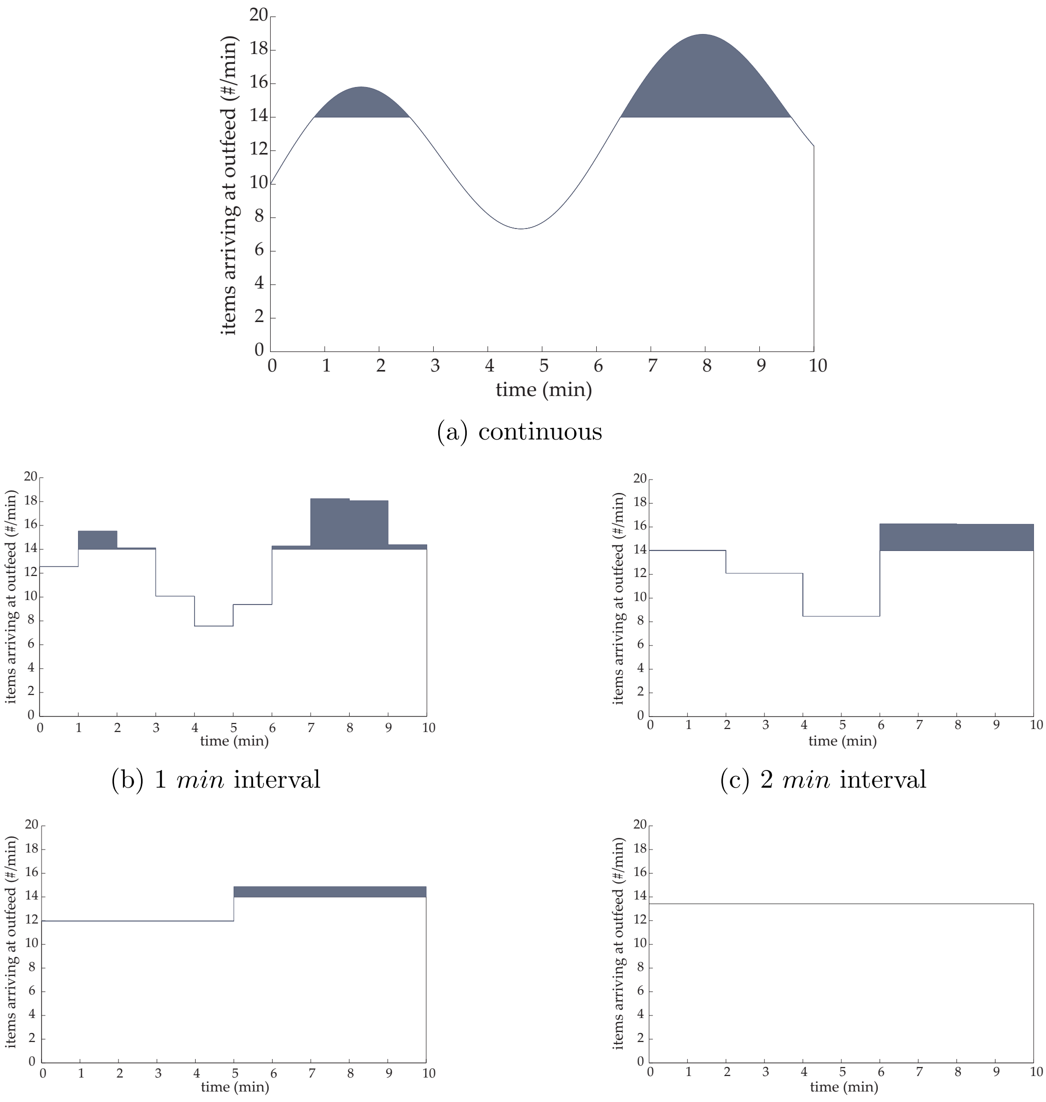
Introducing internal transport times and thus delays in the existing DLBA, does significantly complicate the scheduling operations. The DLBA only had to keep track of the total number of items in the system destined for a specific outfeed. The assumption was that as long as the total number of items in the sorting process for each of the outfeeds was more or less equal, the resulting workload would be more or less identical and thus optimal.

One of the consequences of taking transport times into account, is that balancing the total number of items in the sorter system is no longer a valid optimisation criterion. The workload should not only be balanced over the different outfeeds, but also over time. Although a similar approach as with the DLBA could be applied by simply selecting the highest value and subtracting all other values from the highest one, it seems that a more rational approach could be developed. One of the benefits of determining for each outfeed at each point in time the expected *outflow* (the number of items that arrive at the chute) is that we also can determine whether the capacity of the outfeed is exceeded or not. Obviously, solutions where the outfeed capacity is exceeded are undesirable, whereas all other can be considered equally good. Figure 5.2a shows a possible expected outflow pattern, i.e. not taking recirculating items into account, for a single outfeed. Assuming that the outfeed has a capacity of 14 ipm (items per minute) the blue solid areas are considered to be undesirable; the excess items would recirculate and need to be accommodated at another, later, point in time. A simple approach would thus be to summate these areas for all the outfeeds and use this as a minimisation criterion.

This approach assumes that only the volume of excess items is relevant. However, one could argue that not only the volume, but also the rate at which these excess items arrive is important. A solution where two excess items arrive at two separate points in time is likely to be preferred over a solution where they would arrive simultaneously. After all, after a recirculation these excess items merge with the outflow that was originally destined to arrive at that point in time, and a single item can more easily be accommodated than two items. An approach in which the excess volume at a specific point in time is squared when determining the suitability of a container is therefore preferred.

Determining the squared excess outflow in a continuous manner, as assumed in the previous paragraphs, is impractical. Not only does it require the continuous monitoring of individual items, it also requires the algorithm to construct outflow figures similar to Figure 5.2a each time a scheduling decision has to be made. A computationally less challenging approach, is to use time buckets instead. In the time bucket approach, procedures determine for each item in which time bucket it is likely to arrive at the outfeed. When a scheduling decision has to be made, the algorithm can consecutively scan the outfeeds and the next n time buckets. Each time the squared value of the excess outflow can be added to the objective value of a specific container. The procedural details of this approach are explained in Section 5.4.

The size of the time buckets is an important model parameter, for it af-



(a) continuous

(b) 1 *min* interval

(c) 2 *min* interval

(d) 5 *min* interval

(e) 10 *min* interval

Figure 5.2: Undesirable Outflows for Sorter Systems

fects the level of detail that can be achieved. Obviously smaller time buckets lead to greater detail in the results, but time buckets that are too small might provide unrealistic results. If, for instance, time buckets span 1 ms, an ordinary unloading operation would result in zero outflow during most of the time buckets and an incidental peak at the exact time an item arrives. Figure 5.2 shows the effect of time bucket size on the level of detail. Whilst Figure 5.2b clearly shows two areas with excess outflow, Figure 5.2e would lead us to the conclusion that there is no excess outflow whatsoever. Finding the optimal balance between detail on one side and realistic and useful results on the other, is therefore important. A clear lowerbound for the size of time buckets would be the time required for unloading a single item. Smaller time buckets would result in the peak behaviour mentioned earlier. Similarly an upperbound can be determined using the time required to unload a single container. In baggage handling unloading a container takes about four minutes, in parcel and postal this requires around 25 minutes. Selecting a time bucket size larger than the unloading time of a single container would provide no additional information compared to the DLBA.

There is, however, a third subject that should be taken into account when selecting the size of the time buckets. In the preceding part of this section, a deterministic problem has been assumed implicitly. The internal transport times are, however, not deterministic. Sometimes items cannot immediately enter the main sorter which causes the items to arrive later than expected. Similarly, when outfeeds are full or blocked items need to recirculate, again causing them to arrive late. Incorporating these delays in the scheduling algorithm itself is not feasible, as this type of information becomes available only one or two minutes before items are expected to arrive at the outfeed. Using very small time buckets suggests that the calculations are very accurate, whereas in reality, due to these uncertain transport times, the calculations are estimations at best. The uncertainty about the internal transport times is, therefore, another reason not to select time buckets that are too small.

Summarised, in order to penalise large excess outflows the squared value of these flows is used as optimisation criterion. In order to achieve enough detail a time bucket size of 1 minute is used. This is approximately a quarter of the time required to unload a single ULD and more or less equal to the smallest distance between an infeed and outfeed in sorter systems. Using time buckets of 1 minute provides enough detail but also results in valid and meaningful outflows.

5.3 Formal Problem Description

By including the internal transport times, the formal problem description increases in complexity. Where for the PHSP the mere fact that a container was docked at an outfeed was enough information, for the ADLBA this information is too limited. In this case, it is important to know whether the container is docked at infeed one or two, as travel times to outfeeds can differ significantly amongst infeeds.

Because focus is now on the excess outflow, the goal function and constraints have to be adapted to keep track of the excess outflow of every outfeed at every time bucket. As a squared goal function was proposed in the previous section, the FPD is no longer a linear integer program. Another possible goal function would be a minimax goal function that minimises the maximum exceeding of the capacity. Drawback of this approach is that a solution in which one outfeed exceeds its capacity by $n + 1$ items is considered to be worse than a solution in which all outfeeds exceed their capacity by n items, whereas in the latter case much more items are forced to recirculate. Obviously many other goal functions, linear as well as non-linear, can be developed and tailored to the specific requirements for the problem at hand. The results of these programs can, however, be very different indeed.

Equations 5.1 to 5.6 provide the FPD of the problem for which the ADLBA was designed. The notation is, for convenience, more or less the same as the notation for the PHSP (Equations 4.1 to 4.6 and Table 4.1). Naturally, additions had to be made and Table 5.1 therefore provides a full overview of the notation. To provide a better layout, instead of the formal notation $\forall\, c \in C$ the abbreviated notation $\forall\, c$ has been used.

$$\text{minimise} \qquad EF_{tot} = \sum_{t \in T} \sum_{l \in L} (EF_{lt})^2 \tag{5.1}$$

$$\text{subject to} \qquad \sum_{c \in C} \sum_{s \in S_c} x_{csut} \leq 1 \qquad \forall\, t,\ u \tag{5.2}$$

$$\sum_{u \in U} \sum_{t \in T} x_{csut} = 1 \qquad \forall\, c,\ s \tag{5.3}$$

$$|S_c| \cdot x_{c1ut} - \sum_{s \in S_c} x_{csu(t+s-1)} = 0 \qquad \forall\, c,\ u,\ t \tag{5.4}$$

$$\sum_{c \in C} \sum_{s \in S_c} \sum_{u \in U} \left(\frac{f_{cl}}{|S_c|} x_{csu(t-t_{ul})} \right) - F_l \leq EF_{lt} \qquad \forall\, l,\ u,\ t \tag{5.5}$$

$$x_{csut} \in \mathbb{B} \qquad \forall\, c,\ s,\ u,\ t \tag{5.6}$$

Compared to the original FPD of the PHSP a few small changes are made with respect to the optimisation criterion. Goal function 5.1 now summates the squared value of the excess outflow EF over all the load docks and time buckets. Another small modification is made to constraint 5.2. Instead of checking whether no more container segments are assigned than there are unloading docks, it now ensures that each unloading dock is used by one container segment per time bucket at the most. Constraints 5.3 and 5.4 remained unaltered, except for the addition of the index u for the unload docks. The combination of the two still ensures that each container segment is assigned exactly once, and that a container is emptied in successive time buckets. Probably the most important changes are made to constraint 5.5, where, in the first part of the equation, $\sum_{c \in C} \sum_{s \in S_c} \sum_{u \in U} \left(\frac{f_{cl}}{|S_c|} x_{csu(t-t_{ul})} \right)$, the internal transport times t_{ul} are incorporated. In order to measure the outflow at load dock

U	set of unload docks, $(u \in U)$
L	set of load docks, $(l \in L)$
T	set of time buckets, $(t \in T)$
C	set of inbound containers, $(c \in C)$
S_c	set of time buckets (segments) needed for container c, $(s \in S_c)$
F_l	outflow capacity for load dock l [*items per hour*]
f_{cl}	number of parcels in container c destined for load dock l
t_{ul}	travel time from unload dock u to load dock l [*time buckets*]
x_{csut}	1 if container c, segment s, is assigned to unload dock u in time bucket t, 0 otherwise
EF_{lt}	excess outflow at load dock l in time bucket t
EF_{tot}	summated excess outflow

Table 5.1: Notation for the FPD of the ADLBA

l at time t the flows that were generated by the unloading docks at time $t - t_{ul}$ have to be used. The second part of the equation ensures that only outflows that exceed the capacity F_l force the value of EF_{lt} to be positive. In theory EF_{lt} could become negative when the outflow does not exceed the capacity. In that case EF_{lt}^2 would be added to the goal function, thereby deteriorating the solution. Because Constraint 5.5 only provides a lower bound for EF_{lt}, EF_{lt} is allowed to become zero, which significantly improves the objective value, without changing the solution. Finally, Constraint 5.6 ensures that all decision variables are binary.

5.4 Algorithm Outline

Because in the ADLBA not only the arrival of an item at an outfeed, but also the time bucket in which it is expected to arrive is important, more procedures are required to keep the system information up-to-date. This section therefore starts by explaining how system information is updated for arriving check-in and released EBS items. The section continues with the procedure to update the system information after a container has been assigned to an infeed. Finally, the assignment algorithm itself is discussed. For convenience, Table 5.2 (page 55) provides an overview of the notation that is used in the flowcharts, equations, and explanations throughout this section.

In the DLBA processing check-in items was relatively straightforward. As soon as an item was announced the counter for its destination was increased by one. For the ADLBA, also the time at which the item is expected to arrive at the outfeed has to be determined. Figure 5.3a provides the flowchart of this updating procedure. First the algorithm determines the outfeed the item should be transported to, using its final destination. Next, the exact time at which the item would arrive at its outfeed (TB_{start}) is determined using:

$$TB_{start} = \left\lfloor \frac{ac + t_{CI} + t_{io}}{tb} \right\rfloor + 1 \tag{5.7}$$

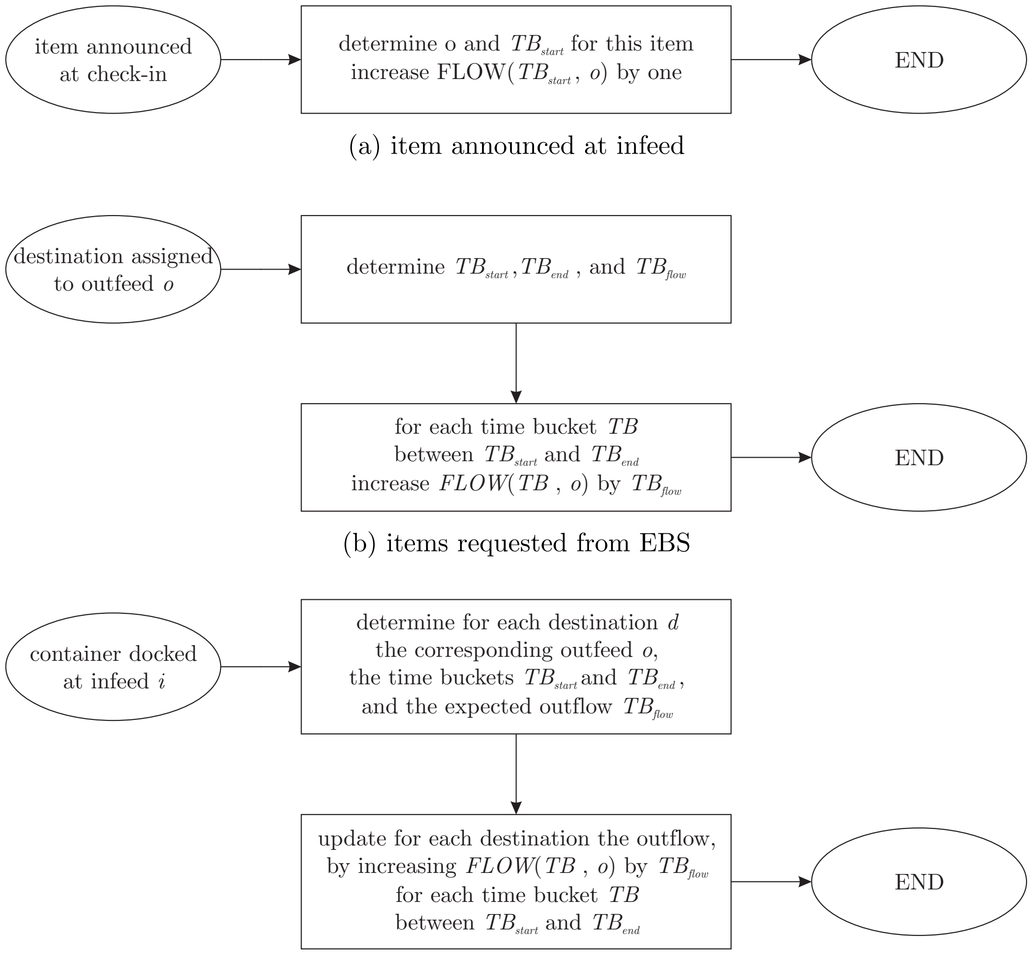
ac denotes the current time in seconds and, because items are announced before they actually arrive at the check-in infeed of the main sorter, a delay time t_{CI} is added. t_{io} represents the time that is required to transport the item from the infeed to the corresponding outfeed, i.e. the outfeed to which the item's destination is assigned. The numerator thus indicates the exact time at which the item would arrive at the outfeed, assuming a free-flow situation. Dividing this time by the size of a time bucket tb and rounding down provides the time bucket index at which the item would arrive. The value of one is added to ensure that the indices of the time buckets are in $\mathbb{N}^+$, not doing so would cause the first time bucket to have index 0. Finally, the variable $FLOW(TB, o)$, which keeps track of the expected outflow, is updated.

Also when items are requested from the EBS, matters are more complicated for the ADLBA than for the DLBA. Figure 5.3b shows the flowchart of the procedure that updates the expected outflows when items are requested from the EBS. The approach not only requires knowledge about which destination (d) is assigned to which outfeed (o), also information about the number of items in the EBS for this destination (EBS_d) and the travel time from the EBS to the outfeed is necessary. For the travel time the notation t_{io} will be used, as the EBS is just a special type of infeed. Based on this information the first (TB_{start}) and last (TB_{end}) time bucket in which the items are expected to arrive at the outfeed is determined using:

$$TB_{start} = \left\lfloor \frac{ac + \frac{1}{F_i/3600} + t_{io}}{tb} \right\rfloor + 1 \tag{5.8}$$

$$TB_{end} = \left\lfloor \frac{ac + \frac{1}{F_i/3600} \cdot EBS_d + t_{io}}{tb} \right\rfloor + 1 \tag{5.9}$$

Because the information has to be updated before the items have entered the sorter system, first the time required to put one item on the conveyor is added to the current time ac. As each infeed processes F_i items per hour, which in the case of the EBS is the rate at which items can be retrieved from the storage area, the time required to unload one item is $\frac{3600}{F_i}$ seconds. This, again, ensures that the numerator of equation 5.8 displays the exact time bucket in



(a) item announced at infeed

(b) items requested from EBS

(c) container assigned to infeed

Figure 5.3: Flowchart of updating procedures for ADLBA

which the first of all items in the EBS is expected to arrive at the outfeed. The last item arrives after all items have been retrieved from the EBS.

For the check-in flow, determining how many items to add was simple, each time a single item arrived and thus one had to be added. For the flow from the EBS things are a bit more intricate. The procedure assumes that the items arrive homogeneously over the different time buckets. The number of items that arrive per time bucket (TB_{flow}) is therefore:

$$TB_{flow} = \frac{EBS_d}{TB_{end} - TB_{start} + 1} \tag{5.10}$$

The only thing that remains for the procedure is to add the value of TB_{flow} to each TB from TB_{start} up to and including TB_{finish}.

ac	current simulation time [sec]
t_{io}	internal transport time from infeed i to outfeed o [sec]
t_{CI}	time between announcement of check-in items and actual arrival on infeed conveyor [sec]
tb	length of one time bucket [sec]
TB	counter for time buckets
TB_{start}	first time bucket in which items arrive at an outfeed
TB_{end}	last time bucket in which items arrive at an outfeed
TB_{flow}	number of items that arrive at an outfeed during one time bucket
C_d	number of items for destination d inside a container
C_{tot}	total number of items inside a container
EBS_d	number of items in EBS with destination d
$FLOW(TB, o)$	expected outflow at time bucket TB for outfeed o
$FLOW_c$	expected outflow if container c is assigned to the selected infeed
$FLOW_i$	expected outflow if the selected container is assigned to infeed i

Table 5.2: Notation for ADLBA equations

When the dispatcher decides that a specific container should be docked at an outfeed, an approach similar to the one used for the EBS can be applied. Figure 5.3c provides the procedure that is used when a container is docked. The most important difference is that in this case the procedure should perform the calculations for each individual destination. That is, for each possible destination, the assigned outfeed o and the values of TB_{start}, TB_{end}, and TB_{flow} should be determined using the following equations:

$$TB_{start} = \left\lfloor \frac{ac + \frac{1}{F_i/3600} + t_{io}}{tb} \right\rfloor + 1 \tag{5.11}$$

$$TB_{end} = \left\lfloor \frac{ac + \frac{1}{F_i/3600} \cdot C_{tot} + t_{io}}{tb} \right\rfloor + 1 \tag{5.12}$$

$$TB_{flow} = \frac{C_d}{TB_{end} - TB_{start} + 1} \tag{5.13}$$

Here C_d denotes the number of items inside the container for destination d, and C_{tot} is the total number of items in the container. The time required for unloading $\left(\frac{1}{F_i/3600} \cdot C_{tot} \text{ seconds}\right)$ therefore depends on all items inside the container, whilst the flow arriving at a specific outfeed is depending only on the destinations that are assigned to this outfeed.

The scheduling algorithm itself (Figure 5.4) is more complicated than the previous algorithms. Not only because the computations are more extensive, but also because many more scheduling variants exist. Most of these variants can, however, be considered special cases of a more generic approach. Ideally, the optimal solution would be found using combinatorial optimization. By determining for each container-infeed combination the expected excess outflow, a set of container-infeed combinations that minimises the total excess outflow can be constructed. This could, however, result in a time consuming optimisation problem. Therefore a constructive heuristic is applied instead: sequentially each available infeed is assigned its best container, until there is only one container left. If there are still infeeds available, this container is assigned to the best available one.

Determining the objective values for a decision, a decision being the assignment of a specific container to a specific infeed, is relatively simple and uses the system information from the $FLOW$ variable. However, time buckets that have already passed are no longer relevant, and information about time buckets that are relatively far in the future is doubtful, because recirculation and merging difficulties may alter these predictions. In this research, focus is therefore on the expected outflow in the next 15 minutes. The set TB_{horiz} denotes all time buckets that are part of the planning horizon. System information of these time buckets is copied and stored in $FLOW_{ic}$.

The effects of the proposed decision have to be determined by updating the values of $FLOW_{ic}$. This can easily be done using equations 5.11 –5.13 and the procedure for assigning containers to infeeds explained earlier. Finally, the objective value for each decision can be determined using equation 5.14. The best assignment is the assignment with the lowest value of EF_{ic}, which represents the summated square of the excess outflow if container c is assigned to infeed i. To ensure that only positive excess outflows are penalised, the maximum of zero and the expected excess outflow is used.

$$EF_{ic} = \sum_{TB \in TB_{horiz}} \sum_{o \in O} \left(\max \left\{ 0, FLOW_{ic}(TB, o) - \frac{F_o \cdot tb}{3600} \right\} \right)^2 \tag{5.14}$$

For completeness, Appendix B provides the pseudocodes of the main scheduling algorithm (page 103) and the algorithm for determining the objective value of a single assignment decision (page 104).

The main drawback of the existing DLBA, the assumption of equal internal

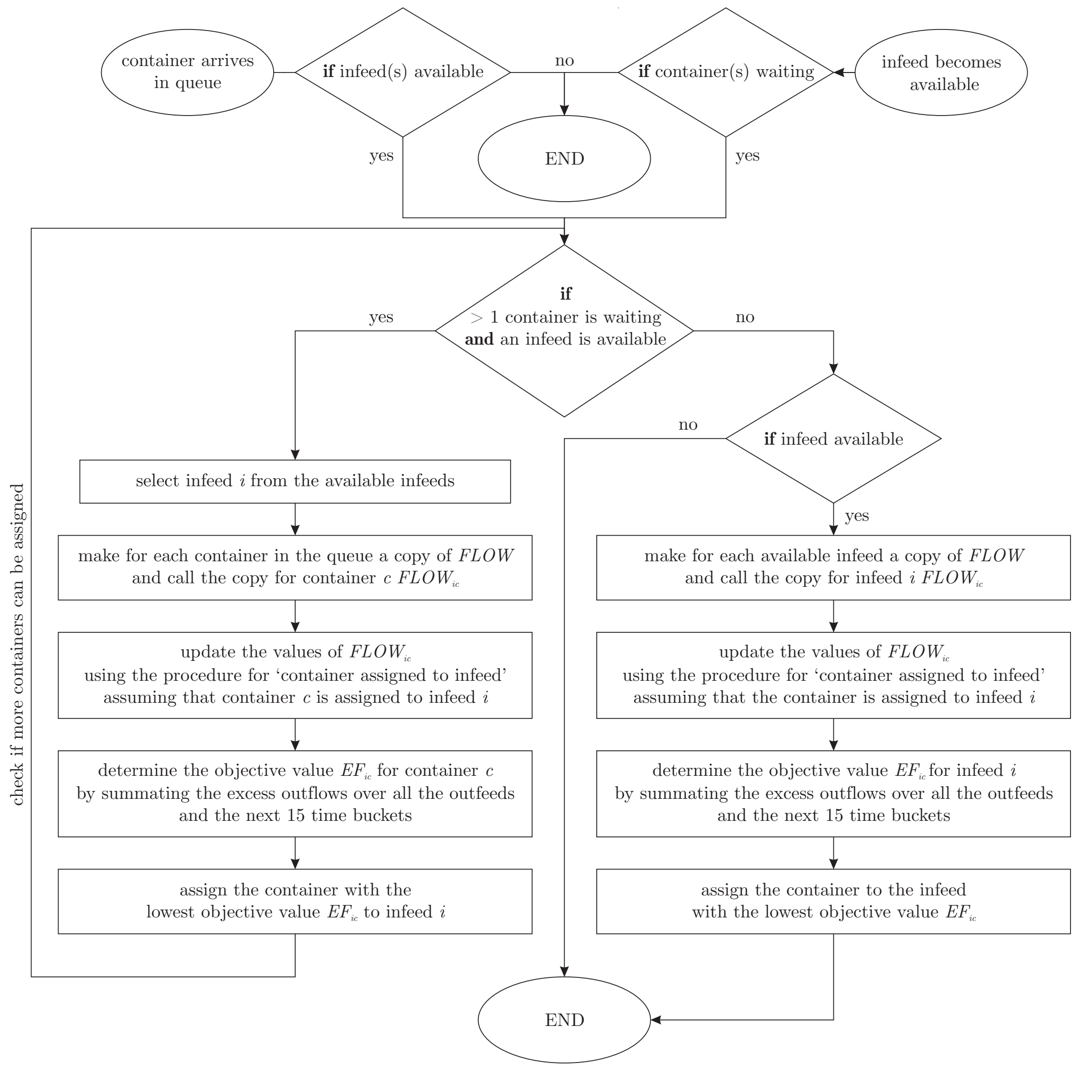


Figure 5.4: Container Assignment Algorithm for ADLBA

transport times, is tackled with the development of the ADLBA. Both the DLBA and ADLBA do, however, have some minor drawbacks that have not yet been discussed. These drawbacks, and the approaches to deal with them, are the focus of the next chapter.

Chapter 6

Scheduling, Urgency & Delayability

The previous chapter focussed on the integration of internal travel times in a dynamic load balancing algorithm. There are, however, two other issues that the DLBA and ADLBA algorithm, as well as arbitrary scheduling and FCFS, do not take into account when assigning containers to available infeeds. This chapter provides two extensions to the DLBA and ADLBA, which enable them to cope with these issues.

