Research School for Operations Management and Logistics

Improving the performance of sorter systems by scheduling inbound containers

K. Fikse, S.W.A. Haneyah, J.M.J. Schutten

Beta Working Paper series 376

BETA publicatie	WP 376 (working paper)
ISBN	
ISSN	
NUR	804
Eindhoven	February 2012

Improving the performance of sorter systems by scheduling inbound containers

K. Fikse[a], S.W.A. Haneyah[a,*] and J.M.J Schutten[a]

[a] IEBIS group, University of Twente, The Netherlands.

k.fikse@alumnus.utwente.nl
s.w.a.haneyah@utwente.nl
j.m.j.schutten@utwente.nl

29th February 2012

Abstract

We investigate the inbound containers scheduling problem for automated sorter systems, in two different industries: parcel & postal and baggage handling. We build on existing literature, particularly on the dynamic load balancing algorithm designed for the parcel hub scheduling problem. We adapt the existing algorithm to a new industry, i.e., baggage handling, and develop it to cover more realistic operational conditions. Furthermore, we provide two extensions to our advanced dynamic load balancing algorithm, and conduct computational studies on different system layouts and given different scenarios. We analyse the efficiency of different scheduling approaches in different industries and different operational settings. One of the extensions that we propose is the delayability extension. We use this extension for the baggage handling industry in combination with currently used scheduling approaches. We find that it significantly improves the performance of these scheduling approaches.

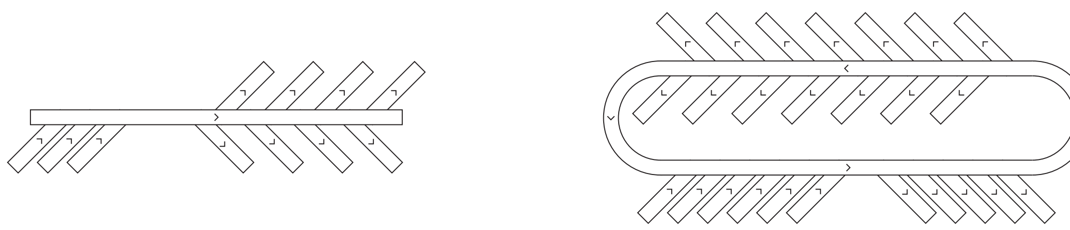
Keywords: Online scheduling; assignment problems; parcel & postal; baggage handling; sorter systems, inbound operations.

1 Introduction

In this paper, we focus on sorter systems that are a main component of Automated Material Handling Systems (AMHSs) used in the baggage handling and parcel & postal industries. We exploit the commonalities between the two industries to describe the sorter systems in a generic way. However, we distinguish the basic physical layout of a sorter

*Corresponding author.

system (a ‘line configuration’, Figure 1a) from more complex sorter systems with a ‘loop configuration’ (Figure 1b). We focus on sorter systems with a loop configuration.



(a) Line Configuration (b) Loop Configuration

Figure 1: Basic Physical Configurations of Sorter Systems

Currently, AMHSs suppliers focus on delivering the hardware, software, and maintenance services for sorter systems. However, 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 the suppliers’ scope. Given the fierce competition, AMHSs suppliers are interested in providing additional services to customers, e.g., providing customers with scheduling tools to use the sorter systems more efficiently. Such tools allow customers to increase throughput without installing expensive additional equipment. Customers that require a new system also benefit because they get a system with performance figures that could otherwise only be achieved by installing more expensive and space consuming equipment. Obviously, such services can significantly improve the competitive position of AMHSs’ suppliers, when offering systems with similar performance but lower costs by showing customers how to use the equipment efficiently. Therefore, this paper investigates scheduling tools that could lead to better use of the existing sorter capacity.

Although baggage handling and parcel & postal are two different industries, an interesting similarity is scheduling 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 & postal as well as baggage handling system-users typically apply a kind of first-come-first-served (FCFS) policy when deciding when and where to unload the *conveyables* (a term for baggage, parcels, and other items that have to be sorted *and* can be carried on a sorter system), although a lot is known about the contents of specific trailers and/or ULDs. As a result, uncontrolled peak flows for a particular outfeed could arise, causing it to fill up completely. These overloaded outfeeds may 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 transports the conveyables to the outfeed for unsorted conveyables, which is a large catchment area downstream the sorter system. The capacity of the sorter system is indirectly reduced, because the unsorted conveyables have to be re-loaded onto the sorter system 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. In a sorter system with a loop configuration, a full outfeed results in *recirculation*, i.e. the conveyable is transported through the entire sorter system again for a second delivery attempt. This reduces the sorter capacity directly, since a recirculating conveyable claims space that otherwise could have been used for another conveyable. In this context, balancing the workload over outfeeds may help reducing the overload incidents and thereby reduce recirculation. This in turn could increase the operational peak capacity on existing systems or reduce the required design capacity for future sorter systems. Therefore, the main problem we try to tackle is how to schedule the unloading operations of inbound containers using the knowledge about their contents, in order to identify and minimize peak flows in AMHSs.