The first extension is the inclusion of priority. If an outfeed is about to close and one container holds many items destined for this outfeed, it is obvious that this container should be scheduled first. Otherwise, all items for this destination arrive tardy and would thus be counted as missorts. Section 6.1 therefore focusses on how to prioritise specific containers.

Besides the possibility to give priority to urgent containers, there is also the option of delaying containers. This is especially useful if a container is filled with items that can currently not be sorted, because their destination is not yet assigned to an outfeed. These items would normally be transported to the EBS, a quite costly solution. Storing these items inside their container on the apron would not only save valuable EBS space, but would also reduce the workload of the sorter system, thereby reducing the probability of missorts. Section 6.2 focusses on how containers can be identified as delayable.

Finally, Section 6.3 discusses the resulting algorithm extensions in detail. Again, special attention is paid to the implementation of these algorithms as they should be able to cooperate with many different scheduling approaches.

6.1 Urgency

A possible approach to determine the urgency of a single bag has already been introduced in the literature review (Chapter 3). Hallenborg (2007) provided an approach to determine which tote was the most urgent (Equation 3.1); information that could be used to determine which carriers should get priority on the overtaking lane in a Destination Coded Vehicle transport system.

One of the more remarkable aspects of the urgency equation by Hallenborg (2007) is that, besides giving priority to urgent totes, it also assigns negative urgencies to totes that are extremely early and should therefore receive no priority at all. This, however, is not a very suitable approach for containers that contain multiple items. The approach of Hallenborg therefore serves only as a basis for a newly developed urgency equation.

It is important to realise that items at some point in time become non-urgent. Although this seems counter-intuitive, the explanation is actually quite clear. At a certain point in time, it is physically impossible to transport the items through the sorter system to the correct outfeed before it closes. Hallenborg (2007) therefore classifies all items for which the assigned outfeed closes within 20 minutes as extremely urgent, and assumes that they would be transported to the aeroplane manually, instead of using the sorter system.

This research, however, assumes a sorter system without a manual labour alternative,[1] which is why extremely urgent (hot) baggage items are also fed onto the sorter system. Determining when it is physically impossible to sort an item on time is difficult, especially because it is not known whether items are unloaded from the container first or last. In order to make sure that items receive full priority until it is absolutely certain that they cannot be delivered on time, the point at which items for destination d become non-urgent, denoted by U_{end}, is equal to the internal transport time t_{io}, where o is the outfeed destination d is assigned to. That is, items become non-urgent when they cannot be delivered on time, even if they are the first item that is unloaded from a container. A side-effect of this approach is that U_{end} is dependent not only on the outfeed to which an item is assigned, but also on the infeed at which the container is to be unloaded. Unfortunately, this approach is not possible for the DLBA, nor for FCFS and ARB. In contrast to the ADLBA, they do not keep track of internal transport times. Therefore a fixed value of 5 minutes, which is an estimation of the average internal transport times, is used instead. To emphasize the difference between the equation of Hallenborg and the one developed here, t_d is used for the time until the outfeed assigned to destination d closes.

Another point in time that is relevant is U_{start}, the moment that items become urgent. Although many different approaches could be selected, a relatively simple one is used. Each destination is urgent for 30 minutes and U_{start} is therefore equal to $U_{end} + 30$ minutes.[2]

[1] This research aims at maximising the performance of sorter systems. Therefore, an alternative that is capable of handling 'difficult' baggage items is not preferred.

[2] U_{start} and U_{end} represent the time until the outfeed closes, which is why $U_{start} > U_{end}$.

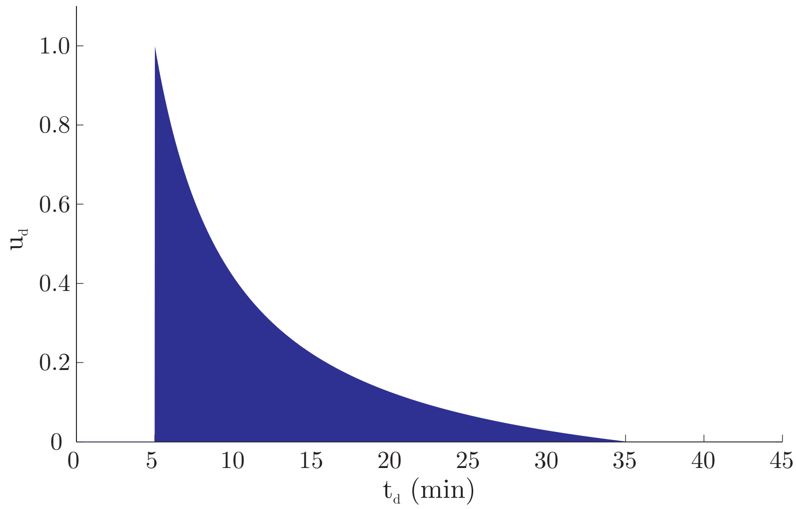


Figure 6.1: Urgency of Items for Destination d

Finally, the urgency of a destination is modelled in such a way that it starts at zero at time U_{start} and equals one at time U_{end}. Urgency is thus determined using equation 6.1. Substituting $U_{start} = 35$ and $U_{end} = 5$ minutes, the priority of destination d follows the graph of Figure 6.1.

$$u_d = \begin{cases} \frac{t_{io}}{t_d} \cdot \frac{U_{start} - t_d}{U_{start} - t_{io}} & U_{end} \leq t_d \leq U_{start} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

Hallenborg (2007) has the advantage of determining the urgency and priority of individual items. He can therefore rank them in such a way that the most urgent one gets the highest priority. The problem at hand in this research, however, requires us to determine the urgency of a container that contains items for a number of destinations, each with its own priority.

Generally speaking, there are three common approaches that could be applied to determine the urgency of a container. A container is assigned either the maximum of all individual item urgencies, the average of all individual urgencies, or the sum of all individual urgencies. Letting J_c denote the subset of items j that are currently in container c and $d(j)$ the destination of item j, the container urgency can be determined using any of the three following equations:

$$u_{\max} = \max_{j \in J_c} \left\{ u_{d(j)} \right\} \qquad u_{\text{avg}} = \frac{1}{|J_c|} \sum_{j \in J_c} u_{d(j)} \qquad u_{\text{sum}} = \sum_{j \in J_c} u_{d(j)} \tag{6.2}$$

Drawback of $u_{\max}$ is that there are numerous situations in which it would overestimate the urgency of one container and underestimate that of another.

If, for instance, there is one destination that has become urgent, a container that holds one item for this destination is just as urgent as a container that holds ten. In reality, however, the latter container would be considered to be more important. The u_{avg} tackles this issue, as the latter container would have an urgency that is ten times higher than that of the former, assuming that they contain the same number of items. Unfortunately, this assumption is also the drawback of this approach, for a container holding only 13 items, provided they are all urgent, may receive priority over a container in which 14 out of 15 items are urgent. The u_{sum} approach settles this issue in favour of the latter container and is therefore used in this research.

Having determined the urgency of individual containers, the question is how to select the container that is to be assigned to the available infeed. In the situation where there are no urgent containers, i.e. all containers have zero urgency, the solution is simple: apply the DLBA or ADLBA scheduling approach to all containers that are waiting to be scheduled. If there is only one urgent container, there also is an obvious solution: assign this container to an available outfeed. This reduces the assignment problem to a situation in which there are multiple containers waiting, some of which have priority. One of the most natural solutions is to select the container with the highest priority. It is, however, not unlikely that this container would severely deteriorate the situation on the main sorter, something we try to prevent by applying load balancing algorithms. An alternative approach would be to use the priority algorithm to make a pre-selection from the available containers, and apply the existing scheduling approaches to this selection. This ensures that a priority container is scheduled, whilst also balancing the workload over the outfeeds. This approach is more true towards the load balancing concept that is the foundation of this research, and is therefore selected.

It does, however, not entirely solve the problem of containers with different priorities. If, for instance, there is one container with a urgency of 1 and one with a urgency of 15, the latter is obvious much more urgent than the former. Subsequently it should receive more priority and thus be scheduled first. Therefore, using a 'rule-of-thumb', only containers with an urgency that is within 25% of highest urgency are selected. This ensures that the DLBA or ADLBA can select the best container from the ones with the highest priority.

6.2 Delayability

The current scheduling approaches do not consider the option of temporarily parking containers. Although the trucks and tugs are obviously parked somewhere, awaiting the request of the dispatcher, this is done in a somewhat unorganised way. Perhaps even more important, at airports parking is prohibited in most places in order to prevent holdups for other ground staff. Therefore, decisions have to be made on where to park which container, some need to be parked close to the infeeds, whereas others might stay on a container park farther away.

The scheduling approaches used in the previous chapters all use the same approach for arriving containers. They are sent to the queue of waiting containers at the parking and are announced to the dispatcher, who can then decide which container should be unloaded at which infeed. It would, however, also be possible to temporarily *delay* specific containers. A container that is delayed, is sent to a separate (remote) container park and is not announced to the dispatcher, thereby preventing it from being assigned to an infeed.

Delaying containers can be advantageous for two reasons. First, many airports lack infeed capacity during peak hours (usually workdays between 6am – 9am). During these peak hours the transfer baggage supply is much higher than during other periods of the day. Although this problem could be solved by installing more transfer infeeds, airports hesitate in doing so, as infeeds are relatively expensive due to the working space that is required around them and the compulsory security screenings. One of the possibilities to reduce these peaks, is to temporarily park ULDs on a remote ULD park. If ULDs that contain only cold baggage are selected, no additional bags miss their flight. In fact, due to less congestion on the sorter system, it is likely that the number of bags that miss their flight is even reduced. Second, cold baggage items are now stored in relatively expensive EBS systems. Storing them in a container on the apron is, obviously, a much cheaper solution. Delaying containers is therefore not only advantageous during peak hours, but could also be used during the remainder of the day. Small EBS systems, however, will always be required, for there will always be early bags in the sorter system.

Unfortunately, delaying containers also has its drawbacks. Although the approach realises a reduction in workload at the time the container arrives, it also increases the workload of the sorter system when the container becomes available to the dispatcher. At that time, not only the containers that are arriving, but also the containers that are waiting at the remote parking have to be accommodated. A second drawback is that it is possible that containers are delayed so much that items inside cannot be sorted on time, with a higher missort rate as consequence.

The decision whether or not a container should be delayed has to be made at the time of arrival and is relatively simple. The question when a container should be made available again to the dispatcher is a harder one. Generally it seems best to make a container available as soon as possible, as this leaves as much time as possible to accommodate the container. Because not all outfeeds are assigned to a new destination at the same time, it is likely that this would cause one item to be transported to an outfeed, whereas others would still require the EBS. This would, obviously, nullify the effects that were achieved by sending the container to the remote parking in the first place. Waiting till the last item can be sorted directly to the outfeed would, however, cause other problems. It might severely decrease the size of the time window in which the dispatcher can assign the container, even to the extend where some items are late, i.e. the outfeed is no longer assigned to the destination. Therefore a balance between these two extremes has to be found. In the baggage handling industry, each outfeed is assigned to a destination for approximately three hours. A suitable approach would therefore be to make a container available an hour after the destination of one of the items is assigned to an outfeed. This still leaves two hours to sort the items for the destination that was assigned to an outfeed first, but also ensures that many other items may be sorted directly to the correct outfeed.

6.3 Algorithm Outline

The priority and delayability algorithm are not independent scheduling approaches, but serve as extensions to the existing scheduling approaches. Therefore their flowcharts do not start with the *container arriving in the queue* and *infeed available* triggers found in earlier chapters. For convenience this chapter only contains the flowcharts of the extensions. Appendix B contains the elaborate pseudocodes for the priority scheduling (page 105) and delayability (page 106) extension.

In fact, the DLBA and ADLBA call the priority algorithm (Figure 6.2) when they start searching for a suitable container for infeed i. The priority algorithm starts by determining for each of the possible destinations the current urgency u_d, using equation 6.1. The values of U_{start} and U_{end} are dependent on the scheduling approach that is currently being used. The value of t_d can easily be determined using the outfeed closing time for destination d and the current time. In theory it is possible to abort the scheduling approach if none of the destinations is urgent, for the resulting conclusion would be that none of the containers can be urgent. This behaviour, however, is not incorporated in the flowchart in order to make it more readable.

For each container in the queue, the urgency is determined using $u_c = u_{\text{sum}} = \sum_{j \in J_c} u_{d(j)}$ and the maximum urgency $u_{MAX} = \max(u_c)$ is updated. All containers that are non-urgent (i.e. $u_c = 0$) are immediately removed from the list. After this step the list is either empty, indicating that there are no urgent containers, or the list contains one or more containers that are considered urgent.

If the list is empty, the DLBA or ADLBA scheduling algorithm is applied to all containers that are currently queueing. If there are urgent containers present, a selection will be made in order to ensure that only very urgent con-

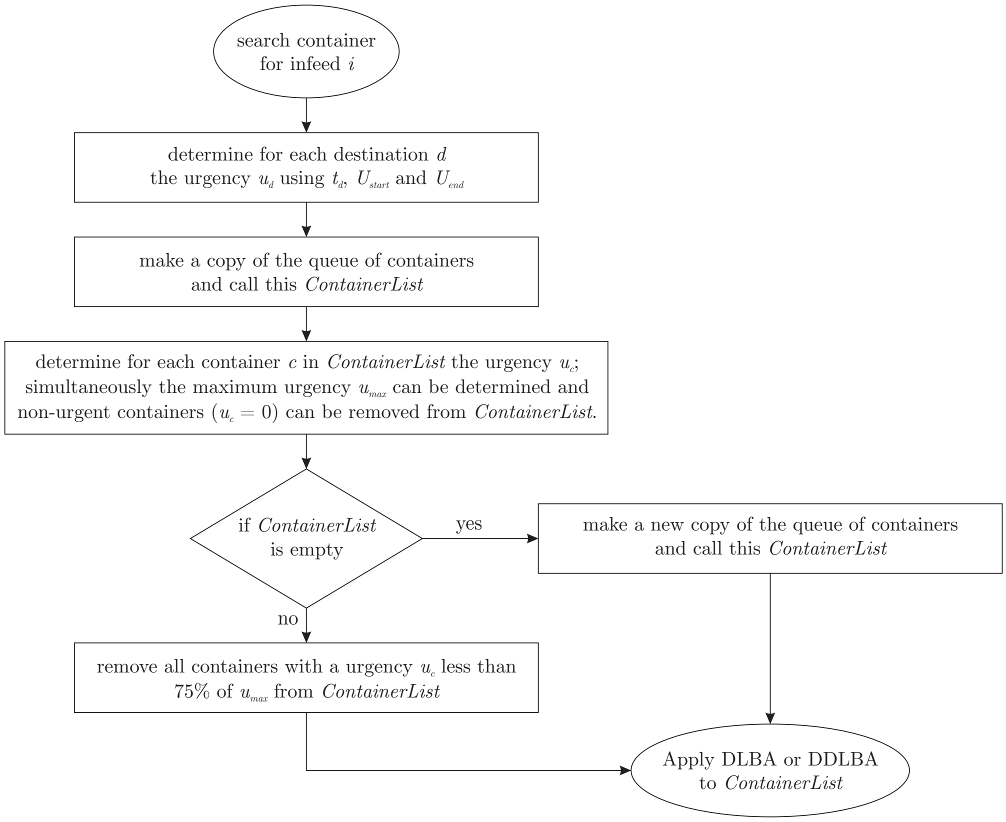


Figure 6.2: Flowchart for Priority Scheduling

tainers are considered by the scheduling approaches. Therefore all containers with an urgency less than $0.75 \cdot u_{MAX}$ are removed from the list, after which the list is sent to the DLBA or ADLBA algorithm to continue scheduling the inbound containers. That is, the DLBA and ADLBA assume that the only queueing containers are the containers that are present on the shortlist.

The algorithm to delay the containers is actually quite straightforward and starts as soon as a container arrives in the queue. In practice this would be when a container is announced, i.e. when the tug driver identifies the container. The algorithm than sequentially visits all destinations and determines whether the container holds items for the destination. If this is the case, the algorithm checks whether this destination is, or has already been, assigned to an outfeed. If so, the container is sent to the normal parking and announced to the dispatcher. If the destination is not yet assigned, the algorithm determines what the start up time for this destination is. Next, the algorithm sets the release time of the container to the minimum of the current container release time and the start up time of this destination.

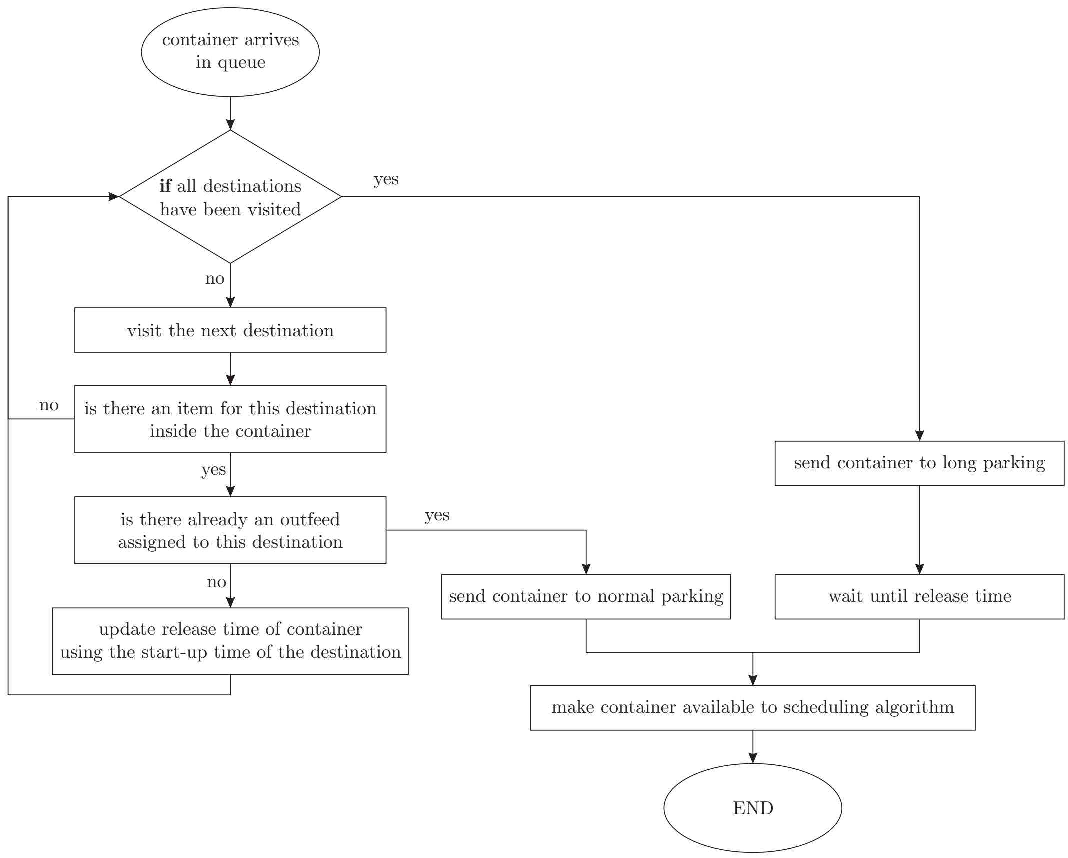


Figure 6.3: Flowchart for Delayable Scheduling

If all destinations have been visited, the container has either been sent to the normal parking, because there was at least one destination that was assigned to an outfeed, or the container release time has been set. If the latter is the case, the release time is increased by 1 hour, thus ensuring that there is enough time to sort all the items. The container is sent to the remote, long-term container park and there it waits until its release time. At that point in time, the container is announced to the dispatcher/scheduling algorithm, but it remains parked in the remote parking. Only when the dispatcher assigns the container to an infeed, it is removed from the container park.

The previous chapters focussed on four different scheduling approaches: current practice (FCFS), academic benchmark (ARB), state-of-the-art (DLBA), and the newly developed time dependent approach (ADLBA). Furthermore, this chapter discussed the development of two extensions to these algorithms. The next chapter therefore aims at determining how well these scheduling approaches and their extensions perform.

Chapter 7

Tests & Results

The previous chapters discussed existing scheduling approaches or focussed on the development of new ones. In order to determine whether the new approaches and extensions indeed improve the performance of the sorter systems, they have to be tested. Generally speaking, there are two approaches that could be used: simulation studies, which use software to mimic the behaviour of real systems, and real-life testing. Testing scheduling approaches in real-life situations is preferred, for it allows us to test using realistic sorter systems and problem data, but is impractical. Not only would this hamper operations at Vanderlande's customers, which is where realistic sorter systems can be found, it would also take months or even years to perform all tests required. We therefore choose to apply simulation studies.

Section 7.1 therefore starts by introducing the simulation models this research uses. Creating a simple sorter system is, with the software packages available, relatively easy. Focus is therefore on how complex features can be incorporated in the models, in order to ensure that they act in more or less the same way real-life sorter systems do.

The differences between the two industries also affect the 'problems' the sorter systems have to cope with. Section 7.2 therefore introduces four different scenarios, which describe the problem instances that sorter systems and scheduling algorithms should be able to handle.

Next, Section 7.3 introduces the criteria that will be used to determine which scheduling approach provides better results. Although many performance indicators are relevant for the parcel and postal as well as the baggage handling industry, some are tailored to measure the performance of a scheduling approach for a single industry.

The different scheduling approaches are tested using the models, scenarios, and performance indicators developed in the previous sections. Section 7.4 discusses the results of these tests.

Finally, Section 7.5 summarises the results, and concludes which approach should be preferred in which situation.

7.1 Simulation Models

One of the major advantages of simulation studies, is that a specific problem instance can be repeated over and over again, so that the performance of each individual approach can be determined for this specific problem instance. Another benefit is that many different problem instances can be simulated relatively quickly. For instance, simulating 1000 problem instances for a single approach, mimicking around 6 hours of sorting each, takes about 90 minutes.

The Applied Materials® AutoMOD™ software package is the industry standard for modelling automated material handling systems, and is therefore used for simulating sorter systems. The layouts and corresponding AutoMOD models are simplified representations of reality. It is, for instance, unlikely that sorter systems with three infeeds and three outfeeds can be found in reality. Simplification is used to keep the models fast and comprehensible, but, in order to be able to apply the results to real sorter systems, some features of real-life systems could not be omitted from the models.

First, baggage items and parcels cannot be transported when they are too close to each other. Therefore an empty space of at least 0.5 times the average length of two items is required between them, both when the conveyors are moving and when they are accumulating. As a result, an item can only be merged onto the main sorter when there is a gap of at least 2 times its length.

Second, a cascading transport capacity is implemented. That is, the rate at which items are placed on or removed from the infeeds/outfeeds is lower than the transport rate of the infeed/outfeed conveyors. This way, the infeed and outfeed conveyors act as a buffer between the workers that load/unload items on/from the sorter system relatively slow and the main sorter, which runs relatively fast. Buffering enables the accumulation of items on the infeed conveyors, and these accumulated items can enter the main sorter at a faster rate than the workers could load them on the sorter. Similarly, buffering makes it possible to sort items at a higher rate than the workers could remove the items from the sorter, until the entire outfeed conveyor is full.

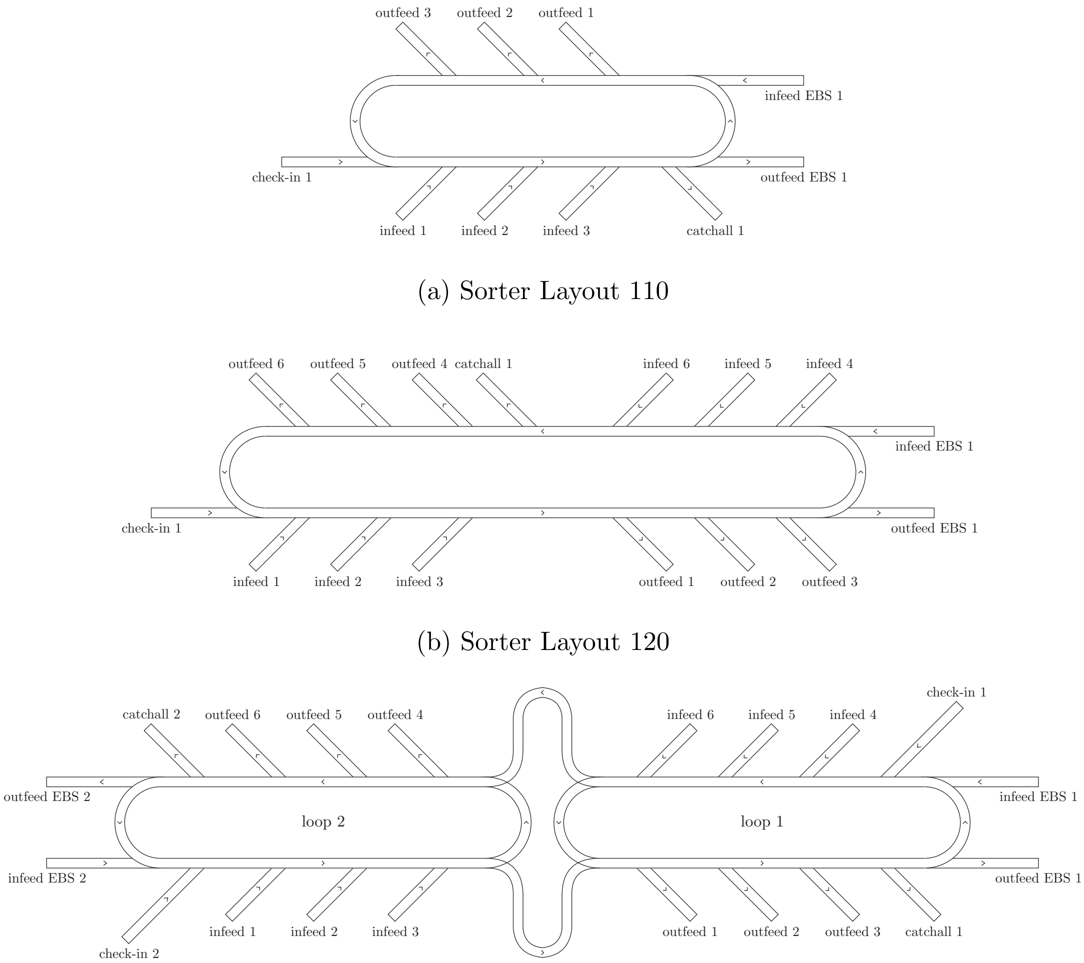
Third, two distinct reasons for recirculation can be identified. The most obvious reason is that the outfeed conveyor is physically full and can therefore not accommodate an additional item. Recirculation caused by this reason is what load balancing algorithms try to minimise; when all outfeeds are assigned the same number of items, the probability of one being full is reduced to a minimum. A more complex reason is that it is not possible to sort two items destined for the same outfeed that are close together on the main sorter. Due to the cascading transport capacity of the sorter systems, the main sorter runs at a higher speed than the outfeed conveyor, the two items would collide when transported to the outfeed conveyor. As a result, each time an item is sorted to an outfeed, this outfeed is *blocked* (that is, no item can be diverted to this conveyor) until the item has been transported far enough to prevent the collision of items. This time period will be referred to as *blocking time*.

Fourth and last, a simple on-line destination server is used to determine where an item should be sorted: at a specific outfeed, at the catch-all or at the EBS. As a result an item travels to the next chute at which point the simulation model determines whether this is the exit point for that item. If so, the software checks whether the outfeed is *available* (the outfeed is not full nor blocked by another item) after which the item is diverted from the main sorter to the outfeed. If the outfeed is not available the item continues on the main sorter. Due to the on-line destination server it is possible that an item that was destined for the EBS, but had to continue on the main sorter because of an unavailable outfeed, is sorted directly to the correct outfeed, because that outfeed was assigned to the destination whilst the item was forced to recirculate on the main sorter.

Unfortunately, not all features that are present in real-life systems could be incorporated in the simplified simulation models. The simulation models therefore allow for items to be merged onto the main sorter whenever there is a window of opportunity, whereas in reality a more intelligent, infeed workload balancing, allocation mechanism is used. Similarly, the early bag storage systems (EBSs) that are used are modelled in a very simple way. It is assumed that items that are diverted to the outfeed that leads them to the EBS do not hamper the main sorter, something that might occur when the crane is unable to store items fast enough. Providing sufficient buffer space between the main sorter and the EBS crane would, however, already prevent this problem from occurring. Furthermore, a very simplistic approach is used when retrieving items from the EBS: when a destination is assigned to an outfeed, all items in the EBS for that destination are moved to a virtual queue and the EBS starts retrieving them according to FCFS. When another destination is assigned to an outfeed, items for that destination join the back of the virtual queue. As a result, the EBS 'produces' only batches of items with the same destination, whereas in reality a mix of destinations might be achieved using a more intelligent retrieval algorithm.

Three simulation models, which are used to determine the performance of the different approaches, are developed based on the principles mentioned above:

- a single sorter in loop configuration with one infeed and one outfeed area, each consisting of three conveyors, one infeed for check-in baggage, and one EBS (Figure 7.1a);
- a single sorter in loop configuration with two infeed and two outfeed areas, each consisting of three conveyors, one infeed for check-in baggage, and one EBS (Figure 7.1b); and
- two sorters in loop configuration, each consisting of one infeed and one outfeed area, which, in turn, consist of three conveyors, one infeed for check-in baggage, and one EBS. The two sorters are connected through crossovers with a limited capacity (Figure 7.1c).



(a) Sorter Layout 110

(b) Sorter Layout 120

(c) Sorter Layout 22*c*

Figure 7.1: Layouts of Three Test Models

These three basic layouts are selected based on layouts that are frequently delivered by Vanderlande. In this research we will use the following tuple notation to identify layouts: number of loops, number of infeed and outfeed areas, specials. Using c to denote the crossovers and 0 to denote no specials, the three layouts mentioned above can be identified by the tuples 110, 120, and 22c. Appendix D discusses the reasoning behind these three models, explains model specific features, and underpins the selection of the various transport capacities. Table 7.1 provides an overview of the cascading transport capacities of the simulation models, for they are an important feature of high-capacity sorters.

items per hour	model 110	model 120	model 22c
infeed rate	400	400	400
capacity infeed conveyor	1200	1200	1200
capacity main sorter	3600	7200	3600
capacity outfeed conveyor	1200	1200	1200
outfeed rate	400	400	400
capacity check-in conveyor	1200	2400	1200
capacity EBS crane	400	800	400
capacity crossover	-	-	1200

Table 7.1: Cascading Capacities per Simulation Model

7.2 Model Scenarios

The previous section introduced the three simulation models used in this research. Just as important for simulation testing, or perhaps even more, are the datasets that describe what a single problem looks like. Because the problems in baggage handling are different from the problems in parcel and postal, e.g. the presence or absence of check-in flows or the size of the containers, both industries should be discussed separately.

A second division can be drawn based on the distribution of items inside one container. If a homogeneous distribution is assumed, the number of items inside one container for a specific destination is on average the same for all destinations. In parcel and postal, such a distribution could be a valid if the flow of items from every origin to every destination is more or less the same. Similarly, in baggage handling a homogeneous distribution could be applied if the number of transfer passengers is more or less the same for every flight. The alternative is the heterogeneous distribution, where some containers hold more items for destination a and others hold more items for destination b. This type of distribution can, for instance, be found in parcel and postal when a majority of the items from origin a should be delivered at b, but only a small portion should go to c or d. In baggage handling this type of distribution occurs when the baggage is segregated at the airport of origin.

Based on these classification criteria, four scenarios can be constructed:

- parcel and postal industry, homogeneous distribution (PP-even);
- parcel and postal industry, heterogeneous distribution (PP-uneven);
- baggage handling industry, homogeneous distribution (BHS-even); and
- baggage handling industry, heterogeneous distribution (BHS-uneven).

These scenarios can, in turn, be defined using up to eight parameters:

- the number of destinations;
- the number of containers that have to be scheduled;
- the interarrival times of these containers;
- the number of parcels inside one container;
- the *batchsize* (i.e. the number of containers that arrive at the same time);
- the number of container types;
- the distribution of items inside one container; and
- the time window of a container

Appendix E explains in detail which parameter values are selected and why they are suitable. The main consideration for these values is that the sorters should be loaded at about 65% of their capacity. Although this is slightly higher than one might perceive in reality, especially in baggage handling such heavy workloads are uncommon, it forces the sorter systems to work at the height of their powers. This ensures that differences between the assignment algorithms become much more evident than would be the case if a normal workload would be applied. Table 7.2 provides an overview of the selected values for the first six parameters.

Special attention should be paid to the process that is used to generate the check-in baggage items. It is important that no baggage items are generated after the cutoff time of a destination. Similarly, not too many baggage items should arrive extremely early, as very few passengers arrive at the airport four hours before departure. Appendix E therefore also discusses the approach that is used to generate check-in baggage items.

The last subject that should be discussed with respect to the scenarios, is the number of problem instances that should be generated for each simulation model – scenario combination. From a statistical perspective as many simulation runs as possible should be performed. This reduces the variability of the results and increases the certainty with which it can be stated that a is better than b. However, simulation runs are computationally expensive, which is why, from a practical point of view, fewer simulation runs are better. Although a fixed number of replications could be selected, after which the precision of the answer can be computed, it seems better to aim for a specific precision and determine how many replications are needed to obtain this precision (Law & Kelton, 2000). Therefore the relative error approach as suggested by Law and Kelton is applied, using 20 initial runs, an accepted relative error of 5%, and a confidence interval of 95%. The results show that for the parcel and postal industry scenarios 50 replications are sufficient for all the simulation models. For the baggage handling scenarios, however, the

		PP-even	PP-uneven	BHS-even	BHS-uneven
destinations [#]		3	3	6	6
containers [#]		25	25	70	70
interarrival time [sec]	lb	250	250	900	900
	ub	325	325	1500	1500
parcels in container [#]	lb	100	100	20	20
	ub	150	150	30	30
batchsize [#]	lb	1	1	4	4
	ub	1	1	6	6
container types [#]		1	4	2	8

(a) Model 110

		PP-even	PP-uneven	BHS-even	BHS-uneven
destinations [#]		6	6	12	12
containers [#]		50	50	150	150
interarrival time [sec]	lb	100	100	800	800
	ub	188	188	1400	1400
parcels in container [#]	lb	100	100	20	20
	ub	150	150	30	30
batchsize [#]	lb	1	1	8	8
	ub	1	1	10	10
container types [#]		1	7	2	14

(b) Model 120

		PP-even	PP-uneven	BHS-even	BHS-uneven
destinations [#]		6	6	12	12
containers [#]		50	50	150	150
interarrival time [sec]	lb	100	100	800	800
	ub	188	188	1400	1400
parcels in container [#]	lb	100	100	20	20
	ub	150	150	30	30
batchsize [#]	lb	1	1	8	8
	ub	1	1	10	10
container types [#]		1	7	2	14

(c) Model 22c

Table 7.2: Scenario Parameters per Simulation Model

results show that up to 3000 replications would be necessary for the model 110 – BHS-uneven combination. Therefore the accepted relative error, for this specific model - scenario combination, is increased to 10%. The conclusion is that, for the baggage handling scenarios, model 110 requires 1000 replications, model 120 requires 500, and model 22*c* requires 750 replications. Appendix F provides a more detailed description of the approach that is used to determine the number of replications.

7.3 Performance Indicators

Because of the differences between the two industries, it is not possible to define one single key performance indicator (KPI). There is, however, in both industries a clear notion of what defines a better solution.

Section 1.2 already mentioned one of the most important issues in baggage handling: ensuring that baggage makes it to its destination together with its owner. The direct costs of failing to do so can easily exceed the 100US$ (SITA, 2010) and within Vanderlande figures of around 300 euros are also known. Besides the direct costs of delivering the baggage at a location desired by the customer, airlines incur indirect costs due to image damage. As a result, airlines pay special attention to baggage delivery accuracy and, consequently, airports are mainly interested in one aspect of baggage handling systems: the number of bags that do not make their flight as a result of a failing sorter system, also known as *missorted bags*. Note that the last part is an important addition, because it explicitly excludes baggage that did not make its flight due to late arrivals or late check-in. In order to be able to compare the results from different approaches, a relative performance indicator is necessary. Therefore, the *missort rate*, the number of missorted bags per thousand bags, is used.

In parcel and postal, however, focus is on a different topic. Whilst in baggage handling the costs of delivering a missorted bag are significant, these costs are virtually not present in the parcel and postal industry. There, an item that missed its connection can be delivered by a trailer or aeroplane that was already scheduled to travel to that location at a later time. The only additional cost an express parcel company has, is the penalty that has to be paid when the service level agreement is breached. As a result, their concern is mainly on ensuring that the fixed costs of the sorter system can be divided over as many parcels as possible, that is, their focus is on *throughput*. Throughput is generally defined as the number of correctly sorted parcels per hour. However, the measured throughput can be influenced significantly by the *warming-up* and *cooling-down period* (Law & Kelton, 2000). The warming-up period is when the sorter system is being filled, that is, parcels are already loaded on the sorter system but there are hardly any parcels leaving the sorter system. Or, more generally, the infeed rate is much higher than the outfeed rate. The cooling-down period is exactly the opposite, there are no containers left to be unloaded and the sorter system is being drained. When the throughput is determined for continuous production systems, these periods should be excluded. They occur only rarely and incorporating them would thus distort the actual figures. The parcel and postal industry, however, often sorts the parcels in *shifts*, which are x-hour sorting periods in which either inbound or outbound parcels are sorted. As a result, these warming-up and cooling-down periods form an important part of the sorting performance within a shift and should therefore be included.

Besides the KPIs described above, other performance indicators (PIs) can be identified. Where the KPIs describe the criteria that are generally used to determine which solution is best, the PIs give an overview of the effects on the surroundings. In case of a draw between two approaches based on the KPIs, the scores on the PIs can be used to determine the best solution. Furthermore the PIs can provide valuable information about the effects of the scheduling approaches on other business processes.

The first and second performance indicator are related to the containers that are waiting to be unloaded. In many cases these containers have to be loaded again to be sent to another location. At express parcel companies, the incoming trailers are unloaded and become outgoing trailers; similarly, at airports, the incoming ULDs are emptied and become outgoing ULDs. It might therefore be important that containers do not have to wait too long before they are unloaded, in order to ensure that they can be used as outbound containers. Furthermore, there should not be too many containers waiting simultaneously, because of the limited space available at the car park. The question what is too long and what are too many containers is outside the scope of this research, but we do provide two performance indicators that keep track of these issues. The *average container waiting time*, measured in minutes, specifies how long a container has, on average, been waiting on the yard, i.e. the time between the arrival of the container and the moment it is assigned to an infeed. The *maximum number of waiting containers* specifies the maximum number of containers waiting in the yard simultaneously, something that is of interest when determining the size of the car park.

The third indicator is the *recirculation rate*, which is the average number of recirculations per item (rpi). Because an item can recirculate for two reasons, either an outfeed is full or it is blocked, separate statistics are recorded for both of them.

The fourth and final performance indicator focusses on the EBS systems and is therefore only relevant for the baggage handling industry. The main goal of the delaying scheduling algorithm (Section 6.2) is to reduce the required EBS space by temporarily parking ULDs at a remote location. It is therefore important to keep track of the *maximum number of items in the EBS*, as this defines the required size of the EBS.

performance indicator	unit		PP	BHS
Throughput	items per hour	iph	√	
Missort rate	items per thousand	‰		√
Avg container waiting time	minutes	min	√	√
Max number of waiting containers		#	√	√
Recirculation rate	recirculations per item	rpi	√	√
Max number of items in EBS		#		√

Table 7.3: (Key) Performance Indicators

Table 7.3 summarises this section and provides a full overview of the (key) performance indicators and their measurement units. Furthermore, it shows the relevance of the performance indicators for each of the industries.

7.4 Results

When discussing the results from simulation studies, it is important to check whether there are differences between the approaches that have been applied. However, it is even more important to determine whether the differences that can be found are significant. Unfortunately, the term significant is somewhat ambiguous, because two types of significance can be distinguished: statistical significance and operational significance.

In this research a difference is considered to be statistically significant, if the H_1 hypothesis is accepted in a pairwise two-tailed t-test with a 95% confidence interval and $n-1$ degrees of freedom. Appendix G explains this type of hypothesis testing, and the interestered reader is referred to the work by Law and Kelton (2000) for more information.

Statistical significance, however, is not by definition relevant from an operational perspective. It is quite well possible that there is a statistically significant difference of 1 item per hour (iph) on the throughput KPI. In practice, however, this difference is not considered to be relevant, as the order of magnitude of throughput is hundreds or thousands of items per hour. On the other hand, a decrease of 1 permillage point on the missort rate KPI might not seem much, but if the average missort rate is only 5‰, the improvement is operationally significant indeed. Therefore, when the absolute difference is mentioned, the relative difference is presented between brackets. It is important to realise that a difference can only be significant from an operational point of view, if (and only if) the difference is statistically significant. In this research we will therefore use the phrase 'operationally significant' to indic-

industry	scenario	FCFS	DLBA	ADLBA	DLBA-pd	ADLBA-pd
PP	even			√	NA	NA
	uneven		√		NA	NA
BHS	even					√
	uneven					√

Table 7.4: Best scheduling approach per scenario

ate that the difference between two results is both statistically significant and large enough to be interesting.

The results of the simulation studies are discussed for each industry separately, for the nature of the problems can be quite different. Furthermore, the extensions developed in Chapter 6 are only valid for the baggage handling industry. This section only contains the essentials of the results, a full overview of the performance of the scheduling approaches on all indicators can be found in Appendix H. The boxplots that are used, have two 'whiskers' that indicate the minimum and maximum value of the 95% confidence interval. The red '+' symbols indicate outliers, i.e. results that are outside the confidence interval. Finally, Table 7.4 provides an overview of the best scheduling approaches.

Parcel & Postal

For the evenly distributed scenario in parcel and postal, it is likely that there is very little difference between the scheduling approaches. After all, there is, on average, no difference between the containers and selecting a less suitable container will hardly affect the performance of the sorter system. Also the randomness, caused by the fact that only the contents of and not the sequence in which the contents are unloaded from the container are known, is likely to have a greater affect on the performance than the scheduling approach used.

The simulation studies show that for model 110 and 120 there is indeed very little difference between the scheduling algorithms. In fact, for neither of these simulation models there is a statistical difference between any of the scheduling approaches, suggesting that each approach is as good as another. Furthermore, Figures 7.2a and 7.2b show that the distribution of the throughput is also more or less similar. Figure 7.2c, however, shows that for model 22*c* a clear difference can be found. The ADLBA approach outperforms all others, with a throughput that is approximately 25 items per hour (1.5%) higher. The original DLBA, however, performs statistically just as well as FCFS and ARB and is therefore not considered to be an improvement.

It is quite hard to find a satisfying explanation for this behaviour, but most likely this behaviour is related to the infeed assignment problem. FCFS, ARB, and the DLBA assign a container to an available infeed, irrespective of its location on the sorter system. The ADLBA, however, assigns the containers to infeeds that are selected based on the load balancing criterion. A possible explanation for the perceived behaviour would thus be that the ADLBA balances the workload over the two separate sorters more evenly, by applying a better infeed assignment algorithm.

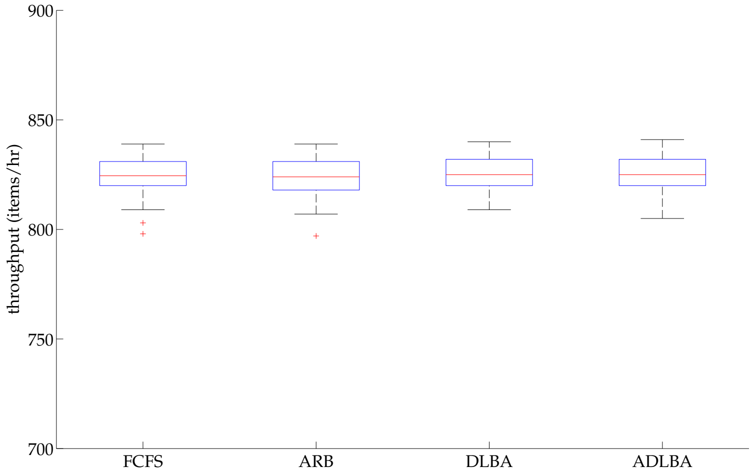
One would expect that this improvement in workload balancing leads to less recirculation, but the results show that this is not necessarily the case. In fact, the ADLBA has, on model 22*c*, statistically larger recirculation rates than all the other scheduling approaches. Compared to FCFS and DLBA the increase is 11.4 percentage point (7.5%) and 13.4 percentage point (8.8%) respectively. A possible explanation would be that the ADLBA is able to keep the sorters filled up completely, which increases the recirculation rates, but also leads to an increased throughput.

The simulation studies also show that there sometimes is a statistical difference in the maximum number of containers that are waiting in the yard. These differences are, however, hardly operationally significant, for they are 0.78 container at the most. Results that are potentially interesting are the differences in average container waiting time in model 22*c*. Here, the ADLBA reduces the average waiting time by a minute and a half (4.5%) compared to the DLBA and the reduction is just over two and a half minute (7.3%) compared to FCFS.

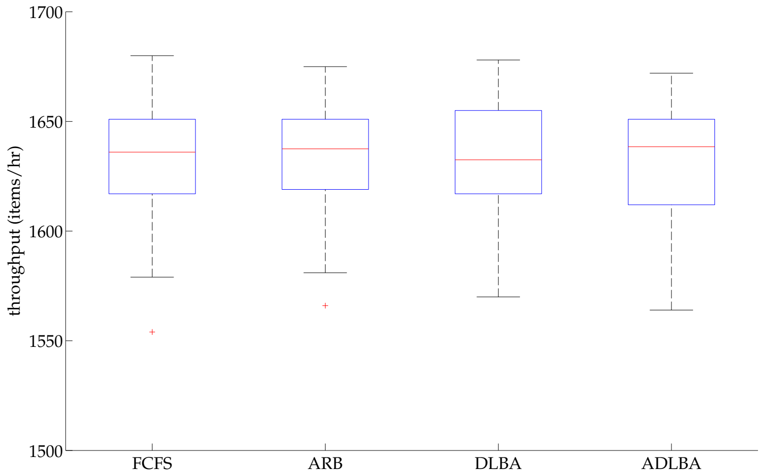
For the unevenly distributed parcel and postal scenario, the differences between scheduling approaches are likely to be much more obvious. The clear difference in contents between containers should provide load balancing algorithms with opportunities to select better containers than non load balancing algorithms would do.

From the simulation studies, we conclude that the load balancing algorithms (DLBA and ADLBA) do indeed outperform the existing approach (FCFS) and the academic solution (ARB) for all the simulation models (Figure 7.3). Particularly the DLBA proves to be an interesting approach, given that it outperforms FCFS by 11, 52, and 63 items per hour (1.4, 3.7, and 4.5%) for models 110, 120, and 22*c* respectively. Also interesting is that the original DLBA proves to be a better scheduling approach then the newly developed ADLBA for all models, which is a conclusion that contradicts the outcome of the PP-even scenario. The differences between the DLBA and ADLBA are, however, mainly important from a statistical point of view. For model 110, 120, and 22*c* the differences are respectively 6, 15, and 18 items per hour (0.8, 1.0, and 1.3%).

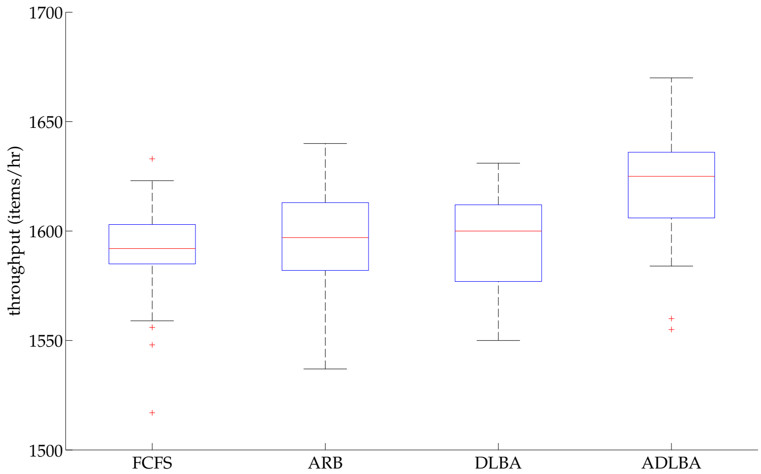
The fact that the DLBA and ADLBA outperform the other approaches shows that load balancing techniques can indeed improve the performance of sorter systems. Load balancing can therefore be appointed as the explana-



(a) model 110



(b) model 120



(c) model 22c

Figure 7.2: Throughput using the PP-even Scenario

tion for the difference between FCFS and ARB on one side and DLBA and ADLBA on the other. The difference between the DLBA and ADLBA is most likely caused by the difference in the level of detail that the approaches use. The DLBA, a quite rough approach, provides better results if the differences between containers are quite significant, whereas the ADLBA, a more detailed technique, proves to be better when the differences between containers are more subtle.

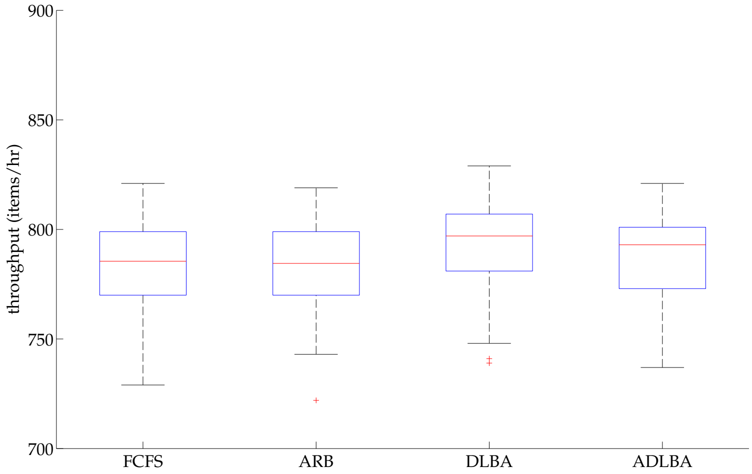
The simulation studies also show that some interesting results can be achieved regarding the waiting containers (trailers). The DLBA is able to reduce the maximum number of containers in the queue, compared to FCFS, by 0.5, 2.5, and 2.7 (4.8, 11.3, and 12.5%) for model 110, 120 and 22*c* respectively. Especially the results of the latter two models are interesting, they raise the suggestion that significant reductions in required yard space could be achieved. Furthermore, the results show that the DLBA is able to significantly reduce the average waiting time of a container. Particularly the reductions for model 120 and 22*c*, approximately 10 minutes and just over 20% of the original waiting time, are impressive. For the maximum number of waiting containers as well as the average waiting time per container, the ADLBA performs just under, with reductions approximately half of those achieved by the DLBA.

Baggage Handling

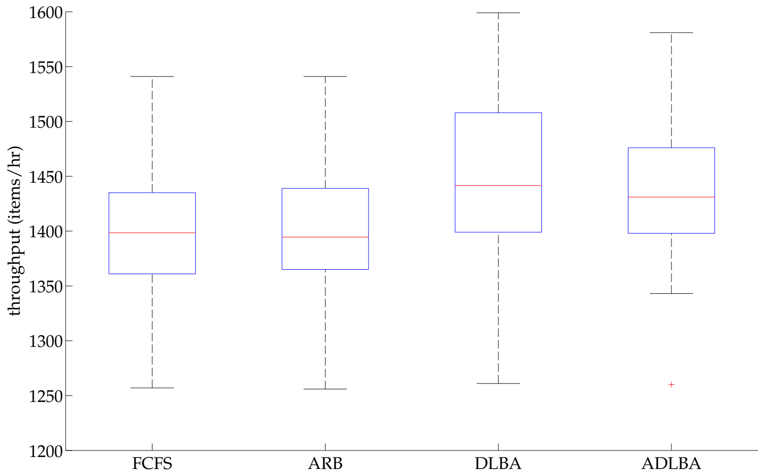
More scheduling algorithms have been tested on the baggage handling scenarios than on the parcel and postal scenarios, because now also the extensions have been examined. In Figures 7.4 and 7.5 and the discussion below, the suffixes ‘p’ and ‘d’ are used to indicate the priority and delayability extension respectively.

For the evenly distributed baggage handling problem, one might expect results similar to those of the evenly distributed parcel and postal problem. There are two reasons why this is not necessarily the case. First, although both scenarios are called evenly distributed, the contents of the containers are not alike. The differences in destinations of the items and the number of items in a single container make that the effects of selecting the wrong container are much more serious in the baggage handling scenario. Second, focus is now on missort rate, which is a different performance indicator altogether. It is therefore difficult to predict the results of the baggage handling scenarios.

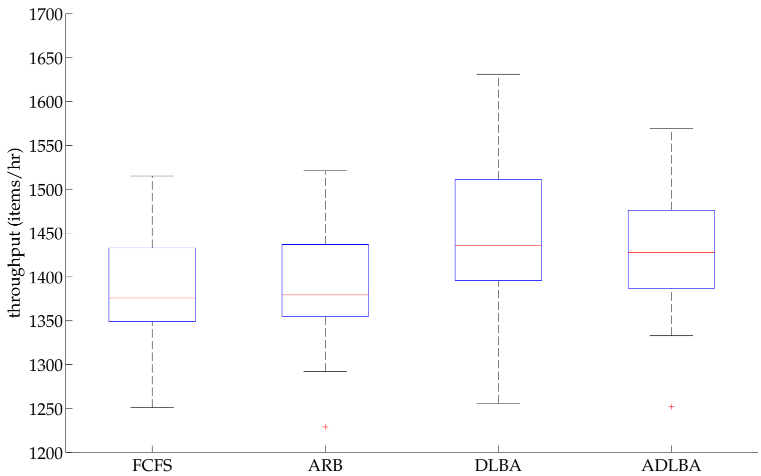
The difference between the PP- and BHS-even scenarios becomes evident immediately, for the differences between the scheduling approaches are statistically as well as operationally significant. Focussing on the four scheduling approaches makes clear that in model 110 the ADLBA approach is preferred, for it outperforms DLBA by 0.4 permillage point (7.0%). The difference of 0.04 permillage point between the ADLBA and FCFS is statistically significant,



(a) model 110



(b) model 120



(c) model 22c

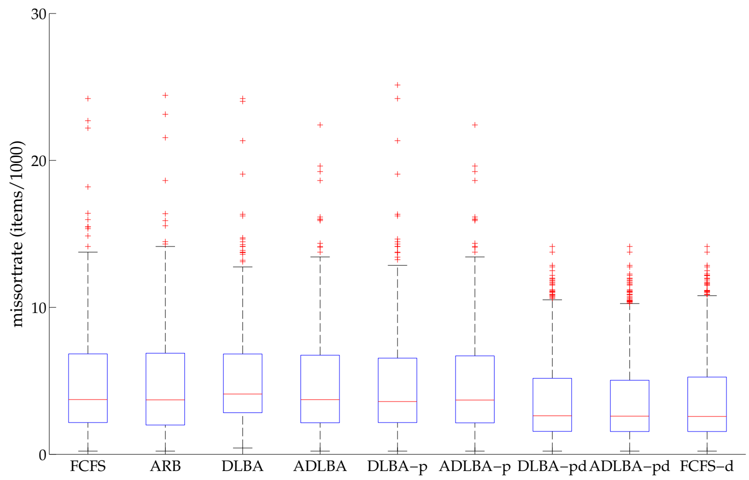
Figure 7.3: Throughput using the PP-uneven Scenario

but can hardly be considered operationally significant. In model 120 FCFS is the best approach, it outperforms DLBA by 1.7 permillage point (2.5%), but the ADLBA has little to learn from it as it outperforms the DLBA by 1.4 permillage point (2.0%). The difference between FCFS and the ADLBA is, however, not statistically significant. Finally, in model 22*c* the roles reverse, for the DLBA is now the better approach. It outperforms FCFS and the ADLBA by 2.8 and 2.5 permillage point (18.4 and 16.3%) respectively. Regarding the other performance indicators the results are quite clear, none of the differences is significant from an operational point of view.