Incoming ULDs at an airport contain either *transfer baggage* or *reclaim baggage* that is transported on dedicated unloading conveyors. We do not consider the flow of reclaim baggage further as it is not critical, and not part of the flow on the main sorter system. In addition to baggage from ULDs, there are bags arriving from check-in desks. As these arrivals are random and unpredictable, we treat them in our model as an *uncontrollable inflow*. Baggage handling systems also have a storage function for early baggage. When the make-up for a flight is 'open', one or more *lateral* (a type of outfeed conveyor frequently used in baggage handling) are assigned to handle the baggage for this flight. The baggage belonging to this flight is retrieved from the *early bag storage (EBS)* and merges with baggage already in transport on the sorter system to arrive at the destined make-up areas. In parcel & postal the temporary storage facilities are not in use: in parcel & postal an outfeed is usually assigned to a single destination during the entire shift. As a result, items in parcel & postal can always be assigned, and delivered, to their outfeeds. Whereas in baggage handling, an outfeed is assigned to multiple flights during the day, and so it is not always possible to deliver an arriving item to an outfeed. Actually, it is common that the outfeed is not yet known upon arrival.

Using a sufficient level of aggregation, Figure 2 presents a combined process model for sorter systems in both industries. In this model, we can set the uncontrollable flow equal to zero to model a parcel & postal sorter system, where no uncontrollable flow of check-in items exist. Likewise, a zero capacity temporary storage models a parcel & postal sorter.

The remainder of the paper is organized as follows: Section 2 presents a literature review of relevant studies. Section 3 develops the advanced dynamic load balancing algorithm that builds on state-of-the-art scheduling approaches for sorter systems. Section 4 further extends the algorithm to deal with priority containers and delayable containers. Section 5 presents the experimental setup and the results of computational experiments. Finally, Section 6 ends with concluding remarks.

2 Literature Review

The 'parcel hub scheduling problem' (PHSP), introduced by (McWilliams, Stanfield, & 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. McWilliams et al. (2005) use a sorter system in line configuration, where they try to minimise the *makespan* of the sorting process. 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 randomly assigns available containers

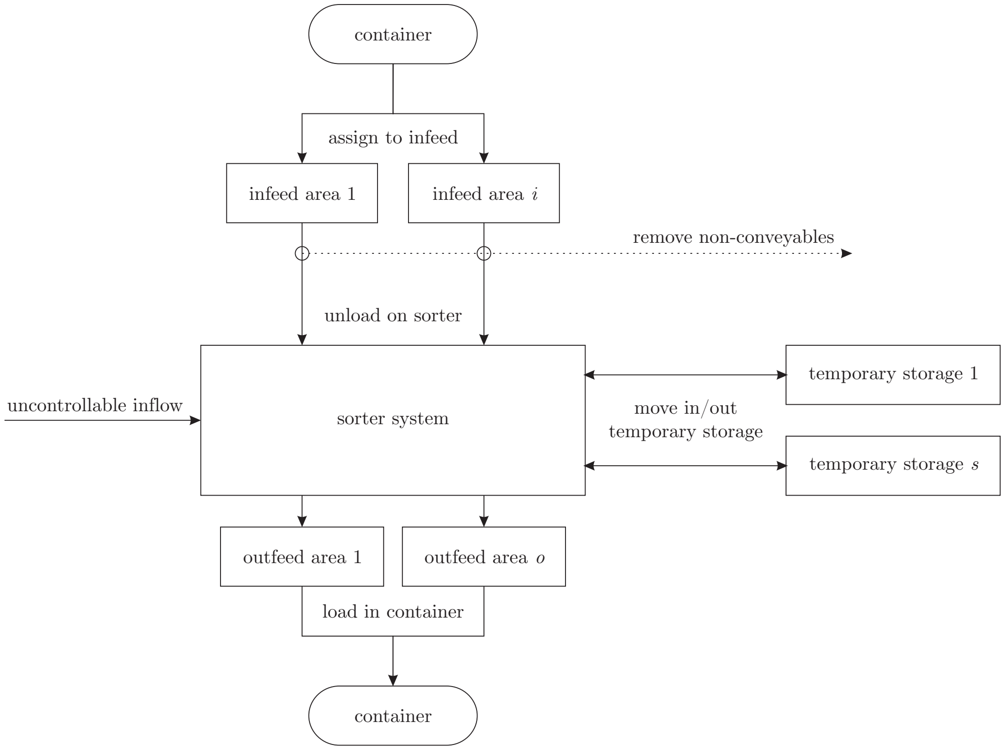


Figure 2: Combined Process Model

to available infeeds. McWilliams (2005) shows that similar results can be achieved using iterative local search or simulated annealing techniques. McWilliams (2009a) aims for an approach to balance the workload on the loading docks. He solves small problems to optimality using a binary minimax programming model. For big problem instances, he uses a genetic algorithm that outperforms the SBSA and ARB approaches used in McWilliams et al. (2005). A drawback of this approach is that due to the minimax problem, there may exist many optimal solutions in a very large non-convex solution space. 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 large problems require more time to solve.