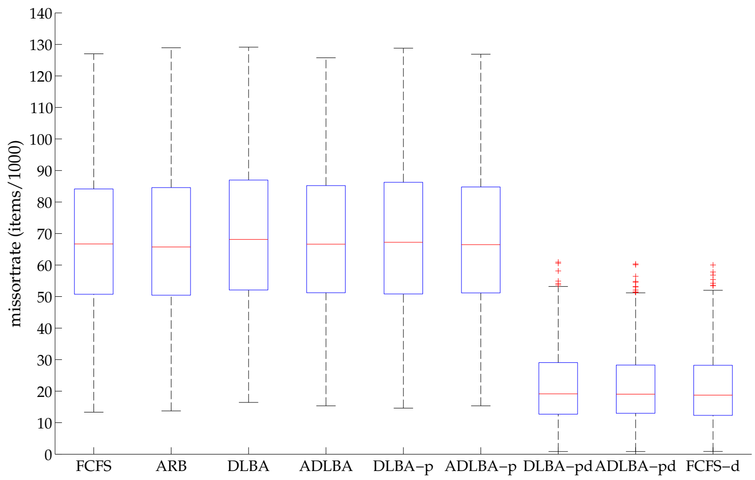
The explanation used for the parcel and postal industry might be applicable in this case as well. The ADLBA is a more subtle approach and therefore appears to be suitable for smaller problems where the workload is not particularly high. The DLBA on the other hand provides better results for the very complex sorter system. It is, however, important to realise that the differences in model 110 and 120 are hardly operationally significant. That is, although the ADLBA and FCFS provide better results for these models, it would hardly be worth the effort to switch approaches.

The priority extension improves the results of the DLBA and ADLBA approach. The differences between approaches with and without the ‘p’-extension are, however, often only statistical and not operational significant. The biggest improvement in terms of percentage is the difference between the DLBA and DLBA-p in model 110, which is only 0.4 permillage point (8.3%). The largest absolute improvement is 0.8 permillage point (6.1%) between the ADLBA and ADLBA-p in model 22*c*. Unfortunately the difference between FCFS and the best performing workload balancing algorithm is not operationally significant for model 110 and 120. In model 22*c*, however, the potential improvement has been enlarged. That is, the difference between FCFS and the DLBA-p algorithm is 3.6 permillage point (23.4%), which is quite an improvement indeed. Again, the results show no operationally significant differences on any of the other performance indicators.

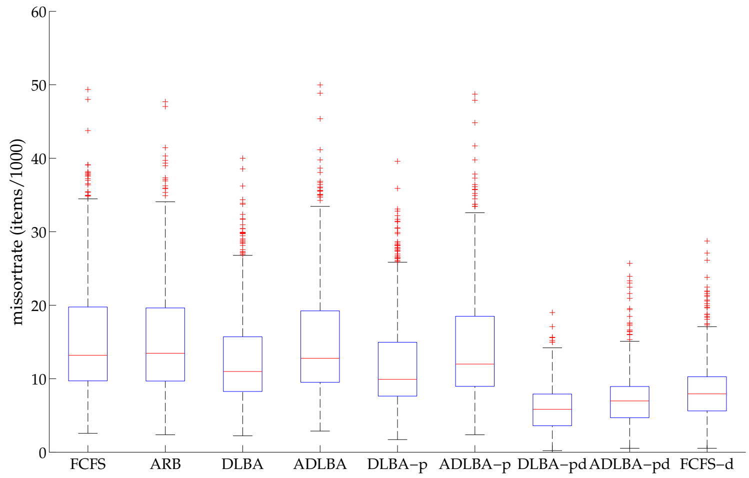
Including the ‘d’-extension tremendously improves the performance of the sorter systems. This was expected, for it reduces the workload of the sorter system. Generally speaking, this extension reduces the missort rate by 1.0, 47.6, and 7.3 permillage point (20.8, 68.7, and 50.1%) on model 110, 120, and 22*c* respectively, when compared with the best performing approach until now. The extreme increase in performance in model 120 is partly due to the unrealisticly high missort rates, which are caused by a relatively high workload. However, in all model – scenario combinations the workload is equal to 70% of the outfeed capacity. Such workloads rarely occur in practice, but magnify the effects of the different scheduling approaches. Applying normal workload would result in missort rates that are 1 ‰ at the most, making it virtually impossible to analyse the differences between the scheduling approaches. Furthermore, the high missort rates suggest that this type of sorter systems is extremely vulnerable to heavy workloads and would therefore benefit the most



(a) model 110



(b) model 120



(c) model 22c

Figure 7.4: Missort rate using the BHS-even Scenario

from more intelligent scheduling approaches.

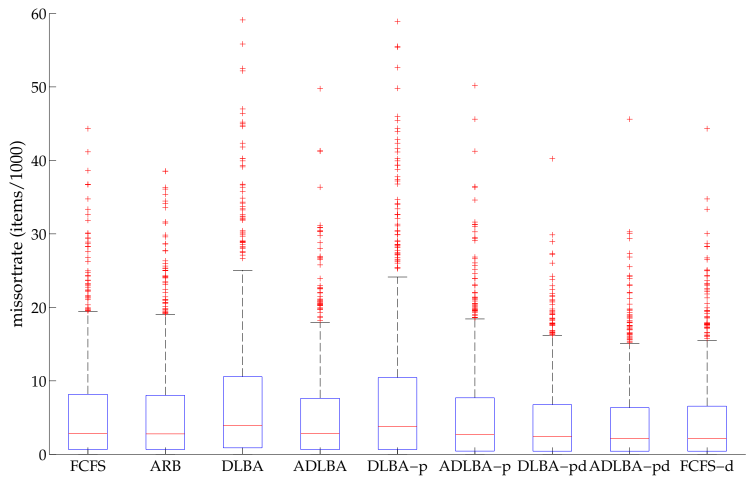
The 'd'-extension also affects the other performance indicators. It reduces, for instance, the required EBS space by 96, 230, and 204 items (approximately 30% in all cases) for models 110, 120, and 22c respectively. In order to achieve this improvement, the algorithm sends on average 5.6 containers in model 110 and 12.2 in model 120 and 22c to the remote parking. Furthermore the simulation studies show that, compared to FCFS, the number of waiting containers can be reduced by approximately 2.5 (10%) containers and the average container waiting time is reduced by 1 and 3 minutes (19.1 and 18.3%) for model 120 and 22c respectively. Interesting is that model 110 shows behaviour that is exactly opposite. Here the number of containers and the average waiting time per container increases by 1.3 (17%) and 2.3 minutes (50.3%) respectively. A possible explanation is that the problem data for this simulation model is such that the dispatcher is tempted to select containers from the remote parking, thereby increasing the waiting time for containers on the ordinary parking.

The analysis raises the suggestion that applying the 'd'-extension in combination with FCFS might provide results that are comparable to the more complicated scheduling techniques. The simulation studies show that this is indeed the case for model 110 and 120. In model 22c, however, the DLBA-pd outperforms FCFS-d by 2.4 permillage point (28.8%) on missort rate, 2.7 containers (14.9%) on the number of waiting containers, and 1 minute (18.3%) on container waiting time. These are operationally signficant differences indeed.

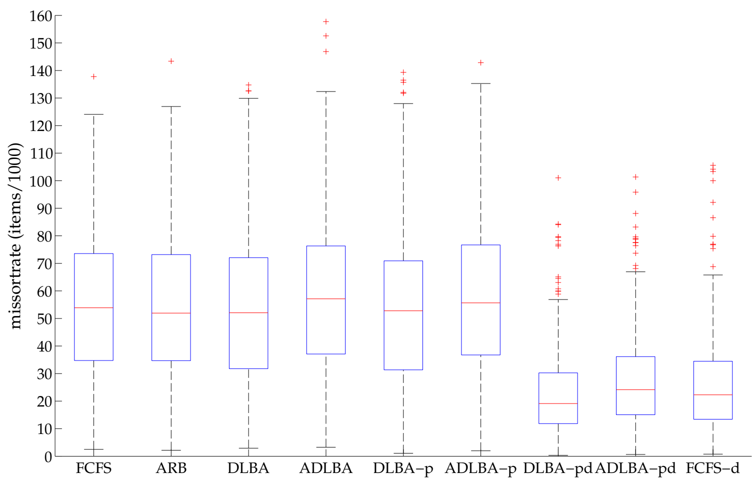
In the unevenly distributed baggage handling scenario the differences between the scheduling approaches become much more evident. In model 110 both FCFS and ADLBA outperform the DLBA by approximately 2.1 permillage point (28.5%), although the difference between the two is not statistically significant. Interesting is that for model 120 and 22c the DLBA is the best performing approach; differences compared to FCFS are 2.0 and 4.6 permillage point (3.6 and 29.5%) respectively. The ADLBA is the least suitable approach and is outperformed statistically as well as operationally by FCFS. All other performance indicators show no operational significant difference, except in model 22c. There, the DLBA reduces the maximum number of waiting containers and the average container time, compared to FCFS, by 2.3 containers (9.0%) and 2.0 minutes (14.7%) respectively.

The explanation for these results is probably the same as the one used for the evenly distributed baggage handling scenario: the ADLBA is a subtle approach that performs best in situations where the differences are marginal, whereas the DLBA performs better in situations where the differences between the possible decisions are much more evident.

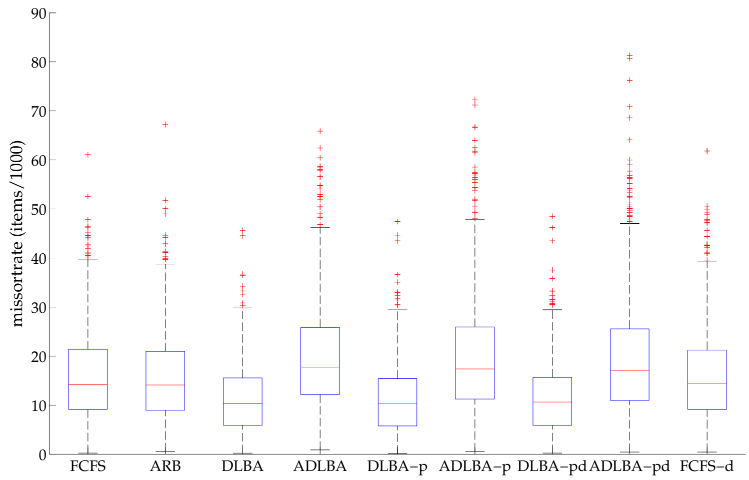
Including the 'p'-extension improves the results, but only marginally. That is, the difference between FCFS and the best performing load balancing ap-



(a) model 110



(b) model 120



(c) model 22c

Figure 7.5: Missort rate using the BHS-uneven Scenario

proach, ADLBA-p in model 110 and DLBA-p in model 120 and 22*c*, is approximately 0.2 permillage point larger. The difference between the original scheduling approaches and the approaches with the ‘p’-extension is, however, not statistically significant. Furthermore, the results for the other performance indicators are similar to those discussed before.

The inclusion of the ‘d’-extension has, again, a positive effect on the performance of the scheduling approaches. In model 110 the ADLBA-pd is clearly the best approach, and outperforms FCFS by 1.7 permillage point (19.6%). In model 120 and 22*c* the DLBA-pd is the better scoring approach, outperforming FCFS by 33.1 and 4.4 permillage point (59.0 and 27.9%) respectively. Not only the key, but also the other performance indicators are affected by the delayability extension. Similar to the results of the BHS-even scenario, FCFS is able to reduce the maximum number of waiting containers and average container waiting time in model 110 by 1.5 containers (17.5%) and 2.2 minutes (32.3%) respectively. Based on these performance indicators, the DLBA-pd is also the best approach for model 120 and 22*c*. Compared to FCFS the maximum number of waiting containers is reduced by 4.4 and 1.7 containers (27.3 and 6.9%) respectively and, in addition, the DLBA-pd reduces the average container waiting time by respectively 2.5 and 2.9 minutes (44.6 and 20.5%). Furthermore the simulation studies show that both workload balancing approaches are able to reduce the required EBS space by 97, 213, and 198 items (approximately 40%) for model 110, 120, and 22*c* respectively, by sending on average 5.6, 12.2 and 12.2 containers to the remote parking.

Because the improvements caused by the inclusion of the ‘d’-extension are quite significant indeed, simulation studies using the FCFS-d scheduling approach have been performed. These studies show that there is no statistical significant difference between FCFS-d, DLBA-pd, and ADLBA-pd in model 110. In model 120 and 22*c*, however, the DLBA-pd is still the better scheduling approach and outperforms FCFS-d by 3.0 and 4.6 permillage point (11.6 and 28.4%) respectively.

7.5 Conclusion

The results show that the newly developed ADLBA is only interesting for parcel and postal sorter systems where the distribution of items over the destinations is more or less identical for each origin. In that case an increase in throughput, although limited, can be achieved for more complex sorter systems. In relatively simple sorter systems, however, the ADLBA is not likely to contribute to the performance, but it also does not counteract it. As soon as the item distribution is more diverse, the original DLBA outperforms not only the current practice FCFS, but also the newly developed ADLBA in all simulation models.

This difference is probably caused by the difference in subtlety between the

DLBA and ADLBA. It is probable that the ADLBA uses an approach that is too detailed, thereby 'overscheduling' the problem. It still does, however, provide results that are better than current practice. This shows that the newly developed approach is to be preferred over current practice, although in some cases more could be gained by applying the original DLBA approach.

For both baggage handling scenarios the results show that the workload balancing approaches DLBA and ADLBA do indeed improve the performance of baggage handling systems. Again the ADLBA is the preferred solution when the differences between decisions are only marginal, i.e. small sorter systems and containers that are much alike. The DLBA is preferred for situations where the differences are much more obvious, i.e. larger and more complicated sorter systems and large differences between containers.

Even more interesting are the conclusions regarding the extensions that have been developed. The effects of the priority extension are only marginal, for the differences are often not operationally significant. The delayability extension, however, shows that impressive improvements on all performance indicators are possible. These results also show that applying a FCFS policy in combination with a 'd'-extension is, in many cases, as good as applying one of the workload balancing approaches in combination with a 'd'-extension. Only for complicated sorter systems such as model 22c does the DLBA-pd approach provide results that significantly better than those of FCFS-d.

Summarised, both dynamic load balancing approaches provide interesting solutions to the inbound container assignment problem. In some cases they clearly outperform the current practice FCFS, whereas in other cases FCFS is as good as, or even better, than the workload balancing approaches. The application of the delayability extension in the baggage handling industry is interesting as well, for it could significantly improve the performance of sorter systems. Based on these provisional conclusions and the issues discussed in earlier chapters, the next chapter draws the final conclusions.

Chapter **8**

Conclusion

Chapter 1 introduced the purpose of this research using two research goals. This chapter focusses on determining whether these goals have been attained and provides suggestions for further research. It starts, however, by briefly restating the background of the problem.

The high-capacity sorter systems that are the focal point of this research are used by, amongst others, express parcel companies, airports and airlines. When determining which container (trailer or ULD) should be unloaded next, they often apply a first-come-first-served (FCFS) approach. However, quite often the contents of individual containers are known, knowledge that could be used when determining which container should be unloaded. Focus of this research was therefore to develop a more intelligent approach that uses this knowledge when assigning containers to infeed locations.

The primary goal of this research was therefore defined as the *'development of an approach that balances workload on the outfeeds of a sorter system, by assigning containers to infeeds using the conveyable destination data'*. The literature review (Chapter 3) provided an already existing workload balancing approach, the dynamic load balancing algorithm (DLBA) by McWilliams (2009a). Because this approach has some shortcomings, a new workload balancing approach, the advanced dynamic load balancing algorithm (ADBLA) has been developed (Chapter 5).

The results of simulation studies show that load balancing approaches can indeed improve the performance of sorter systems, but the extent of the improvement is highly situation-dependent. When the contents of containers are much alike, FCFS is often as good as any other approach. Only for complicated sorters systems with crossovers, the simulations studies showed that the ADBLA is able to improve the throughput by 1.5%. If, however, there are large differences between the contents of containers, the DLBA proves to be the best scheduling approach and increases the throughput by up to 4.5% or 63 items per hour. Interesting is that the improvement of performance amplifies with increasing complexity. Furthermore the results show that if the ADBLA or DLBA is able to improve the throughput, it also reduces the number of waiting containers and the average container waiting time. Again, for the most complex model the improvements are the largest, for the DLBA reduces the queue of containers by 2.7 (12.5%) and the average waiting time by 10 minutes (20%).

For the baggage handling industry the number of items that does not make their flight (missort rate) is much more important than throughput. However, also in this case workload balancing approaches are able to improve performance, although, again, only for larger or more complicated sorter systems. The DLBA, for instance, is able to reduce the missort rate in a complex sorter system with containers that are alike by 2.8 permillage point (18.4%) compared to FCFS. When the contents of containers are more diverse, improvements up to 4.6 permillage point (29.5%) compared to FCFS are possible. In that case it is also possible to reduce the number of waiting containers by 2.3 (9.0%) and the average container waiting time by 2 minutes (14.7%).

Because in baggage handling some containers are considered to be urgent, for they contain items that should make a flight that leaves soon, extensions to the existing scheduling approaches that use this information are preferred. The second research goal was therefore the *'development of extensions to the workload-balancing-approach, which could further improve material handling processes using information on conveyable destination and yard layout'*. This resulted in two scheduling extensions (Chapter 6), which can be applied to most inbound container assignment algorithms. The first extension ensures that containers with items that are due to leave soon receive priority over other containers. The second extension sends containers that contain only baggage items that are not due in several hours to a remote parking, thereby reducing the workload of the sorter system during peak hours.

Unfortunately, the effects of the priority extension are only marginal and therefore not significant. The most likely explanation for this, somewhat disappointing, result is that the extension only affects containers that are extremely urgent, and these containers are quite rare. Furthermore, the selection of urgent containers leads to a less balanced workload, because it limits the selection possibilities of the main workload balancing algorithm.

The effects of the delayability extension, on the other hand, are impressive. The workload balancing approaches in combination with the delayability extension reduce the missort rates by 20 – 70%. Besides, they reduce the need for early baggage storage space by approximately 30%, which makes them quite a money-saver. Most of these improvements, however, are caused by the delayability extension itself, for if the current practice FCFS is combined with this extension, it performs equally well in half the cases. For more complex sorter systems, the DLBA continues to be the best approach, because its missort rates are several permillage points lower.

There are basically two final questions that should be answered in this conclusion. First, do the proposed scheduling approaches and extensions indeed lead to improved performance? and second, is the research direction of these approaches a fruitful one?

The results discussed above show that the existing workload balancing approach DLBA by McWilliams (2009a) improves the performance of sorter systems. Especially when the contents of containers are not alike the DLBA can significantly increase throughput and reduce missort rates. The newly developed ADBLA, however, only appears to improve the performance in one specific instance, namely a complex sorter system with crossovers combined with a parcel and postal scenario where containers are more or less the same. A likely explanation for this conclusion is that the ADBLA is too detailed for realistic scheduling problems. The stochasticity that is inevitably part of sorting operations is so high that the detailed solution of the ADBLA is counterproductive.

The priority extension does improve results, but the differences are often not operationally significant. Because implementing this approach would require dispatchers to continuously determine the priority of containers and shifting the most urgent ones forward, the cost-benefit ratio of the extension is unfavourable. The delayability extension, however, provides impressive improvements, in combination with load balancing algorithms as well as FCFS. Besides, airports can relatively easy implement this extension, which makes it a very interesting approach indeed.

We therefore conclude that load balancing algorithms in general can significantly improve performance of high-capacity sorter systems, especially when the contents of containers are more diverse. Approaches that model the sorter system in great detail, such as the ADBLA, do not improve the performance of the original DLBA, and are therefore of little use. Furthermore, the simulation studies show that delaying containers that are non-urgent reduces the workload of the sorter system, thereby reducing the missort rates substantially. This delayability extension is therefore an interesting approach that deserves more attention, although it can easily be applied 'as is' in the baggage handling industry.

Further Research

'Research leads to research', and we therefore conclude this thesis by mentioning a few subjects that could benefit from further research.

First, this research shows that a more detailed approach does not necessarily lead to better performance. Also, the literature review showed that there are very few other approaches that focus on these types of problems. It would therefore be interesting to start a more explorative study that determines how the DLBA could be altered in order to improve the performance. Interesting would be, for instance, not to use the actual contents of containers, but schedule them using a kind of classification system. Other subjects that should receive more attention, are, amongst others, the way the DLBA handles the EBS and how the DLBA copes with multiple sorters.

Second, this thesis already indicated that the DLBA has some restrictive assumptions that limit the application of the approach in reality. Because this study has been an initial 'proof of concept' little effort has been put into relaxing them. Both industries could, however, benefit from research that focusses on the relaxation of these assumptions. Especially the premise that containers are docked as soon as the dispatcher decides to unload them, is unrealistic and could influence the assignment decision as well as the performance.

Third, the delayability extension that has been developed in this research is only one of the possible variants that could be applied. Because this approach appeared to perform quite well, it would be useful to do further research into this extension. Interesting would, for instance, be to determine whether other variants could further improve the results, or achieve the same results with less hindrance for the operations at the airport. Furthermore, because extensions proved to be much more effective than workload balancing approaches, it is advisable to put effort in the development of extensions rather than developing new workload balancing approaches.

Fourth, this research has tested the different scheduling approaches and extensions using simplified simulation models. Reality, however, is complex and stubborn. It is therefore recommended that the approaches developed in this thesis, or in future research for that matter, are properly tested using realistic problem data and complex, realistic simulation or emulation models. Although this requires more simulation time, it provides significantly better, i.e. more reliable, results. Given that these models are developed anyway, for they are necessary to determine the expected performance, the additional tests require little effort, only time.

Finally, one of the objectives of this research was to develop a single scheduling approach for both express parcel companies and the baggage handling industry. The DLBA demonstrates that such an approach is possible, for it manages to increase throughput and decrease missort rates. The development of the two extensions for the baggage handling industry, however, shows that it is easier to develop tools for a single trade. In addition, the advantages of these extensions are potentially much larger than those of 'one size fits all' scheduling approaches. It is therefore recommended that future research does not fixate itself on finding one solution for both industries, but rather aims at finding a good approach for one of the trades.

References

Abdelghany, A., Abdelghany, K., & Narasimhan, R. (2006). Scheduling baggage-handling facilities in congested airports. *Journal of Air Transport Management*, *12*(2), 76–80.

Cheng, Y. (1997). A knowledge-based airport gate assignment system integrated with mathematical programming. *Computers & Industrial Engineering*, *32*(4), 837–852.

Cohen, Y. & Keren, B. (2009). Trailer to door assignment in a synchronous cross-dock operation. *International Journal of Logistics Systems and Management*, *5*(5), 574–590.

Frey, M., Artigues, C., Kolisch, R., & Lopez, P. (2010). Scheduling and planning the outbound baggage process at international airports. In *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management.* Macao, China.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*, *33*(4), 419–428.

Hallenborg, K. (2007). Decentralized scheduling of baggage handling using multi-agent technologies. In E. Levner (Ed.), *Multiprocessor scheduling.* Vienna, Austria: I-Tech Education and Publishing.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–97.

Law, A. M. & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). Singapore: McGraw-Hill.

Li, Z. P., Low, Y. H., Shakeri, M., & Lim, Y. G. (2009). Crossdocking planning and scheduling: Problems and algorithms. *SIMTech technical reports*, *10*(3), 159–167.

Mack, C. (1969). The Bradford method for the assignment problem. *New Journal of Statistics and Operational Research*, *1*, 17–29.

McAree, P., Bodin, L., & Ball, M. (2002). Models for the design and analysis of a large package sort facility. *Networks*, *39*(2), 107–120.

McAree, P., Bodin, L., Ball, M., & Segars, J. (2006). Design of the federal express large package sort facility. *Annals of Operations Research*, *144*(1), 133–152.

McInerney, F. & White, S. (1995). *The Total Quality Corporation: how 10 major companies turned quality and environmental challenges to competitive advantage in the 1990s* (1st ed.). New York, NY, USA: Truman Talley Books/Dutton.

McWilliams, D. L. (2005). Simulation-based scheduling for parcel consolidation terminals: a comparison of iterative improvement and simulated annealing. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, & J. A. Joines (Eds.), *Proceedings of the 2005 Winter Simulation Conference.* Orlando, FL, USA.

McWilliams, D. L. (2009a). Genetic-based scheduling to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, *56*(4), 1607–1616.

McWilliams, D. L. (2009b). A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers & Industrial Engineering*, *57*(3), 958–962.

McWilliams, D. L. (2010). Iterative improvement to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, *59*(1), 136–144.

McWilliams, D. L., Stanfield, P. M., & Geiger, C. D. (2005). The parcel hub scheduling problem: A simulation-based solution approach. *Computers & Industrial Engineering*, *49*(3), 393–412.

McWilliams, D. L., Stanfield, P. M., & Geiger, C. D. (2008). Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers. *Computers & Industrial Engineering*, *54*(4), 709–720.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, *5*(1), 32–38.

Robusté, F. & Daganzo, C. F. (1992). Analysis of baggage sorting schemes for containerized aircraft. *Transportation Research part A: Policy and Practice*, *26*(1), 75–92.

SITA. (2010). *Baggage report 2010.*

Tošić, V. (1992). A review of airport passenger terminal operations analysis and modelling. *Transportation Research part A: Policy and Practice*, *26*(1), 3–26.

Vanderlande Industries. (2010). *Annual Report 2010.*

Werners, B. & Wülfing, T. (2010). Robust optimization of internal transports at a parcel sorting center operated by Deutsche Post World Net. *European Journal of Operational Research*, *201*(2), 419–426.

Yu, W. & Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, *184*(1), 377–396.

Appendices

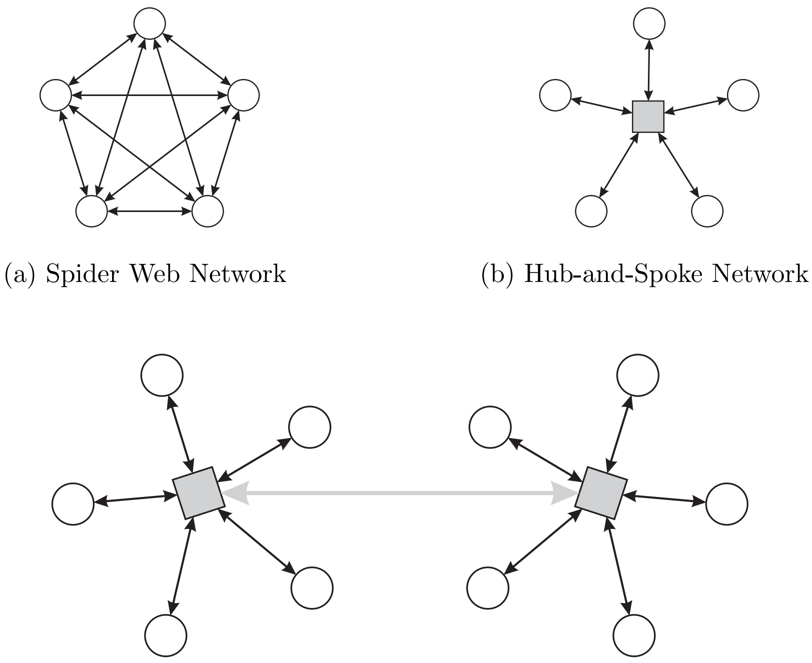
Appendix A

Network Layouts in the Parcel & Postal Industry

In the parcel and postal industry, three different network layouts are frequently used (although the third is a combination of the other two; see Figure A.1). The spider web network (Figure A.1a) is a network layout that is usually built gradually when a delivery company decides to open new depots one by one. Besides, it is a very interesting layout if, and probably only if, the transported volume from a depot to all the other depots is large enough to fill an entire truck. This, however, is only the case for very large express parcel companies. The other solution is the hub-and-spoke network (Figure A.1b) in which all parcels are transported to a central hub where they are sorted. This approach is especially interesting if a depot does not 'generate' enough parcels for a full truckload to all the other depots. Furthermore it allows a company to have many smaller and thus less expensive depots whilst making maximum use of the economies of scale in transport and sorting at the hub. Finally we find the combination network layout where the hubs of multiple hub-and-spoke networks are connected through a spider web network (Figure A.1c). This network layout is only useful for integrators (large express delivery companies such as DHL, UPS, FedEx, etc.) since it requires many large sorting facilities (*hubs*) that are relatively far apart from each other (e.g. on another continent).

Unfortunately each of the different network layouts has its own parcel and postal process, as is shown in Figure A.2. There are however also a lot of similarities between the three processes, which we could use to our advantage. We will therefore discuss the higher level parcel and postal processes and pay special attention to the (dis)similarities.

Consumer clients and small business clients will generally drop their parcel at a post office or the service point of a express parcel company and we therefore start our overview at this location. For large business clients such



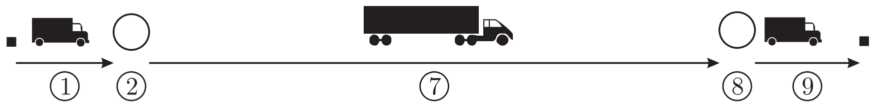
(a) Spider Web Network (b) Hub-and-Spoke Network

(c) Combination Network

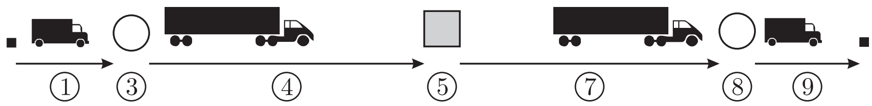
Figure A.1: Parcel & Postal Network Layouts

as on-line shops it is also possible that the express parcel company visits their warehouse to pick up the parcels. Note that in both cases the express parcel company, using a van or small truck, visits a location and picks up a set of parcels that are transported (1) to the nearest outbound depot. If the parcels are already labelled with the required barcodes, the driver will scan them, so that the contents of his van are known.

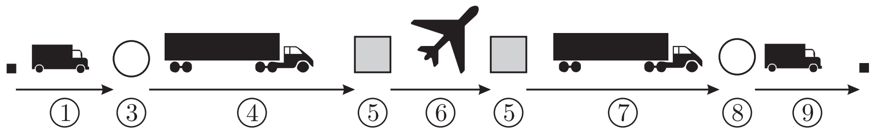
If the driver has scanned barcodes, they will be uploaded to the main server, whilst the van is unloaded at the outbound depot (2,3). If no suitable barcodes have yet been applied, the outbound depot is usually equipped with scanners that determine the destination and a machine that will label the parcels with a machine-readable barcode. If the company uses a spider web network the parcels will be sorted in the outbound depot (2). If the number of outfeeds of the sorter is limited, the parcels will be sorted based on destination depot, however when more outfeeds are available it would be possible to apply a finer sorting strategy. If there is, for instance, one destination depot with a very limited sorting capacity, the company could decide to 'presort' some of the parcels for this destination depot. When the company uses a hub-and-spoke or combination network there usual is no need to sort the parcels at the outbound depot (3).



(a) Spider Web Process



(b) Hub-and-Spoke Process



(c) Combination Process

Figure A.2: Parcel & Postal Processes per Network Layout

If we for now continue with the more complex hub-and-spoke and combination processes, we see that the unsorted parcels are transported (4) to a central hub. It is important to realise that at this point in time the contents of the trailer are not necessarily known. If the vans are unloaded and trailers are loaded randomly (i.e. without control) the contents of the trailers are not known to the company. If, however, each van is unloaded into a specific trailer the contents of these trucks can be known, which is information that could be used when assigning the trailers to the unload docks at the hub.

At the hubs (5) the parcels are sorted based on their final destination or the subsequent hub they should visit. For parcels that are shipped overseas usually aeroplanes are used to fly (6) the parcels from one hub to another. In that case the parcels are usually loaded in ULDs, which makes a finer sorting strategy possible. Given that the parcels have now been sorted, the contents of each ULD are known and this knowledge could be used when assigning the ULDs to the different infeeds.

Finally, the parcels are transported (7) towards the inbound depot, and because the parcels have all been sorted at least once (either at the outbound depot or at the hub), the contents of each trailer are known. At the inbound depot (8) the parcels are sorted for the last time, based on the delivery route that should deliver the parcel to the customer (9).

Appendix **B**

Algorithm Pseudocodes

Algorithm B.1 Arbitrary Scheduling

while there is at least one container in the queue **and**
there is at least one infeed location available **do**
  → select the next available infeed
  → randomly select a container from the queue
  → assign this container to the infeed location
end while

Algorithm B.2 First-Come-First-Served Scheduling

while there is at least one container in the queue **and**
there is at least one infeed location available **do**
  → select the next available infeed
  → select, from the queue, the container that arrived first
  → assign this container to the infeed location
end while

Algorithm B.3 Dynamic Load Balancing Algorithm Scheduling

```
while there is at least one infeed available do
    if there is exactly one container in the queue then
        → assign this container to the first available infeed
    else if there is more than one container in the queue then
        for all containers j in the queue J do
            for all destinations d in the set D do
                → add the items for destination d on the sorter system
                (a_d) and in container j (g_dj) together and store as c_dj
            end for
            → initialize the maximum number of items c_j to 0
            → determine which destination d assigned to an outfeed (x_d =
            1) from the set D has the highest number of items c_dj and
            assign this value to c_j
            → initialize the workload imbalance f_j to |L| · c_j
            for all destinations d in the set D assigned to an outfeed
            (x_d = 1) do
                → subtract the number of items for destination d c_dj from
                the workload imbalance f_j
            end for
        end for
        → determine which container j from the queue J has the lowest
        workload imbalance (f_j)
        → assign this container to the first available infeed
    end if
end while
```

Algorithm B.4 Advanced Dynamic Load Balancing Algorithm

Require: a set of available infeeds I **and** a set of available containers C

$\rightarrow$ determine TB_{horiz}
while $|C|$ is larger than one **and** $|I|$ is larger than zero **do**
  $\rightarrow$ select the next available infeed i from I
  for all containers c in the set C **do**
    $\rightarrow$ determine objective value EF_{ic}
  end for
  $\rightarrow$ determine which container c has the lowest excess outflow EF_{ic}
  $\rightarrow$ assign container c to infeed i
end while

if $|C|$ is equal to one **then**
  for all infeeds i in the set I **do**
    $\rightarrow$ determine the objective value EF_{ic}
  end for
  $\rightarrow$ determine which infeed i has the lowest excess outflow EF_{ic}
  $\rightarrow$ assign container c to infeed i
end if

Ensure: $C = \emptyset$ **or** $I = \emptyset$

Algorithm B.5 ADLBA: Determine Objective Value

Require: a selected container c **and** a selected infeed i
Require: a set of time buckets that denote the optimisation horizon TB_{horiz}

$\rightarrow$ copy the values of $FLOW$ to a temporary variable $FLOW_{ic}$
for all destinations d in the set D **do**
  $\rightarrow$ determine to which outfeed o destination d is assigned
  $\rightarrow$ determine TB_{start}, TB_{end}, and TB_{flow}
  for all time buckets TB in the set TB_{horiz} **do**
    $\rightarrow$ increase $FLOW_{ic}(TB, o)$ by the value of TB_{flow}
  end for
end for

for all outfeeds o in the set O **do**
  for all timebuckets TB in the set TB_{horiz} **do**
    if the value of $FLOW_{ic}(TB, o)$ is larger than the outfeed capacity $\frac{F_o \cdot tb}{3600}$ **then**
      $\rightarrow$ add the square of the difference between the expected outflow $FLOW_{ic}(TB, o)$ and the capcity $\frac{F_o \cdot tb}{3600}$ to the objective value EF_{ic}
    end if
  end for
end for

return the objective value EF_{ic} to the calling algorithm

Algorithm B.6 Priority Scheduling

Require: an available infeed i **and** a set of available containers C

→ copy the available containers C to $ContainerList$
→ initialise $u_{max} = 0$

for all destinations d **do**
    → determine u_d using t_j, U_{start}, and U_{end}
end for

for all containers c in $ContainerList$ **do**
    → initialise $u_c = 0$
    for all destinations d **do**
        → increase the urgency of the container u_c by the number of items for destination d multiplied by the urgency of destination d u_d
    end for
    if $u_c = 0$ **then**
        → remove container c from $ContainerList$
    else
        → update $u_{max} = \max(u_{max}, u_c)$
    end if
end for

if $ContainerList$ is empty **then**
    → copy the original list of containers C to $Containerlist$
else
    for all containers c in $ContainerList$ **do**
        if $u_c < 0.75 \cdot u_{max}$ **then**
            → remove container c from $ContainerList$
        end if
    end for
end if

return $ContainerList$ to the DLBA or DDLBA scheduling algorithm

Algorithm B.7 Delayable Scheduling

```
for all destinations d do
    if destination d in container then
        if outfeed for destination d not yet assigned then
            → send container to normal parking
            return to scheduling approach
        else
            → update release time of container to minimum of current
            release time and start-up time of outfeed + 1 hour
        end if
    end if
    → send container to long parking
    → wait until release time
    return to scheduling approach
end for
```

Appendix C

Optimality of FCFS

Although the first-come-first-served (FCFS) approach can be applied to many different scheduling problems, there is one type of problem for which it provides optimal schedules. If one always selects the container that has been waiting the longest, it seems as if one is minimising the maximum waiting time. This appendix shows that, if one assumes that the processing times of all containers are identical, FCFS is indeed a heuristic that results in an optimal solution for the minimax waiting time problem.

For instance, assume that two containers are already waiting for some time and container one arrived first, i.e. $t_1 > t_2$ where t_1 and t_2 denote the waiting times of containers 1 and 2 respectively. The processing times are identical and thus one may assume that $p_1 = p_2 = 1$. Not applying FCFS would result in container 2 waiting for t_2 as it is scheduled immediately and container 1 waiting for $t_1 + 1$ as it is scheduled second. As $t_1 > t_2$, it also holds that $t_1 + 1 > t_2$ and thus the maximum waiting time is $t_1 + 1$. If FCFS would be applied container 1 would wait for t_1 as it is scheduled immediately and container 2 would wait for $t_2 + 1$. In this case it is not clear which waiting time is longer, so we consider both possibilities: possibility one with a waiting time of t_1 and possibility two with a waiting time of $t_2 + 1$. If FCFS would not provide the optimal solution, there should be instances where FCFS would result in larger waiting times than not applying FCFS, the latter waiting time being $t_1 + 1$. Obviously $t_1 + 1 > t_1$ so for possibility one FCFS does provide the best solution. For possibility two we find that $t_1 + 1 > t_2 + 1$ since $t_1 > t_2$ is given. Again, FCFS provides the best solution, which proves that FCFS would be an optimal heuristic for this problem.

If, however, the assumption of identical processing times is abandoned, FCFS is not by definition the best approach. Using the same example as before we now find that not applying FCFS results in a waiting time of $\max\{t_1 + p_2, t_2\} = t_1 + p_2$ because $t_1 > t_2$. Applying FCFS would result in a waiting time of $\max\{t_1, t_2 + p_1\}$, which could be either t_1 or $t_2 + p_1$. In

U	set of unload docks, $(u \in U)$
T	set of time buckets, $(t \in T)$
C	set of inbound containers, $(c \in C)$
r_c	arrival time of container c (release time)
x_{ct}	1 if container c is assigned to time bucket t, 0 otherwise
W_{max}	maximum waiting time over all containers C

Table C.1: Notation for the FPD of FCFS

the former case FCFS is obviously better as $t_1 < t_1 + p_2$. In the latter case, FCFS is only better if $t_2 + p_1 < t_1 + p_2$, which is not necessarily the case. For this problem FCFS therefore does not guarantee to give the optimal solution.

Because FCFS proved to be an optimal heuristic for the first problem described above, minimising the maximum waiting time whilst assuming identical processing times, a formal problem description (FPD) of this specific problem could contribute to the understanding of the approach. Using the notation from Table C.1, an integer linear program (ILP) can be defined:

$$\text{minimise} \quad W_{max} \tag{C.1}$$

$$\text{subject to} \quad \sum_{c \in C} x_{ct} \leq |U| \qquad \forall\, t \in T \tag{C.2}$$

$$\sum_{t \in T} x_{ct} = 1 \qquad \forall\, c \in C \tag{C.3}$$

$$x_{ct} \cdot t - r_c \leq W_{max} \qquad \forall\, c \in C,\ \forall\, t \in T \tag{C.4}$$

$$x_{ct} \in \mathbb{B} \qquad \forall\, c \in C,\ \forall\, t \in T \tag{C.5}$$

Goal function C.1 minimises the maximum waiting time (W_{max}) over all containers C, FCFS could therefore be considered a waiting time balancing approach. Constraint C.2 ensures that at any point in time there are no more containers being unloaded than there are unloading docks and constraint C.3 forces the integer linear program to schedule each container completely, although this formulation alone would allow a linear program (LP) to pre-empt the unloading process. This pre-emption is prevented by constraint C.5 that states that the values of x_{ct} should be binary, which is what makes this an ILP, and results in each container being unloaded in a single time bucket. Finally, constraint C.4 ensures that the value W_{max} does indeed represent the maximum waiting time. $x_{ct} \cdot t$ evaluates to 0 if the container is not unloaded in this time period and results in a negative waiting time. If the container is unloaded in time bucket $x_{ct} \cdot t$ evaluates to t and the waiting time is equal to the difference between the time the container is unloaded (t) and the time the container arrived (r_c). As a result the maximum value of constraint C.4 for each of the containers is always positive and does indeed represent the waiting time incurred by this container.

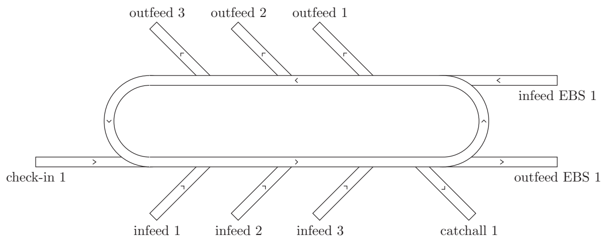
Appendix **D**

Simulation Models

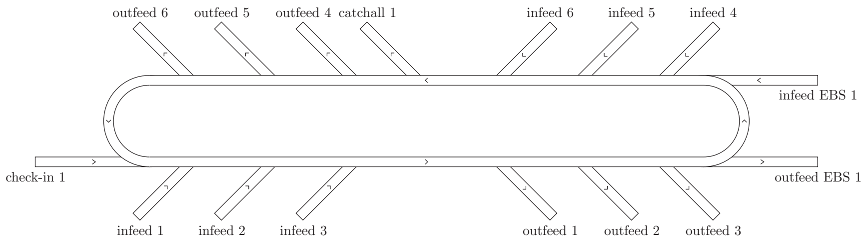
The three simulation models used in this research are selected to cover both simple and complex sorter systems. Model 110 is a relatively simple sorter system with only a limited number of infeeds and outfeeds. Model 120 is a bit more difficult, for the internal travel times are larger and there are more infeeds and outfeeds. Finally, Model 22c is the most complex of the three, with as many infeeds and outfeeds as model 120 but with crossovers that can severly hamper, but also improve, operations. This appendix explains the reasoning behind these simulation models, and justifies the selected transport capacities.

Model 110 (Figure D.1a) is probably one of the most simple sorter systems that could be constructed when studying scheduling of inbound operations. It consists of one main sorter system with a capacity of 3600 items per hour (iph), three infeed conveyors with a capacity of 1200 iph each, and three outfeed conveyors, also with a capacity of 1200 iph each. A blocking time of 0.6 seconds, based on the outfeed conveyor capacity and the space required for one item, is enough to prevent collisions. The items are placed on, and removed from, the conveyors by workers at a rate of 400 iph, which is a figure commonly used in the baggage handling industry. The time required to change the containers is not modelled explicitly and is assumed to be included in the time that is required to position/remove an item. The outfeed and infeed conveyor that connect the main sorter to the EBS have the same capacity as the other infeeds and outfeeds, and the crane inside the EBS is able to retrieve items at a rate of 400 iph. Finally the check-in baggage arrives on a dedicated conveyor that also has a capacity of 1200 iph.

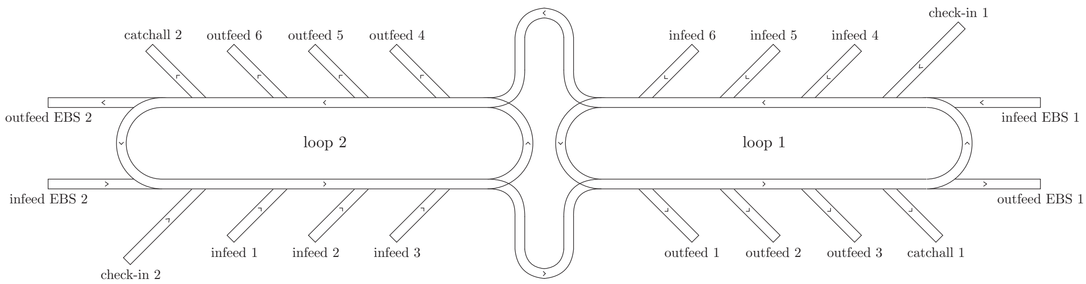
Model 120 (Figure D.1b) is, from a scheduling perspective, a bit more complicated than model 110, because there are more choices when assigning a container to an infeed. Model 120 consists of six infeed conveyors and six outfeed conveyors all with a capacity of 1200 iph, again workers place items on, or remove them from, the conveyors at a rate of 400 iph. Because the



(a) Sorter Layout 110



(b) Sorter Layout 120



(c) Sorter Layout 22*c*

Figure D.1: Layouts of Three Test Models

total infeed and outfeed capacity is doubled, compared to model 110, also the capacity of the main sorter and the check-in conveyor is doubled to 7200 iph and 2400 iph respectively. The capacity of the infeed and outfeed conveyor of the EBS is not changed and neither is the retrieval rate of the EBS crane. Model 120 is especially interesting for this study, because of the alternating infeed and outfeed areas. It might, for instance, be an interesting approach to assign containers to infeed 1, 2, or 3 when the majority of the items is supposed to go to outfeed 1, 2, or 3. Analogously one would assign a container to infeed 4, 5, or 6 when the majority of the items is destined for outfeed 4, 5, or 6. However, this might cause items to recirculate because the outfeeds are blocked by the preceding items. A better approach may therefore be to assign the containers in such a way that the workload for each of the outfeeds from infeed area one is equal to the workload from infeed area 2. It is clear that finding a good scheduling approach for this model, although being a simplified version of systems that are implemented in reality, might not be so obvious as it appears on first glance.

Model 22*c* (Figure D.1c) is designed to combine the challenges from both model 110 and 120. Its physical layout is based on two model 110 sorters in loop configuration, which are connected by crossovers. These crossovers have a reduced capacity of 1200 iph and their use should therefore be limited in order to prevent the main sorters from cluttering. At first glance, model 22*c* could be considered a simple combination of two separate model 110 problems. However, it is likely that items that are destined for outfeeds on main sorter 2 arrive at infeeds on main sorter 1. Containers hold items for all destinations and even the check-in desks may occasionally accept items that should be redirected to the other main sorter. The simple solution is, similar to the solution proposed for model 120, to unload a container at the main sorter that is the destination of the majority of the items inside. However, in reality one might decide to unload the container at the 'wrong' main sorter, for example when all infeeds at the 'correct' sorter are occupied and a flight is bound to close within a few minutes. Unloading a container at the 'wrong' main sorter might enable the baggage to still make it to the flight. Also in the parcel and postal industry motives exist to unload containers at the 'wrong' main sorter. Because companies in the express parcel business focus on throughput, they want to keep both main sorters filled up completely. This ensures that items are sorted at every opportunity, as soon as an outfeed conveyor becomes available.

Appendix **E**

Model Scenarios

The model scenarios used in this research are not exhaustive, many other variants and alternatives could be constructed. They do, however, mark the borders wherein realistic scenarios can be found. This appendix does not only mention the selected parameter values, it also aims at justifying these values by discussing the reasoning behind them. Finally, this appendix also includes the explanation of the selected approach to generate check-in baggage items.

Number of Destinations The number of destinations depends on whether a parcel and postal or baggage handling scenario is used. For parcel and postal the number of destinations is equal to the number of outfeeds and each outfeed is assigned to one destination throughout the entire sort shift. For baggage handling the number of destinations is twice the number of outfeeds. At the start of the simulation the first outfeed in every outfeed area (i.e. outfeed 1 and, if present, 4) is assigned to the destination with the same index. After fifteen minutes the second outfeed in every outfeed area is assigned to the destination with the same index and after another fifteen minutes the third outfeed in every outfeed area is assigned to the destination with the same index. Each destination is assigned to an outfeed for three hours and then left unassigned for half an hour, to ensure that all items that were present, waiting ULDs for instance, can be removed. So after three hours of simulation time the first outfeeds in every outfeed area become available and after three-and-a-half hour they are assigned to the destination with the same index plus the number of outfeeds (i.e. in model 110 destination 4 is assigned to outfeed 1 and in model 120 destination 7 and 10 are assigned to outfeeds 1 and 4 respectively). Again the outfeed closes after three hours of simulation time.

Number of Containers The number of containers is assumed to be fixed for all problem instances of one scenario - model combination, otherwise the problem size (number of items that have to be sorted) would differ too much between the problem instances. The interarrival times of the containers and the number of parcels inside one container are uniformly distributed between a lower- and upperbound, which are set for each scenario - model combination. The values of these parameters are set in such a way that combined they result in a workload that is approximately 65% of the total outfeed capacity for parcel and postal scenarios, assuming that the total sort should take around four hours. Furthermore, all containers are scheduled to arrive in two hours, in order to ensure that the scheduling algorithm has indeed containers to choose from. For baggage handling scenarios the combination should result in a workload that is around 25% of the total outfeed capacity, assuming that each outfeed is open for six hours.[1] Because these three parameters determine the workload for the sorter system, they are required for any scenario.

In baggage handling often multiple containers arrive at the same time, because they are unloaded from the same plane. The number of containers that arrive simultaneously is uniformly distributed between the lower- and upperbound of the batch size parameter. When a lower- and upperbound different from one are used, the interarrival time parameter describes the interarrival times between two batches of containers.

Container Contents For all containers that belong to the same container type, the distribution of destinations over the items inside is the same. Unfortunately only the PP-even scenario can suffice with one container type, for all other scenarios more are required. Letting n denote the number of outfeeds, for PP-even each destination has a probability of occurrence of $\frac{1}{n}$. For PP-uneven $1 + n$ container types are used, one with an even distribution and n with a preferred destination. In the container type with even distribution, each destination has a probability of occurrence of $\frac{1}{n}$. In the container types with a preferred destination the preferred destination has a probability of occurrence of 0.50, for all the other destinations this probability is $\frac{0.50}{n-1}$. The probability that a specific container type is selected is $\frac{1}{1+n}$, on average each container type is thus selected an equal number of times.

For baggage handling the distribution of destinations inside one container is more complicated. If a distribution similar to the PP-uneven distribution would be used, many baggage items would arrive after their flight has already left. In order to reduce the missort rate caused by this modelling error, time windows are introduced. As a result, the BHS-even scenario consists of two container types. Letting n denote the number of outfeeds again, the first container type holds mainly baggage items for the first n destinations. That is around 85% of the contents of the containers is for the first n destinations, each of these destinations thus having a probability of occurrence of $\frac{0.85}{n}$. The remaining 15% is for the last n destinations with a probabilty of occurrence of $\frac{0.15}{n}$ per destination. Because this container type primarily contains items for which the outfeed closes between three and three-and-a-half hours, we assume

[1]The remaining 40% is used for the check-in baggage which details are explained later.

that it can only arrive during the first three hours. The second container type consists solely of items for the last n destinations, each with a probability of occurrence of $\frac{1}{n}$. In order to ensure a continuous arrival of containers, the time window of this container type is contiguous to the time window of the previous container type, i.e. it can arrive between three and seven hours simulation time. As these two container types cannot be requested simultaneously, no probability of occurrence of these container types is necessary.

The BHS-uneven scenario consists of $2+2n$ container types, where the first two are equal to the ones mentioned for the BHS-even scenario. The other $2n$ container types prefer one destination and 90% of their contents is for this preferred destination. Each of these preferred container types has a time window at which it can arrive that starts as soon as the outfeed is assigned to this destination and ends 30 minutes before the outfeed closes. For the first n preferred containers, which have a preferred destination between 1 and n, the remaining 10% is distributed evenly over all other destinations, each of them having a probability of occurrence of $\frac{0.10}{2n-1}$. For the latter n preferred containers, with preferred destinations between $n+1$ and $2n$, the remaining 10% is evenly distributed over the other last n destinations, each of them having a probability of occurrence of $\frac{0.10}{n-1}$. It is important to make this distinction, as we did for the BHS-even scenario, because the first n container types arrive early, whereas the latter n container types arrive after the flights of the first n destinations have left. Furthermore, in the BHS-uneven scenario two up to seven different container types can arrive simultaneously, which makes it necessary to determine the probability of occurrence of a specific container type. The probability of occurrence of a preferred container type is three times the probability of the evenly distributed container types. For example, when one evenly distributed container type and two preferred container types can be selected, the probability for the evenly distribute type is $\frac{1}{1+3+3} \approx 14\%$ and for the two preferred container types this is $\frac{3}{1+3+3} \approx 43\%$.

Check-In Baggage Items The flow of check-in items, which is only present in the baggage handling scenarios, should also be discussed. In order to model the arrival process of these baggage items as realistic as possible, the flow of check-in items is modelled independently for each destination, using a Poisson distribution. This distribution is famous as it models the unknown arrival process of independent and identically distributed (IDD) non negative random variables (Law & Kelton, 2000). Also very attractive is the fact that the combination of two Poisson distributions also results in a Poisson distribution: if $X_1 \sim Poisson(\lambda_1)$ and $X_2 \sim Poisson(\lambda_2)$ then $X_1+X_2 \sim Poisson(\lambda_1+\lambda_2)$. A practical example: assume two destinations for which it is known that $\lambda_1 = 100$ and $\lambda_2 = 50$, that is, we expect 100 baggage items for destination 1 and 50 for destination 2 in the next hour. According to the Poisson process, a baggage item for destination 1 is expected to arrive in $\frac{1}{100} \cdot 3600 = 36$ seconds, but a baggage item for destination 2 can be expected in $\frac{1}{50} \cdot 3600 = 72$ seconds. We are, however, interested in when the next baggage item arrives irrespective of the destination of this baggage item. Again using the Poisson process, this baggage item arrives in approximately $\frac{1}{\lambda_1+\lambda_2} = \frac{1}{100+50} \cdot 3600 = 24$ seconds and has destination 1 with probability $\frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{100}{50+100} = \frac{2}{3}$ and destination 2 with probability $\frac{50}{50+100} = \frac{1}{3}$.

The flow of check-in items is generated in such a way, that for each destination 40% of the total outflow capacity (rate at which items are removed from the outfeed conveyor multiplied by the time an outfeed is assigned to a destination) originates from the check-in stream. For model 120 a proportion of 35% is used to prevent unrealistic missorts rates (around 20%) from occuring. In order to ensure that all baggage is able to arrive before the outfeed closes, the check-in desks are modelled to close 30 minutes before the outfeed closes. Modelling early check-ins (check-ins before the outfeed is assigned to the destination) is somewhat difficult, simple distributions either result in large flows of check-in baggage during the first minutes of the simulation or result in an unequal distribution of baggage items over the different destinations. The number of bags that are early is therefore based on the time until the outfeed opens for a destination and the time an outfeed is assigned to a destination. Using t to denote the time at which an outfeed is assigned to a specific destination in minutes, the fraction of baggage items that arrive before this time can be determined using:

$$p_{early} = \frac{0.10 \cdot t}{0.10 \cdot t + 0.90 \cdot 150} \tag{E.1}$$

If, for example, an outfeed is assigned directly to a destination, $t = 0$ and, as may be expected, 0% of the baggage items arrive early. If, however, an outfeed is assigned to a destination at $t = 210$, about 13% of the baggage items will arrive in the first two-and-a-half hours and the remaining 87% in the next two-and-a-half hours, when the outfeed is assigned to the destination of these baggage items.