Recently, McWilliams (2009b) developed a relatively simple dynamic load balancing algorithm (DLBA). 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. He finds that this simple algorithm performs much better than random assignments (makespan reduction of 15%). Furthermore, it appears that this DLBA is generally better (makespan reduction of 8%) in large complex problems than the approach of McWilliams (2010). However, a number of restrictive assumptions, e.g., concerning the arrival and departure processes and (un)loading speed, limit the practical application of the DLBA.

A problem close to ours is the crossdocking problem, for which Cohen and Keren (2009) develop an algorithm given forklifts as the mean of freight transport. The algorithm does not suit our problem where conveyors are the mean of transport. Although a crossdock is defined as a no-inventory sorting facility, many studies explicitly use temporary storage. Li, Low, Shakeri, and Lim (2009) consider the situation in which the floorspace in the centre of the facility is used to temporarily store products. Li et al. study a problem where each inbound trailer is also an outbound trailer that has to be loaded directly after it has been unloaded, unlike our problem. They use a heuristic based on the parallel uniform scheduling problem. Yu and Egbelu (2008) focus on coping with the possibilities of limited intermediate storage, by scheduling the inbound and outbound operations of a crossdock to minimise the makespan of the operation. They provide both a mathematical model to solve the scheduling problem to optimality and a quite extensive heuristic algorithm. However, their approach requires a number of restrictive and unrealistic assumptions, e.g., all trailers are available at the start of the operation and the unloading sequence of products from an inbound trailer can be determined.

McAree, Bodin, and Ball (2002) test the Bin and Rack Assignment Model (BRAM) using a realistic case from a large package sort facility. This algorithm was specifically designed for air terminals where inbound ULDs are assigned to bins to be broken into individual pallets. The main goal of McAree et al. is to minimize the operational cost. Because the BRAM is too complex to solve, they develop 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, Bodin, Ball, and Segars (2006) find solutions for the different layouts with running times ranging from few minutes to few hours, which is quite fast for large scale investment decisions, but too slow for online scheduling decisions.

Gue (1999) determines which docks to use for unloading and which for loading in a crossdock facility. The author uses a simple algorithm based on scheduling rules and logic similar to that in the approaches of McWilliams (2009b) and Yu and Egbelu (2008).

Werners and Wülfing (2010) 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. Werners and Wülfing aim at minimizing the total transport effort, i.e. reducing the total distance travelled on the sorters. In order to solve this large complex problem, they hierarchically decompose the problem into two subproblems. Werners and Wülfing show that their approach ensures a balanced workload over the different areas in the sorting centre, whilst providing robust solutions. However, they do not discuss the inbound unloading process, they solely focus on scheduling the outbound process.

In the baggage handling field, Robusté and Daganzo (1992) provide an extensive overview of the possible presorting strategies, whilst aiming at minimising baggage handling costs. They 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. Robusté and Daganzo conclude that airlines could achieve significant cost reductions if they would segregate the baggage for the larger destinations at the origin airport.

Abdelghany, Abdelghany, and Narasimhan (2006) address another problem that recently received a lot of attention, which is the outbound assignment problem, i.e. assigning outfeeds (make-up areas) to specific flights. Frey, Artigues, Kolisch, and Lopez (2010) apply a mathematical approach for a 'baggage handling system scheduling problem'. They consider a baggage handling facility with an EBS system, and assign flights to workstations and carousels. They solve a decomposed problem to determine when to retrieve bags from the EBS. This problem could be converted into a scheduling problem for inbound containers, but there are two main limitations due to: First, the assumption that full knowledge is available is not fullfilled. Second, the runtime of the algorithm is too long.

Although not entirely related to the scheduling and assignment literature, Hallenborg (2007) presents an interesting approach to determine the 'urgency' of a bag. Although he focusses on agents-based scheduling in *DCV (Destination Coded Vehicles)* baggage handling systems, the urgency equation of a bag may be useful for us to determine the urgency of a *container* of bags. Hallenborg proposes an approach where a bag becomes urgent when it has a time allowance below a threshold U_t remaining, before the destined make-up lateral closes (*cutoff time*). U_{max} is the maximum time a bag can have before cutoff. 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:

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

When the time until cutoff decreases, urgency increases at a decreasing rate until it is 0 at time U_t. From that time on, the bag becomes urgent, its urgency increases at an increasing rate until it is infinity when the make-up area closes, i.e. when $t = 0$. This approach may provide a good solution to determine which containers need to be unloaded first in order to ensure that their contents are indeed timely sorted.