Appendix **F**

Required Number of Replications

The relative error approach, described by Law and Kelton (2000), is used to determine the number of replications that is required to achieve statistically sound results. This approach requires the expected average and expected variation that could be determined using an initial run with only a limited number of replications. Therefore each simulation model - scenario combination is simulated 20 times, and for each of the performance indicators the required statistics are collected. According to Law and Kelton, the minimum number of replications that is needed for a specified precision can be obtained using:

$$n^\star = \left(\frac{t_{i-1,1-\alpha/2}}{\gamma' \cdot \overline{X}}\right)^2 \cdot \mathrm{VAR}(X) \tag{F.1}$$

Where $n^\star$ is the minimum number of replications, $t_{i-1,1-\alpha/2}$ is the value from the t distribution with $i-1$ degrees of freedom and a confidence interval of $1-\alpha$, i is the number of replications that was performed in order to determine $n^\star$, γ' is the adjusted relative error, $\overline{X}$ is the average value of the performance indicator, and $\mathrm{VAR}\,X$ is the variance of the performance indicator value.

We would like to determine the average value of the performance indicators with a maximum relative error (γ) of 5% with a confidence interval of 95%. Or, stated differently, in 95% of the cases the estimated average value $\overline{X}$ should not deviate more than 5% from the actual average value μ. The adjusted relative error γ' can be calculated relatively easy using $\frac{\gamma}{1+\gamma} = \frac{0.05}{1+0.05} = 0.048$ and $t_{20-1,1-0.05/2} = 2.093$. Table F.1 shows the value of $n^\star$ for each of the simulation model - scenario combinations. In order to be able to compare the results of individual problem instances, it is necessary to use the same number of replications for both baggage handling (and for both parcel and postal) scenarios within one simulation model. Generating over 2500 solutions, as would be required for the baggage handling scenarios of model 110, however,

	PP-even	PP-uneven	BHS-even	BHS-uneven
model 110	13	28	846	2591
model 120	10	23	265	419
model 22c	10	24	443	689

Table F.1: Required Minimum Number of Replications ($\gamma = 5\%, \alpha = 5\%$)

is unrealistic. Therefore an increased relative error γ of 10% is used for these simulations, thereby reducing the required number of replications to approximately 800. Summarised, for the baggage handling scenarios 1000 problem instances are generated for model 110, 500 for model 120, and 750 for model 22c. For the parcel and postal scenarios 50 replications are sufficient for all the simulations models. When assessing scheduling approaches it should also be checked whether the specified accuracy, a maximum relative error of 5% with 95% certainty, is achieved or if additional replications are required.

Appendix **G**

Statistical Significance

This appendix explains the pairwise two-tailed t-test, which is used to determine whether two scheduling approaches perform statistically different on a (key) performance indicator. This type of statistical test can only be used if two datasets have the same size and the individual results are comparable. That is, both should contain the same number of results and result 1 of dataset 1 should be based on the same problem instance as result 1 of dataset 2. Because the same number of replications is performed for each of the scheduling algorithms and each algorithm is applied to the same problem instances, this test can be applied to the problem at hand.

The difference between two scheduling approaches is called 'statistically significant', when the difference cannot be explained by a random error term ϵ. Why this is important and how this can be tested, is something that is best explained using an example. Assume that we have two sets of results X and Y, the throughput values for FCFS and ARB on model 110 - PP-even scenario, for instance. Each set consists of n values, as n simulation runs have been performed for both FCFS and ARB. The sample means $\overline{X}$ and $\overline{Y}$ and sample standard deviations s_X and s_Y can now easily be determined. Unless the approaches are identical or very few simulation runs have been used, it is likely that $\overline{X} \neq \overline{Y}$, i.e. there is a difference between the two means. However, only the sample means $\overline{X}$ and $\overline{Y}$ have been determined and not the true means μ_X and μ_Y, which can be very different indeed. In Section 7.1 (page 119) we stated that we aim at finding averages that do not deviate more than 5% (or 10%) from the true means. As a result, it is possible that even though $\overline{X} \neq \overline{Y}$, both sets of values have the same true mean ($\mu_X = \mu_Y$). Because the same number of simulation runs and the same problem instances are used for all the algorithms, a pairwise comparison can be made. That is, the focus is no longer on the performance of X and Y on problem instance 1, X_1 and Y_1, but on the difference between the two results, $Z_1 = X_1 - Y_1$. Or, more generally, $Z = X - Y$. If both X and Y are derived from the same distribution the sample mean of the difference $\overline{Z}$ should, in the long run, be equal to zero. This is a problem that is usually described as the pairwise two-tailed t-test:

$$t = \sqrt{n} \cdot \frac{\overline{x} - \mu}{s_x} = \sqrt{n} \cdot \frac{\overline{Z}}{s_Z} \tag{G.1}$$

The value of μ is left out of the equation, as the hypothesis is that there is no difference and thus $\mu = 0$. The resulting value t is, under the hypothesis that $\mu = 0$, distributed according to the Student's t distribution with $n - 1$ degrees of freedom. For the 95% confidence level ($\alpha = 5\%$) and $n = 20$, 50 or ∞ the values for t are $t_{0.975,19} = 2.009$, $t_{0.975,49} = 2.093$, and $t_{0.975,\infty} = 1.960$ respectively. If the value for t from equation G.1 is larger than $t_{1-0.5\alpha,n-1}$, X and Y are statistically not identical and X is larger than Y. Similarly, if the value for t is lower than $-t_{1-0.5\alpha,n-1}$, X and Y are statistically not identical and Y is larger than X. In these cases we can state that the differences between X and Y are statistically significant, i.e. we can show using a 95% confidence level that X and Y cannot be part of the same distribution.

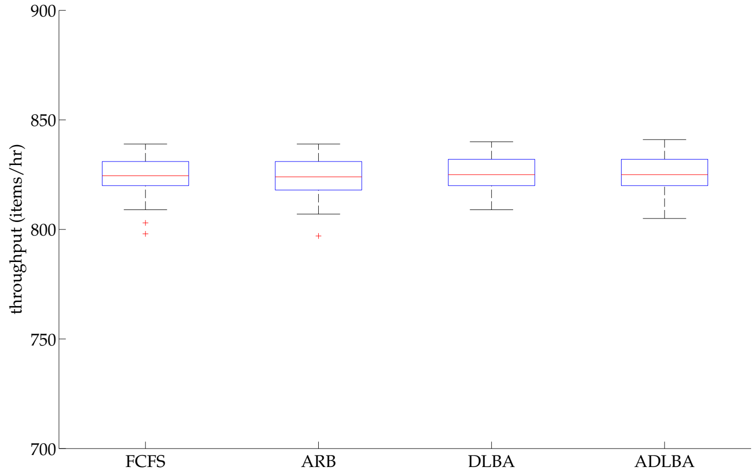
Appendix **H**

Simulation Results

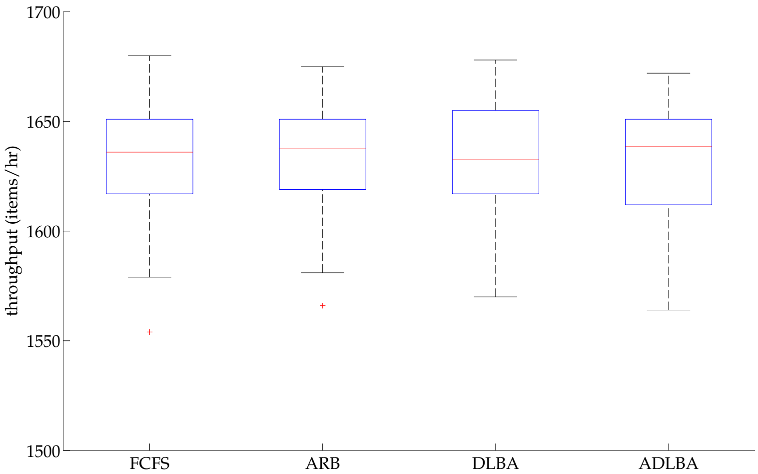
This appendix presents the more elaborate simulation results. It therefore not only provides the results for the key performance indicators throughput and missort rate, but also for the other performance indicators. For convenience, Table H.1 provides for each (key) performance indicator – scenario combination the page on which the results can be found.

performance indicator	PP-even	PP-uneven	BHS-even	BHS-uneven
Throughput	124	128	-	-
Missort rate	-	-	132	139
Maximum number of waiting containers	125	129	133	140
Average container waiting time	126	130	134	141
Maximum number of items in EBS 1	-	-	135	142
Maximum number of items in EBS 2	-	-	136	143
Total recirculation rate	127	131	137	144
Maximum number of containers in long parking	-	-	138	145

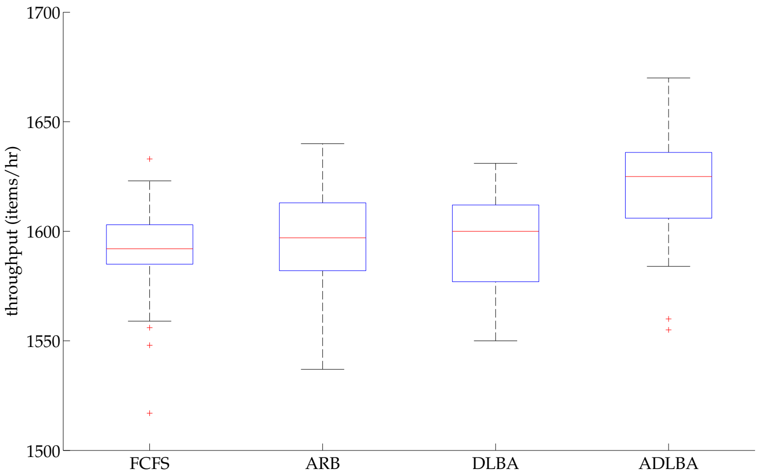
Table H.1: Page overview for simulation results



(a) model 110

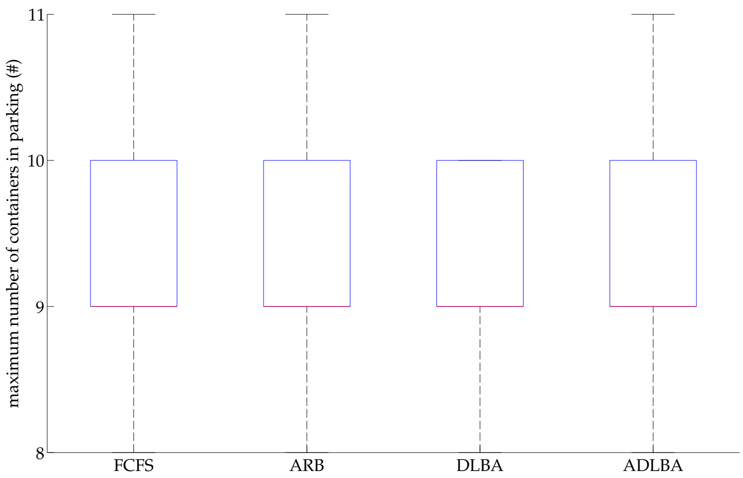


(b) model 120

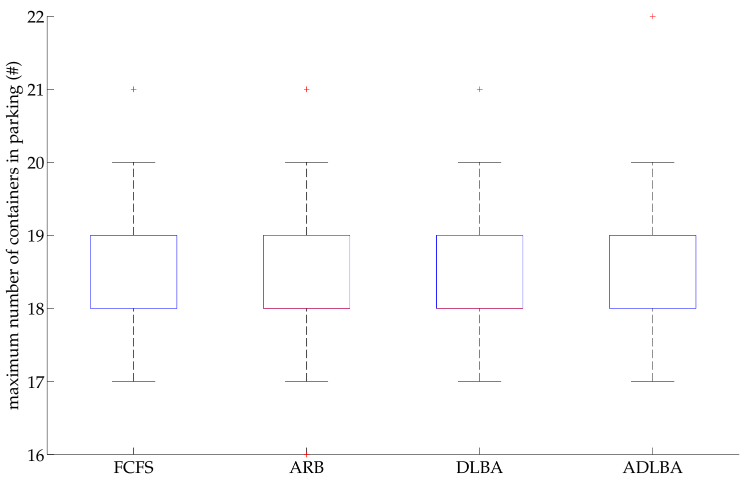


(c) model 22c

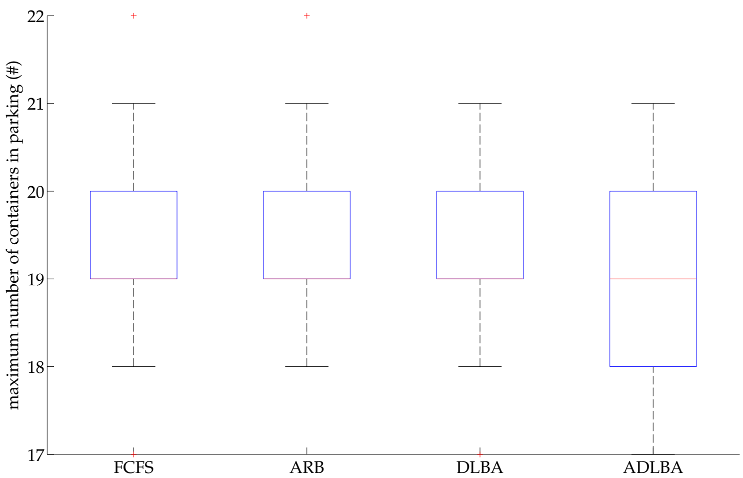
Figure H.1: PP-even — Throughput



(a) model 110

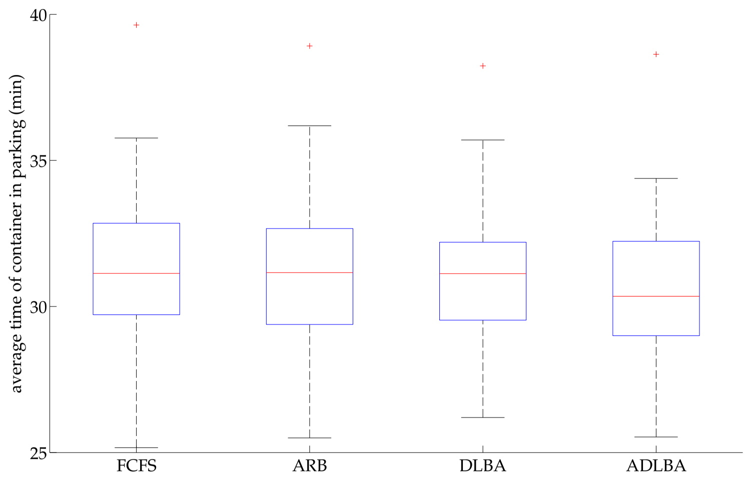


(b) model 120

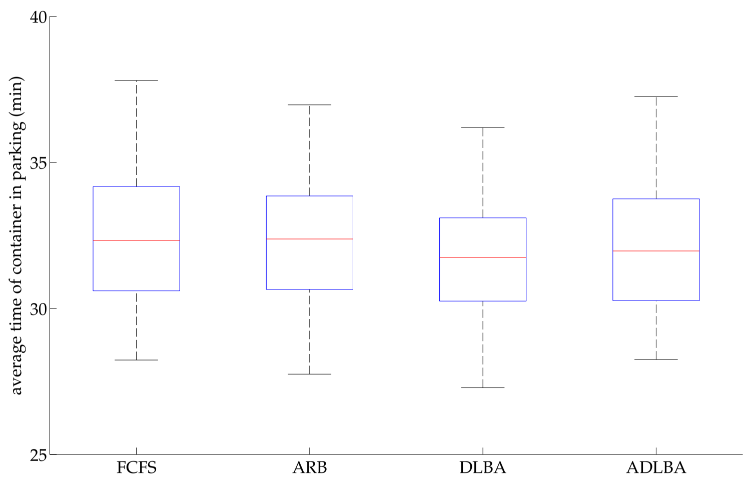


(c) model 22c

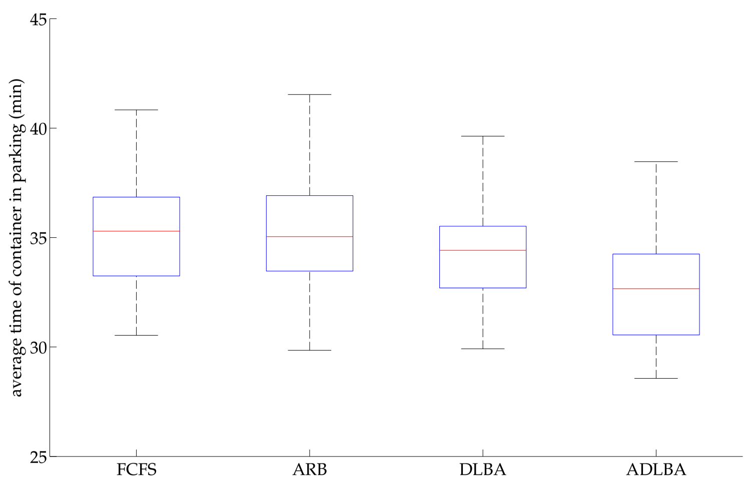
Figure H.2: PP-even — Maximum number of waiting containers



(a) model 110

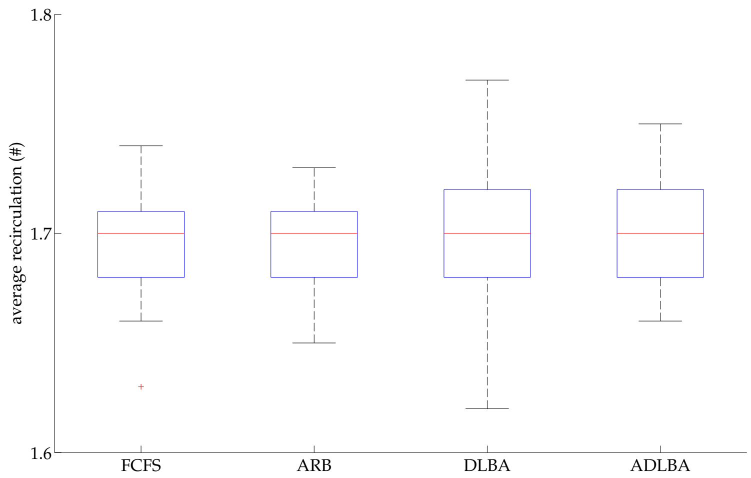


(b) model 120

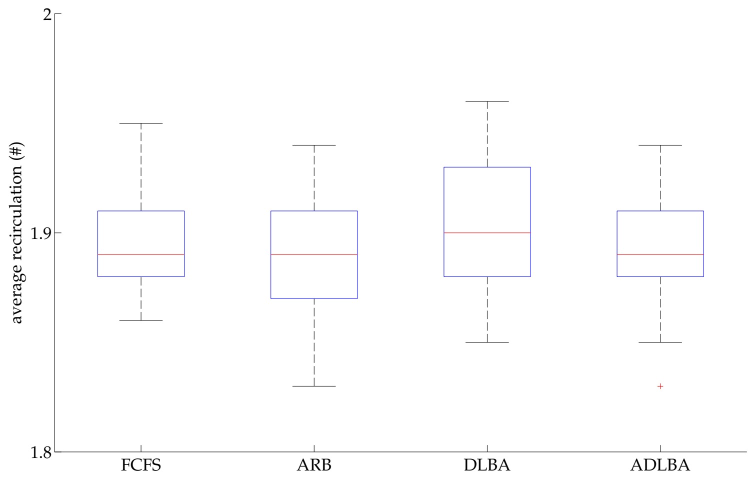


(c) model 22*c*

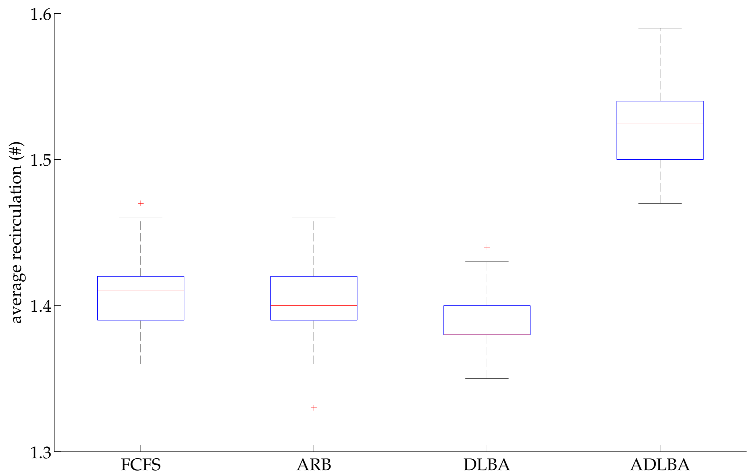
Figure H.3: PP-even — Average container waiting time



(a) model 110

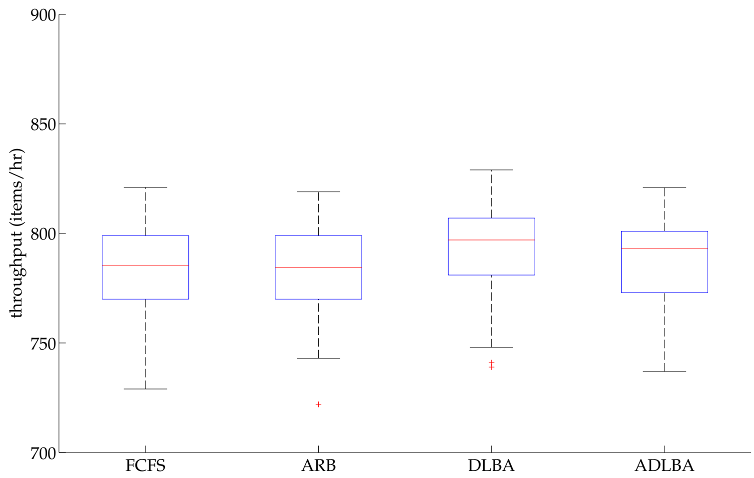


(b) model 120

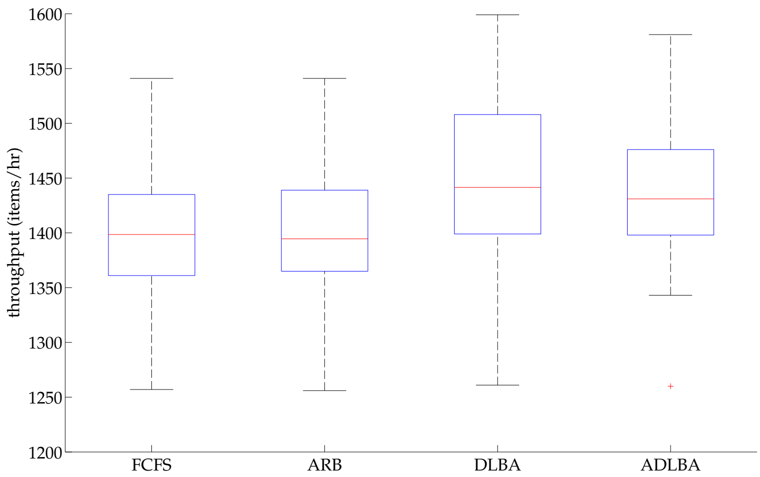


(c) model 22c

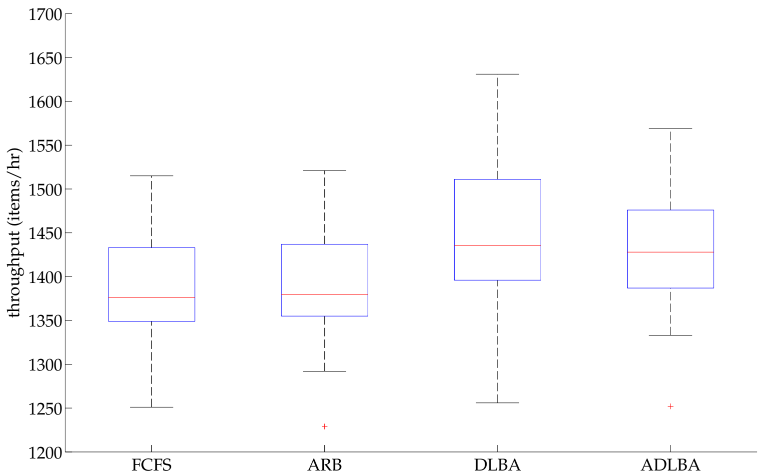
Figure H.4: PP-even — Total recirculation rate



(a) model 110

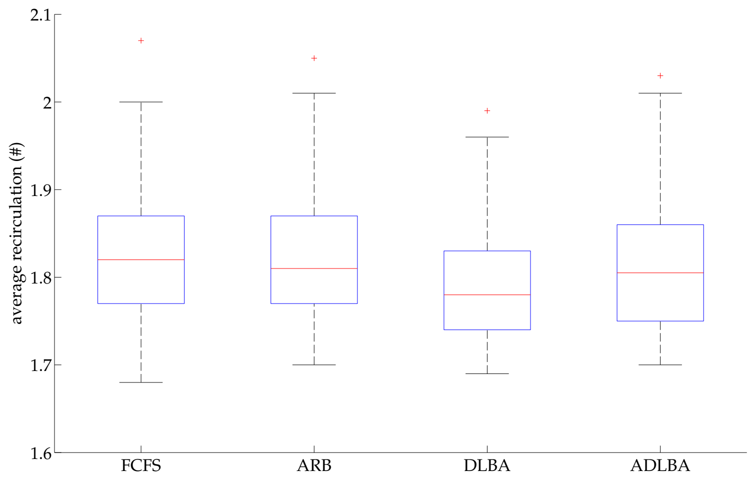


(b) model 120

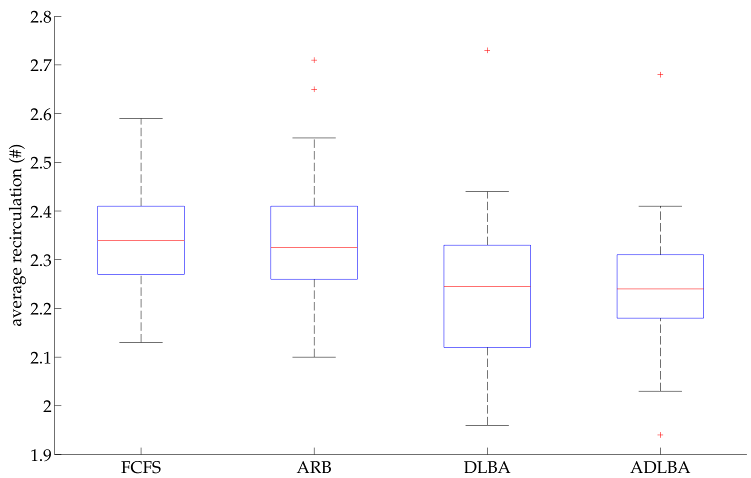


(c) model 22c

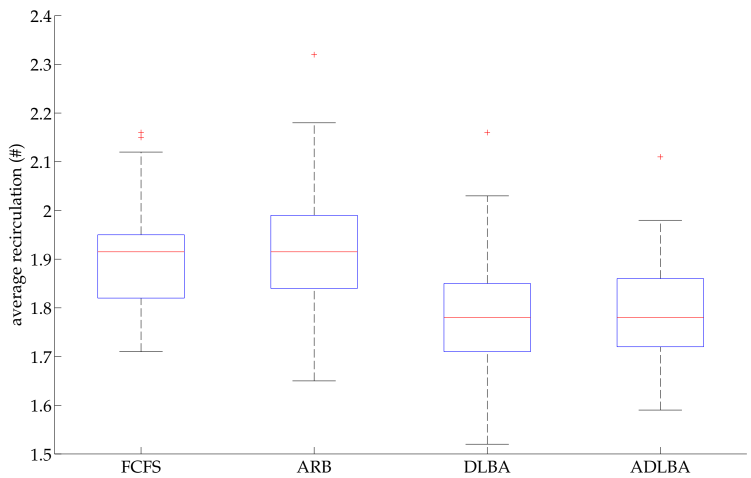
Figure H.5: PP-uneven — Throughput



(a) model 110

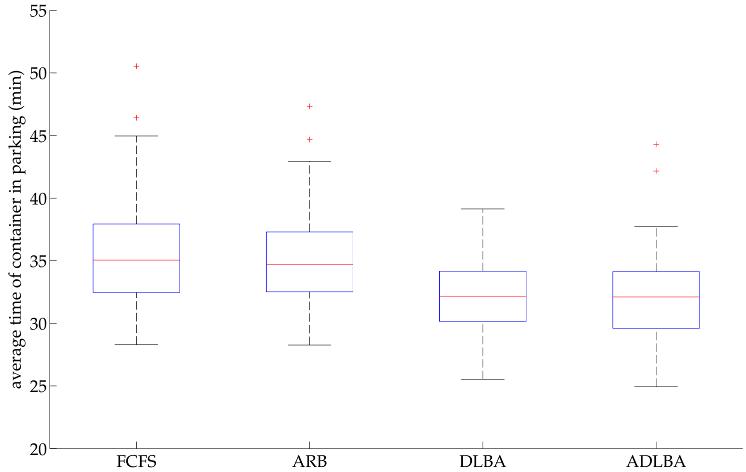


(b) model 120

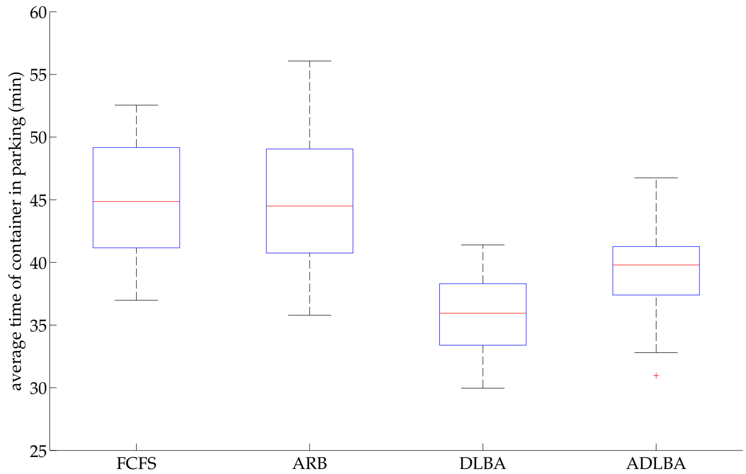


(c) model 22*c*

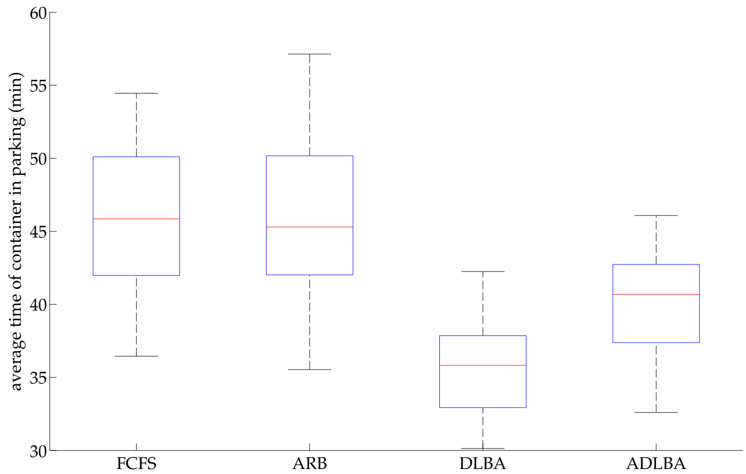
Figure H.6: PP-uneven — Maximum number of waiting containers



(a) model 110

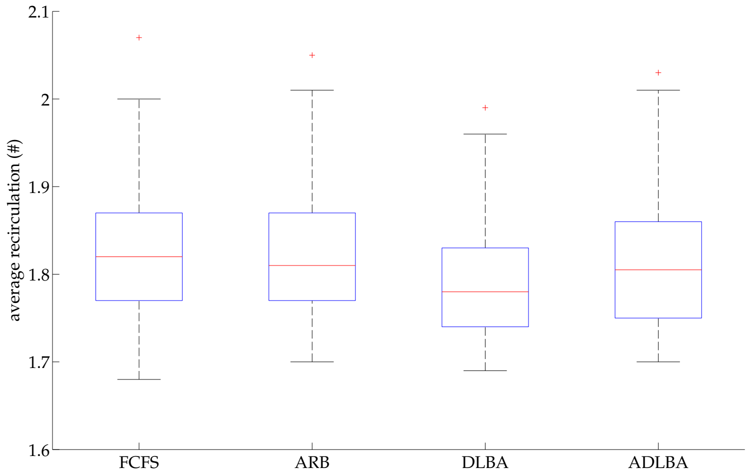


(b) model 120

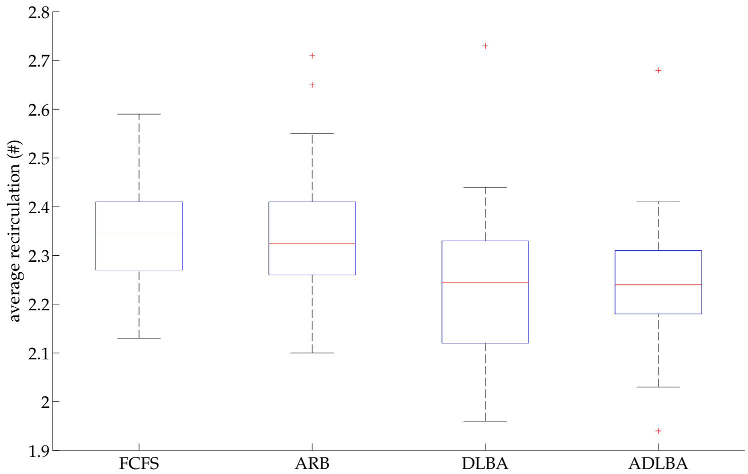


(c) model 22c

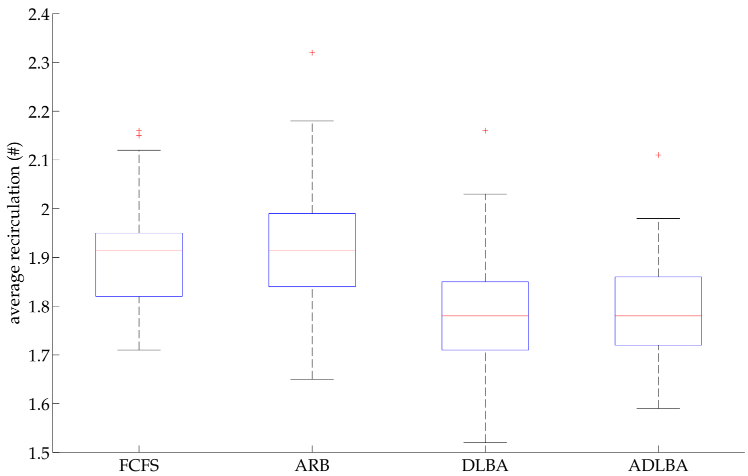
Figure H.7: PP-uneven — Average container waiting time



(a) model 110

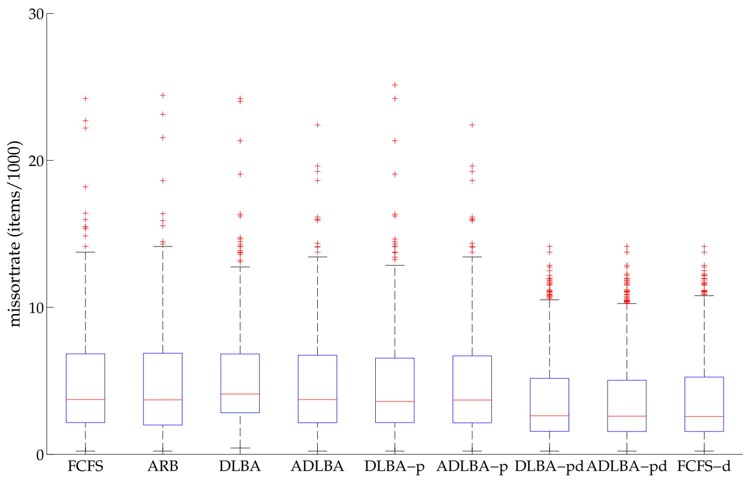


(b) model 120

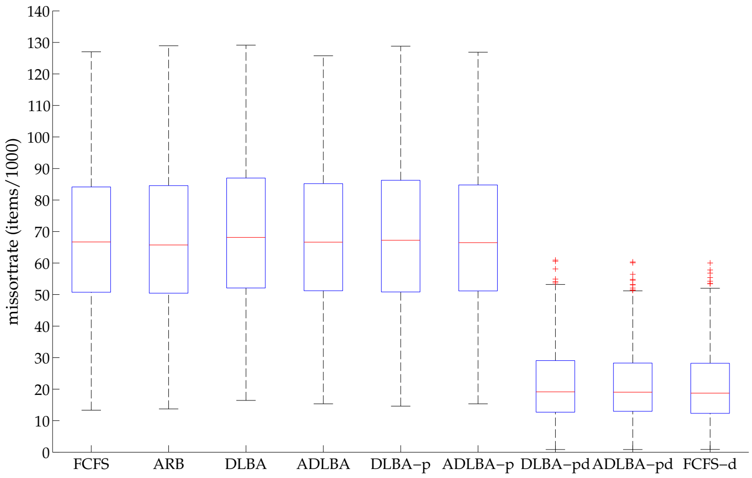


(c) model 22*c*

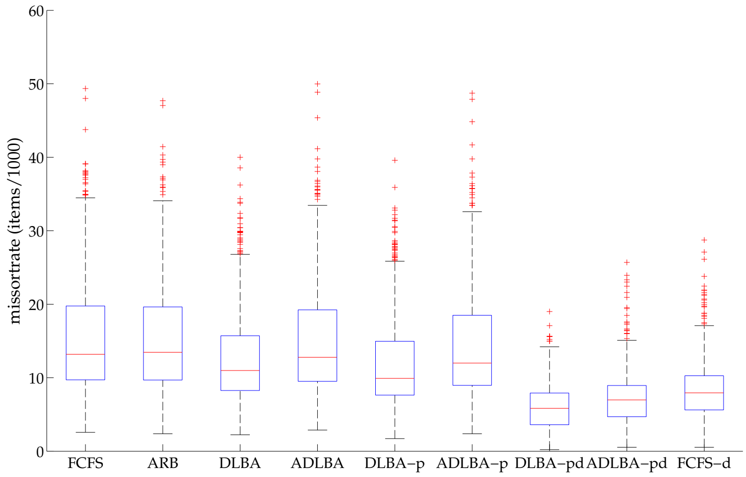
Figure H.8: PP-uneven — Total recirculation rate



(a) model 110

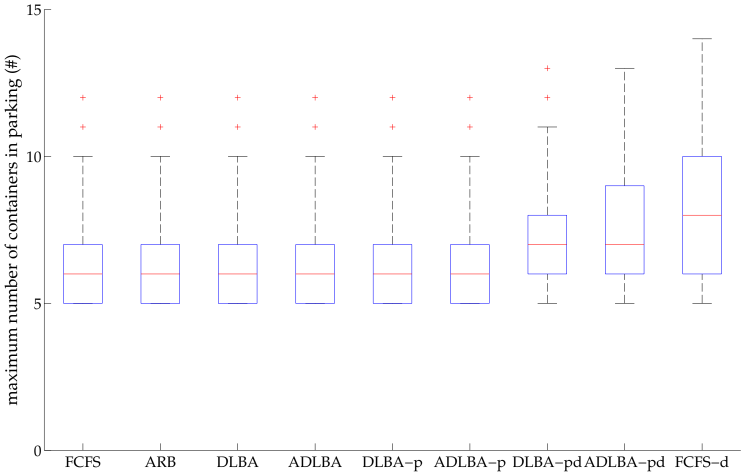


(b) model 120

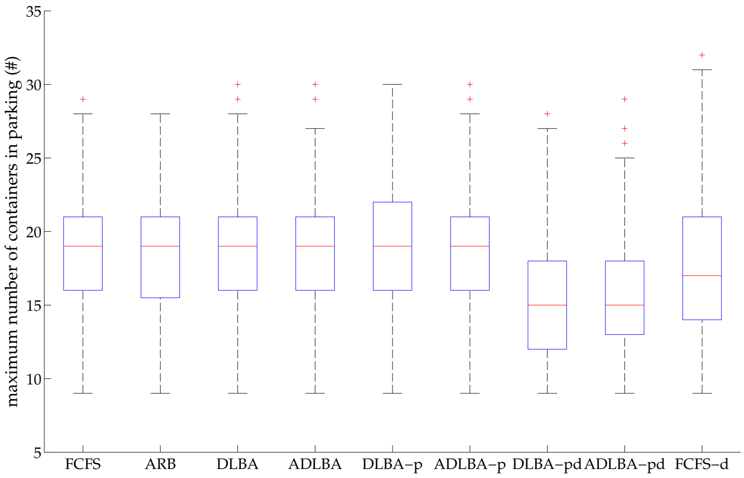


(c) model 22*c*

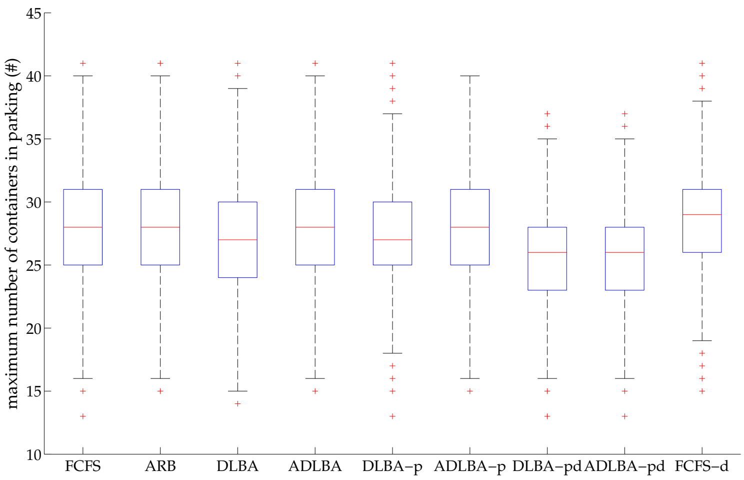
Figure H.9: BHS-even — Missort rate



(a) model 110

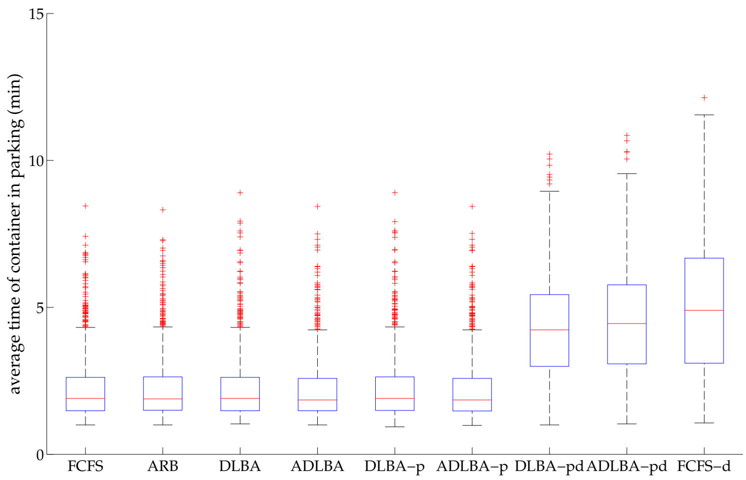


(b) model 120

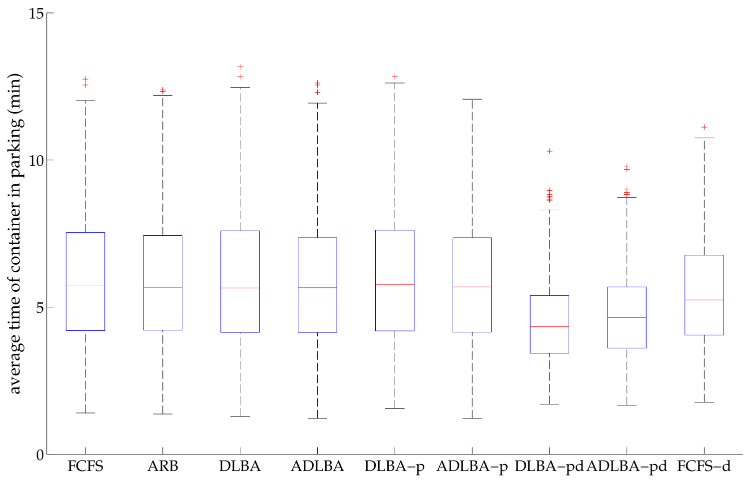


(c) model 22c

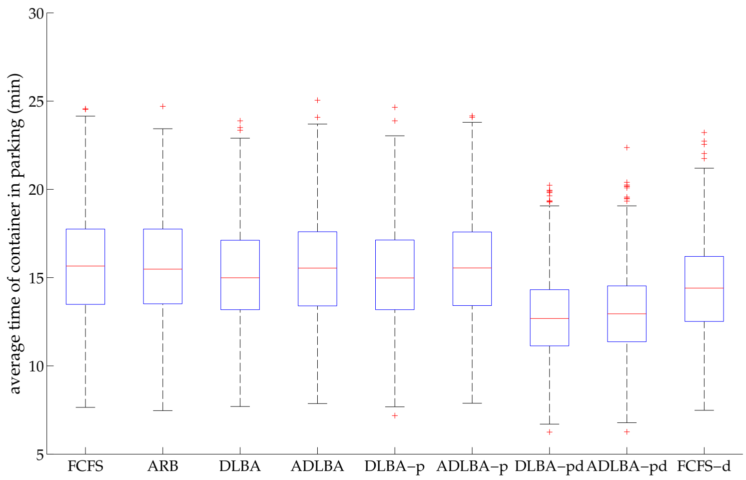
Figure H.10: BHS-even — Maximum number of waiting containers



(a) model 110

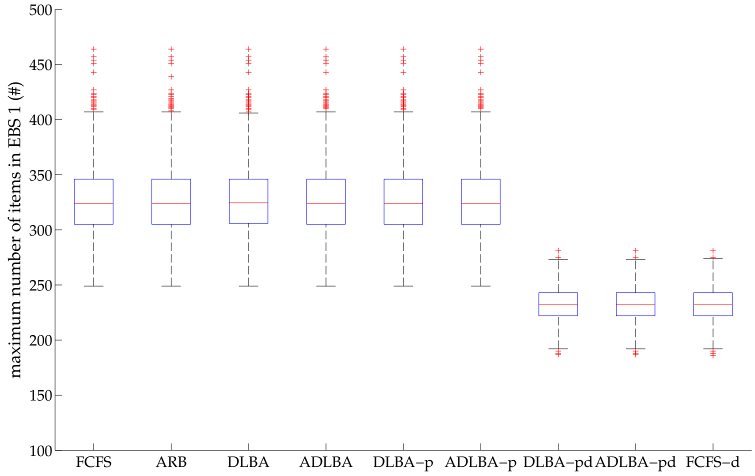


(b) model 120

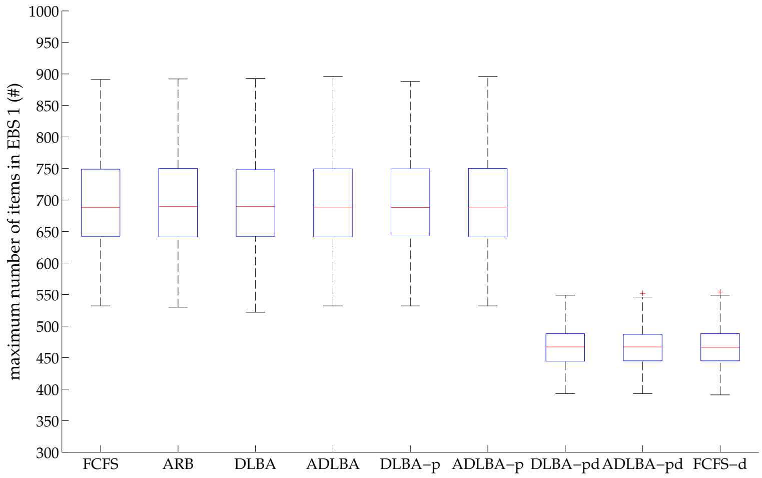


(c) model 22c

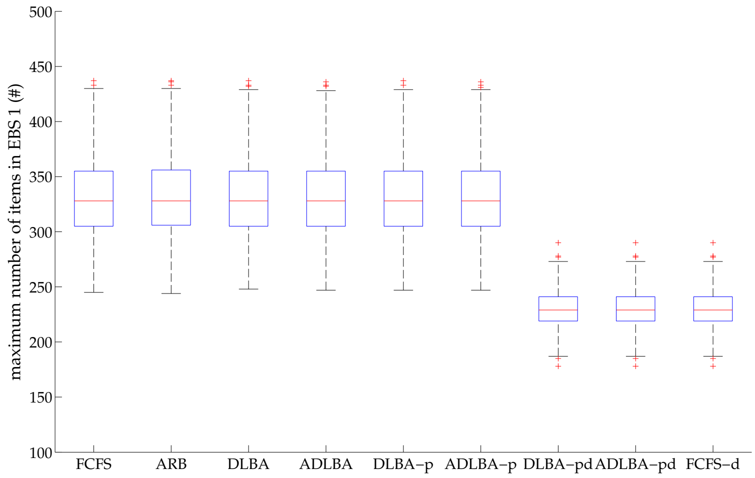
Figure H.11: BHS-even — Average container waiting time



(a) model 110



(b) model 120

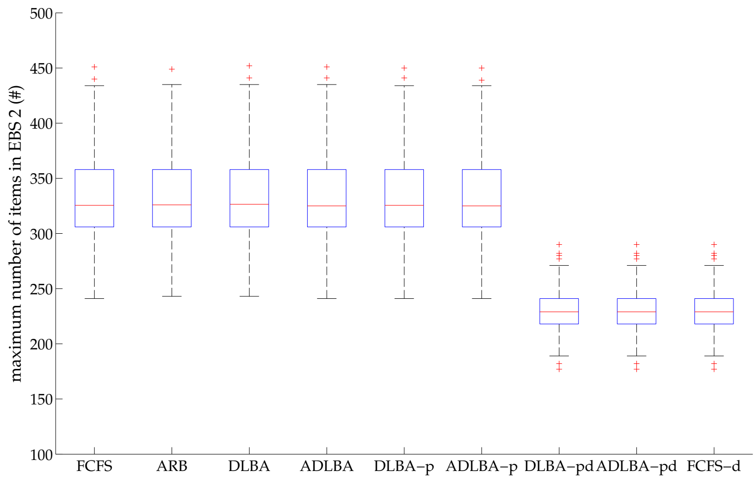


(c) model 22*c*

Figure H.12: BHS-even — Maximum number of items in EBS 1

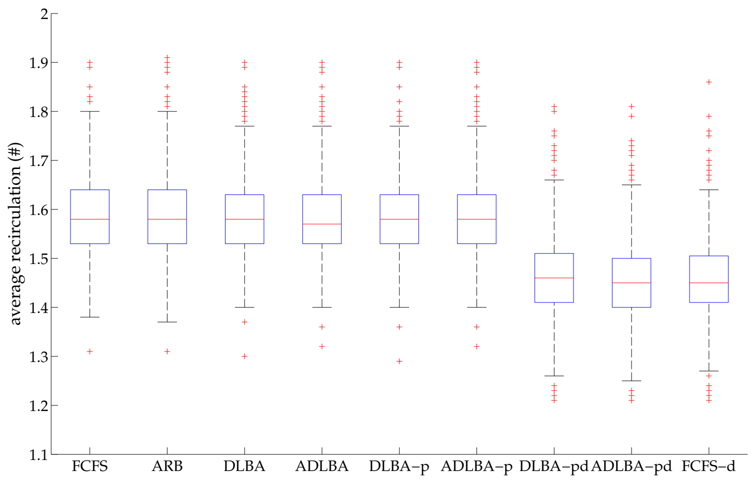
(a) model 110

(b) model 120

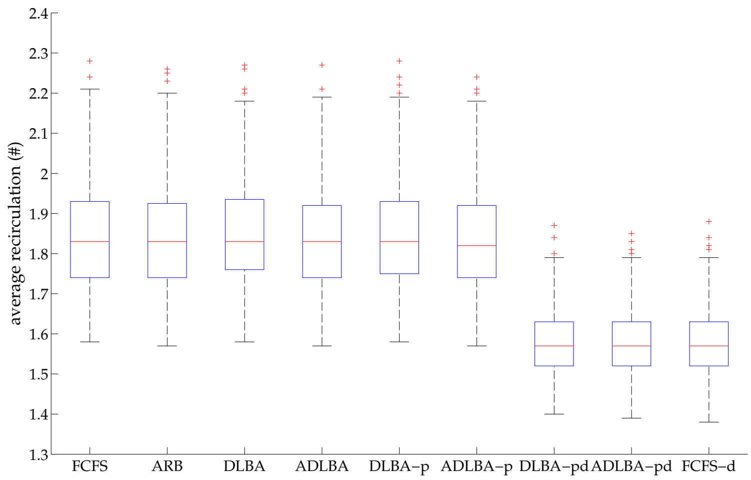


(c) model 22c

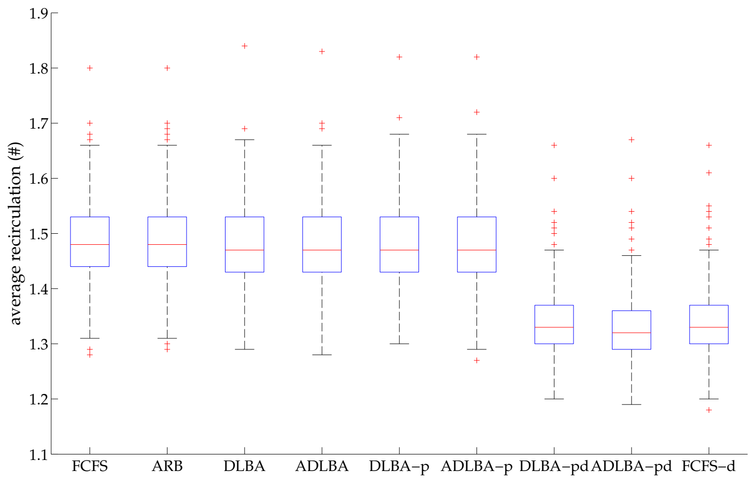
Figure H.13: BHS-even — Maximum number of items in EBS 2



(a) model 110

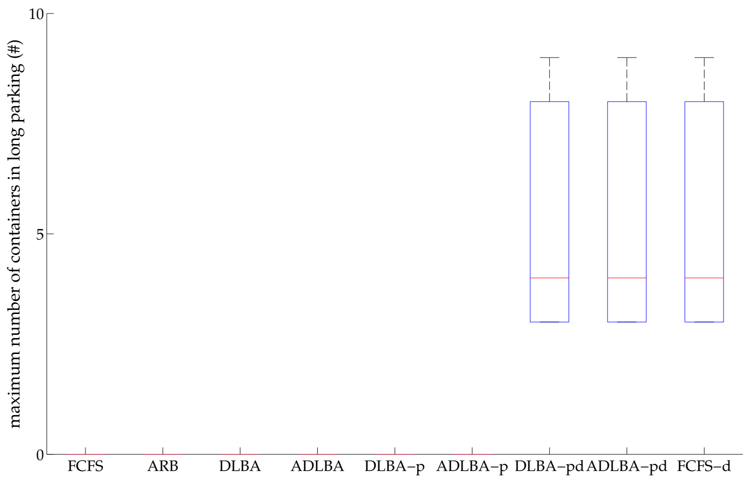


(b) model 120

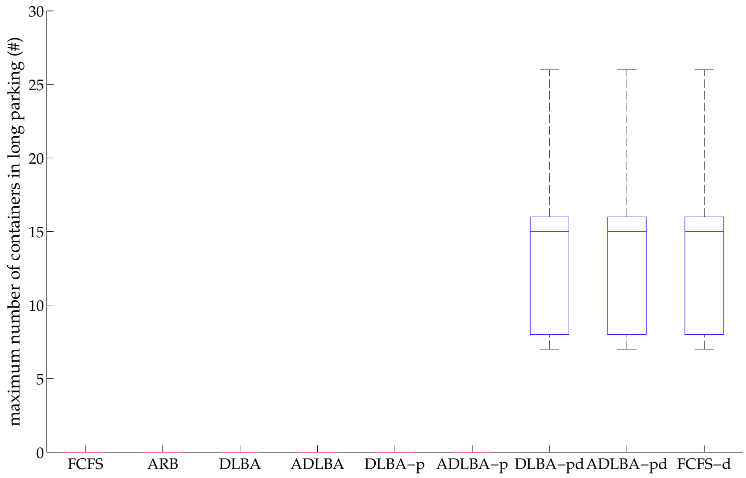


(c) model 22*c*

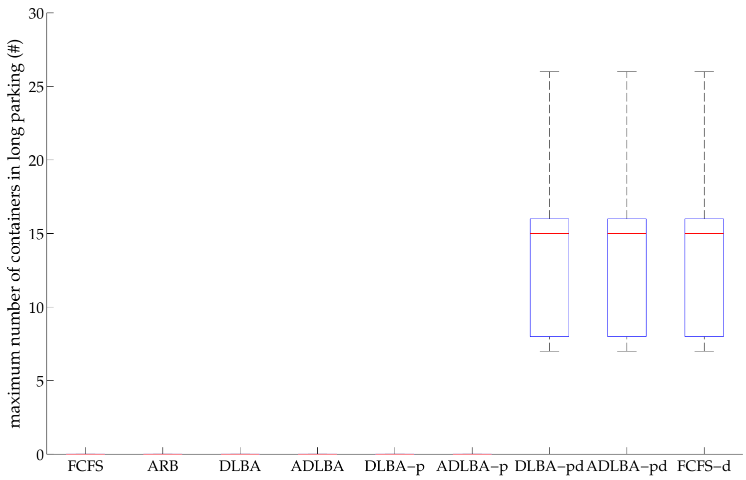
Figure H.14: BHS-even — Total recirculation rate



(a) model 110

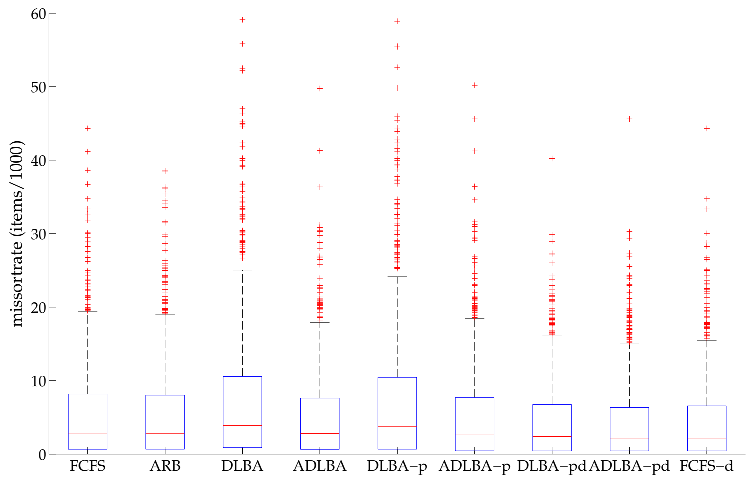


(b) model 120

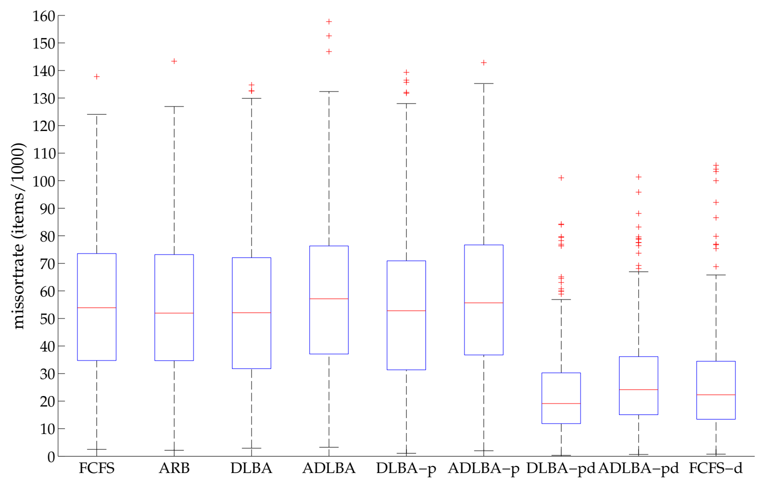


(c) model 22c

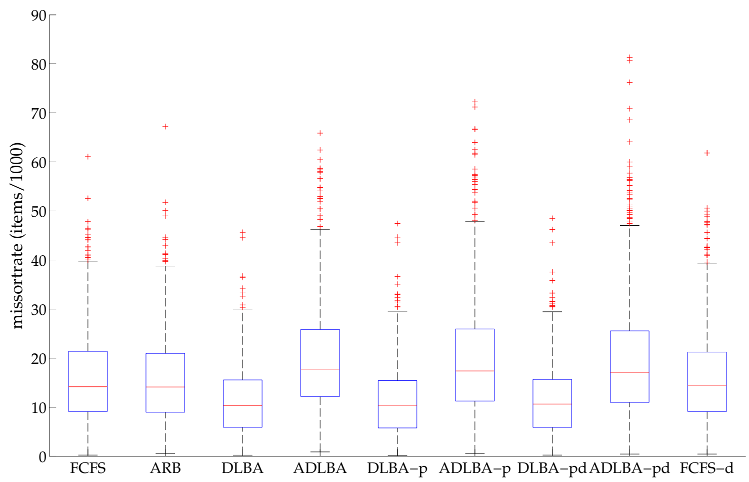
Figure H.15: BHS-even — Maximum number of containers in long parking



(a) model 110

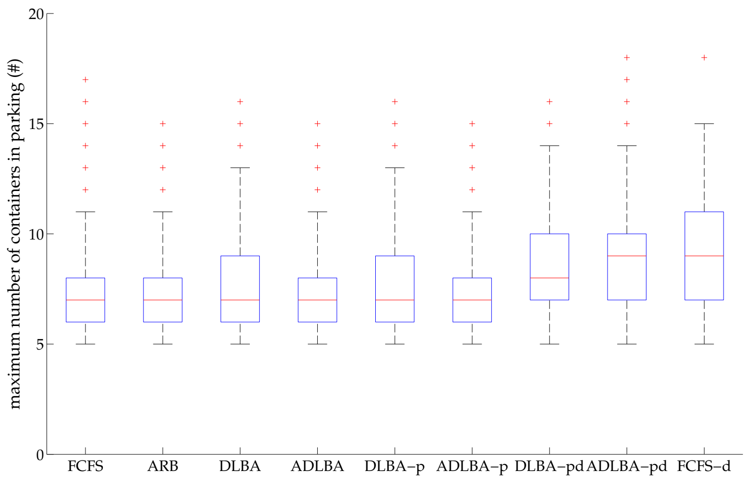


(b) model 120

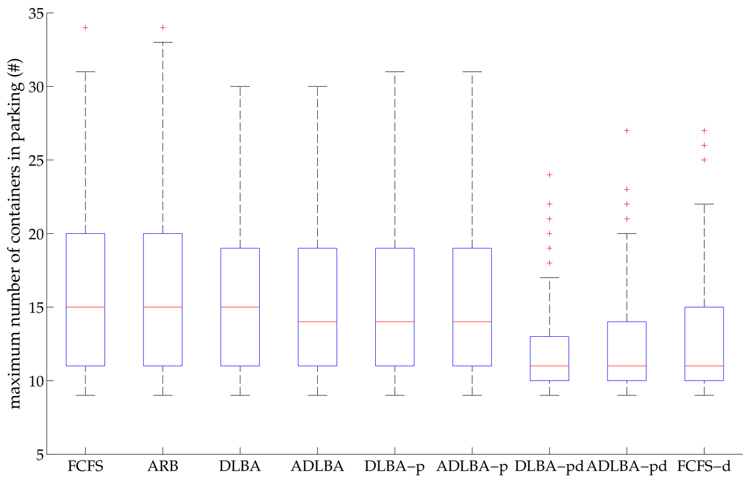


(c) model 22*c*

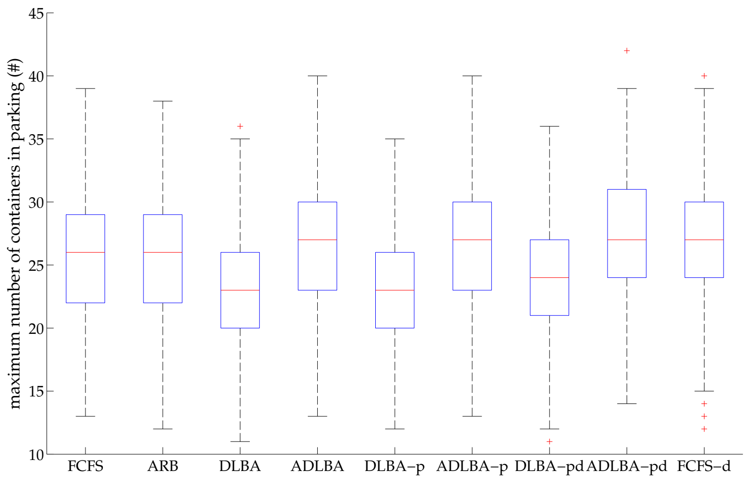
Figure H.16: BHS-uneven — Missort rate



(a) model 110

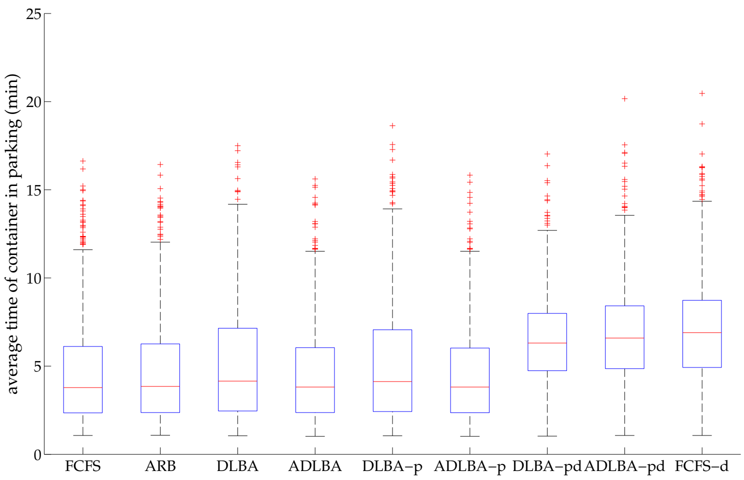


(b) model 120

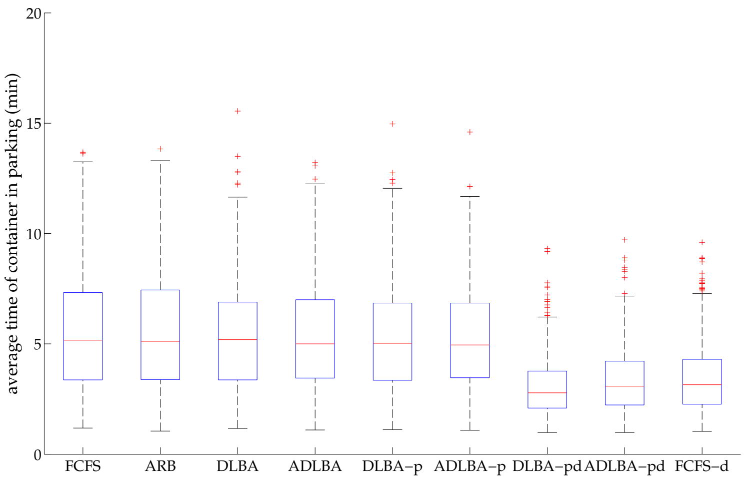


(c) model 22c

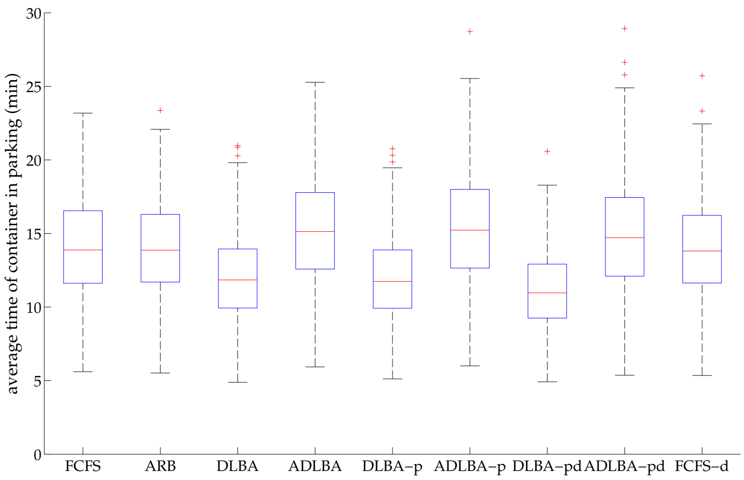
Figure H.17: BHS-uneven — Maximum number of waiting containers



(a) model 110

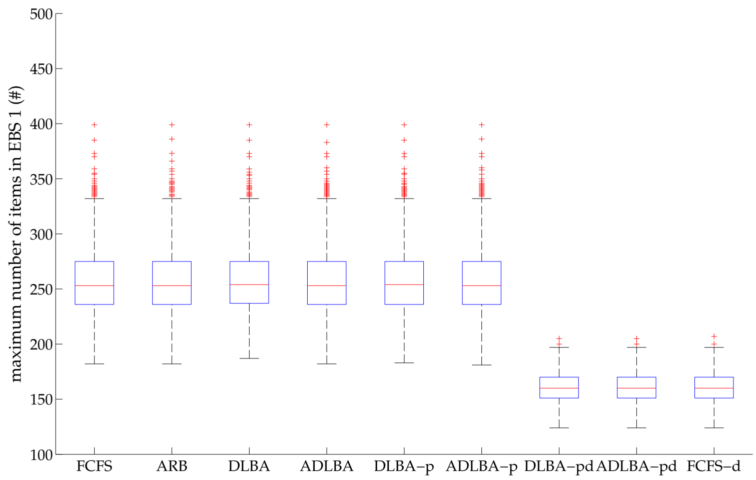


(b) model 120

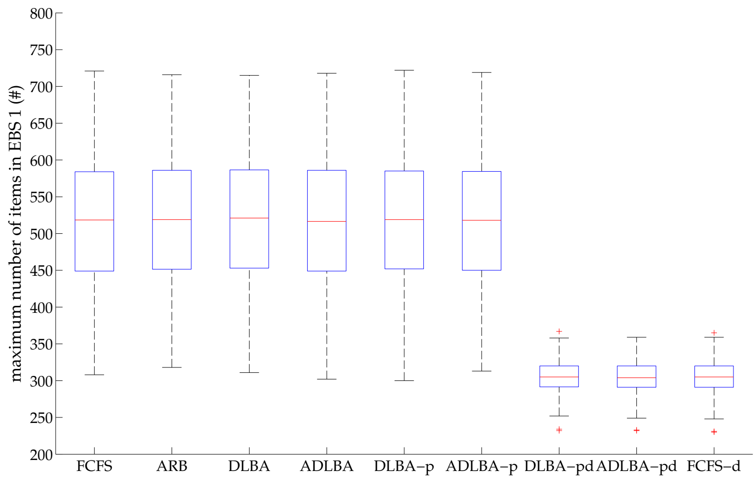


(c) model 22c

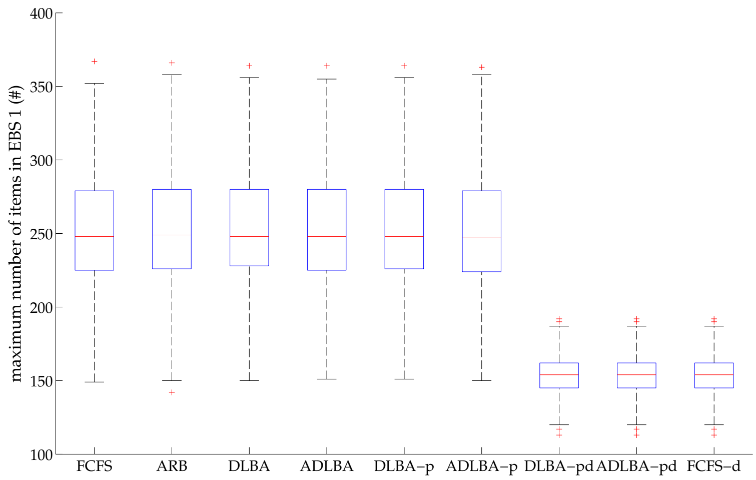
Figure H.18: BHS-uneven — Average container waiting time



(a) model 110

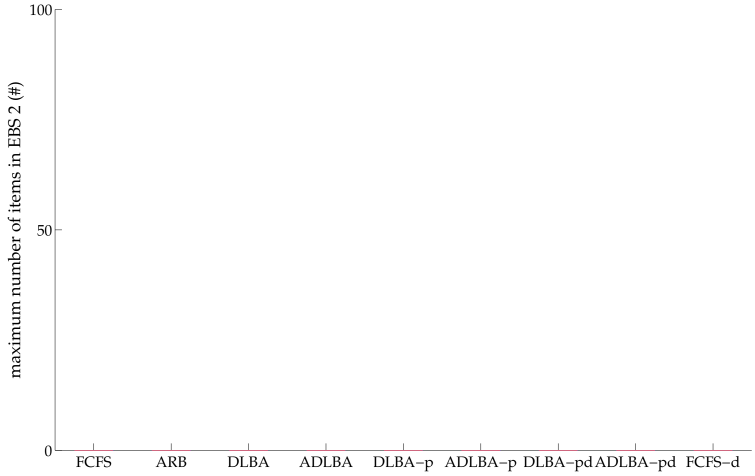


(b) model 120

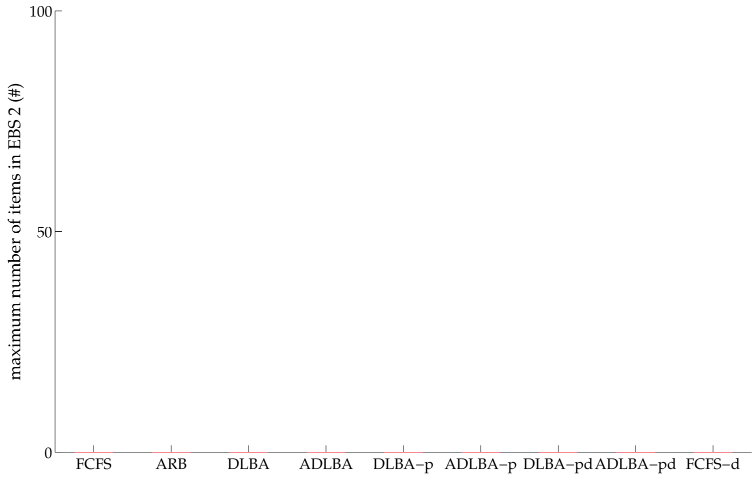


(c) model 22*c*

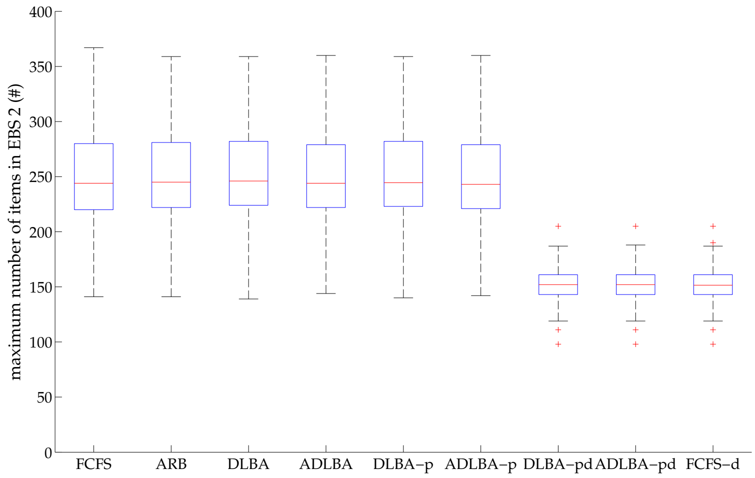
Figure H.19: BHS-uneven — Maximum number of items in EBS 1



(a) model 110

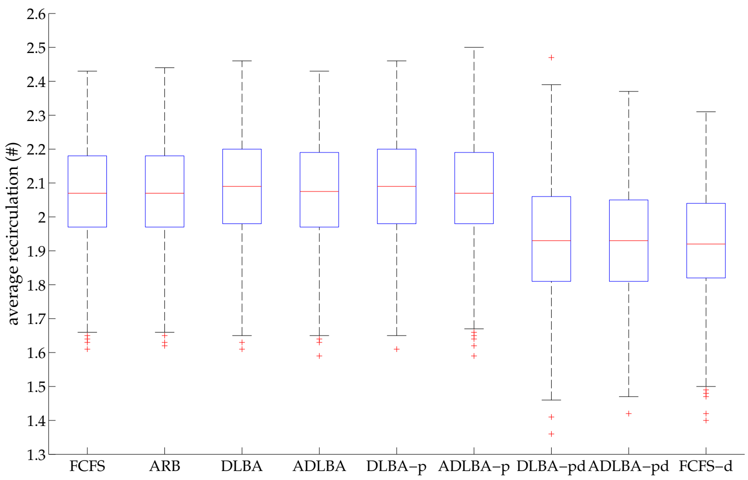


(b) model 120

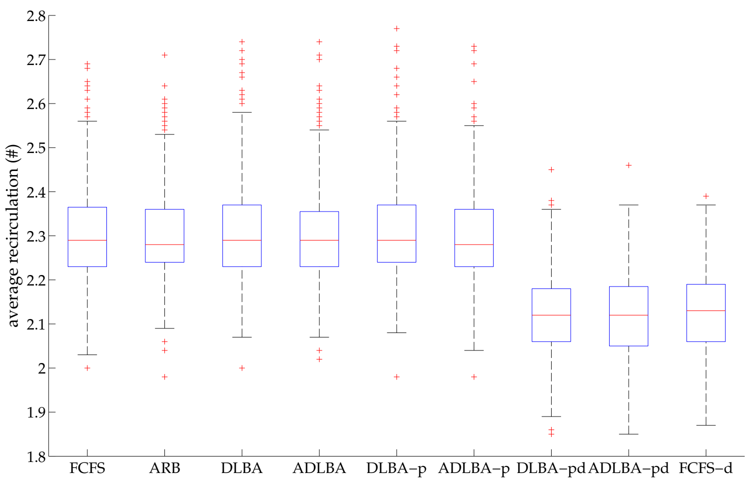


(c) model 22c

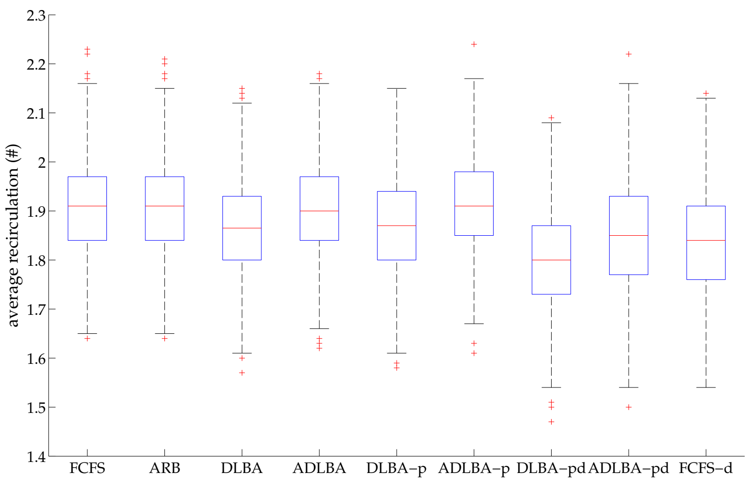
Figure H.20: BHS-uneven — Maximum number of items in EBS 2



(a) model 110

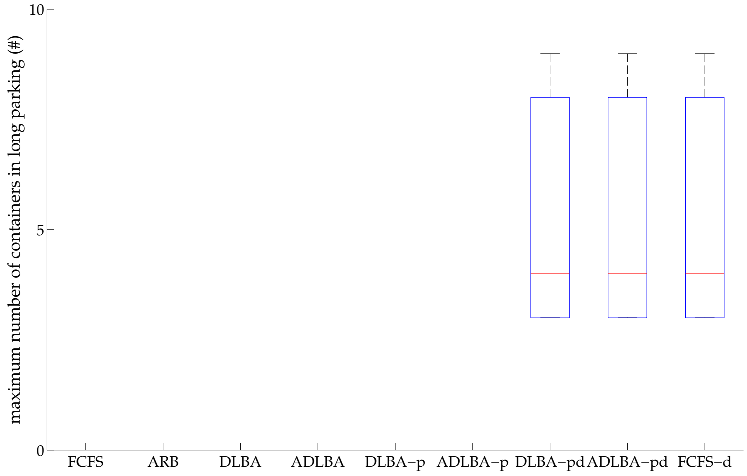


(b) model 120

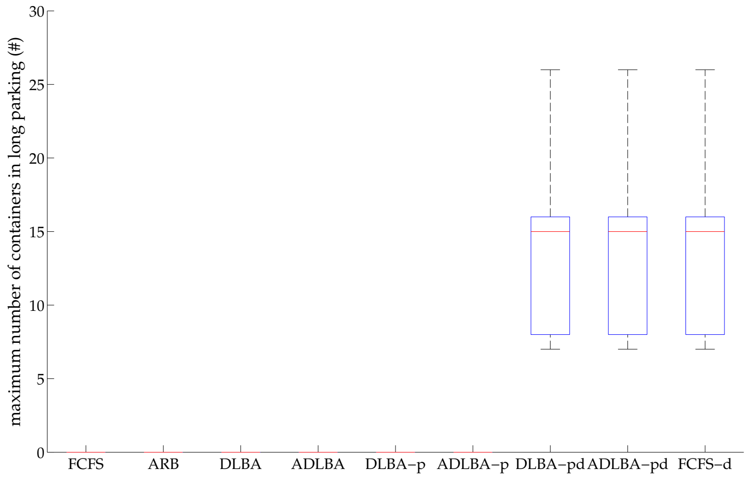


(c) model 22c

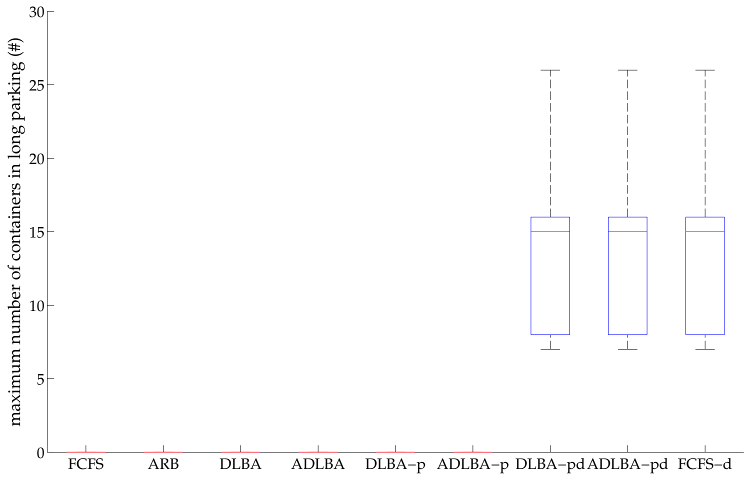
Figure H.21: BHS-uneven — Total recirculation rate



(a) model 110



(b) model 120



(c) model 22c

Figure H.22: BHS-uneven — Maximum number of containers in long parking