From our literature review, we conclude that there is no instant solution to our problem, but we find the parcel hub scheduling problem (McWilliams et al., 2005), especially the Dynamic Load Balancing Algorithm (DLBA) by McWilliams (2009b), to be the most relevant study from different points of view. First, the DLBA is an online algorithm that does not require full knowledge about incoming containers, but uses existing knowledge about containers that are already at the sorting hub. Second, it is a relatively simple and fast approach, which can be implemented in practice. Third, McWilliams (2009b) reports impressive reductions in the makespan of the sorting operation. Finally, it represents a good starting point, being an approach that can accommodate extentions, and adaptation to the baggage handling industry. Section 3 builds further on this conclusion.

3 Advanced Dynamic Load Balancing Algorithm

McWilliams (2009b) propose the Dynamic Load Balancing Algorithm (DLBA) to solve the Parcel Hub Scheduling Problem (PHSP) by McWilliams et al. (2005). The DLBA aims at balancing the workload over the different outfeeds, in order to minimise the probability of an outfeed being overloaded. Based on our literature review, we find that this is the state-of-the-art approach that may act as a starting point for our online scheduling problem. Our problem has two main additional issues to research. First, the assumption of inbound and outbound containers with equal priority, does not hold for the baggage handling industry where flights, and as a result bags, have different deadlines. Second, the DLBA assumes zero internal transport times on the sorter. As a result, a parcel unloaded from a truck on an infeed is immediately loaded onto another truck at an outfeed. Note that the assumption of zero internal transport times is no stronger restriction than equal and fixed internal transport times between any infeed-outfeed pair. This restriction might be a valid simplification for sorter systems with certain configurations, or generally speaking, when unloading a container requires much more time than the internal transport of items on the sorter. However, this may not hold for all sorter systems, especially not for large baggage handling systems where a ULD (40 bags) can be unloaded in less than 5 minutes onto a large sorter systems with loop configuration, multiple infeed areas, and route complexities.

In this section, we develop the DLBA into an advanced dynamic load balancing algorithm (ADLBA), which takes (unequal) transport times into account. The DLBA only has to keep track of the total number of items in the system destined for a specific outfeed. The assumption is 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 balanced. Incorporating transport times means that the workload should not only be balanced over the different outfeeds, but also over time. Determining for each outfeed at each point in time the expected *outflow* (the number of items that arrive at the chute) indicates whether the capacity of the outfeed is exceeded or not. However, not only the volume of excess items is relevant, but also the rate at which these excess items arrive is important. Therefore, we use the squared value of excess flows as an optimisation criterion to heavily penalise large excess flows. Another possible goal function would be a minimax goal function that minimises the maximum excess figure. Drawback of this approach is that a solution in which one outfeed exceeds its capacity by $n + 1$ items is considered worse than a solution in which all outfeeds exceed their capacity by n items, whereas in the latter case many more items are forced to recirculate. Determining the squared excess outflow continuously, is impractical. A computationally less challenging approach, is to use time buckets. In the time bucket approach, we determine for each item in which time bucket it is likely to arrive at the outfeed. The size of the time buckets is an important model parameter, for it affects the level of detail that can be achieved. In order to achieve sufficient 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 under study. Using time buckets of 1 minute provides sufficient detail but also results in valid and meaningful outflows. A concept related to time buckets is container *segments*, where we divide the load of each container into fictitious segments of equal size, each needing exactly one time bucket to be unloaded.

By including the internal transport times, the formal problem description (FPD) increases in complexity from the integer linear program for the PHSP McWilliams (2010) provides. Unlike the PHSP, for our problem infeeds are not identical. Therefore, it is important to know at which infeed a container is docked, since travel times to outfeeds can differ amongst infeeds. Equations 2 to 7 provide the FPD of the problem for which the ADLBA was designed. Table 1 provides a full overview of the notation.

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) needed to unload container 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}	total squared excess outflow

Table 1: Notation for the formal problem description of the ADLBA

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

$$\text{subject to} \qquad \sum_{c \in C} \sum_{s=1}^{S_c} x_{csut} \leq 1 \qquad \forall\, t,\, u \tag{3}$$

$$\sum_{u \in U} \sum_{t \in T} x_{csut} = 1 \qquad \forall\, c,\, s \tag{4}$$

$$S_c \cdot x_{c1ut} - \sum_{s=1}^{S_c} x_{csu(t+s-1)} = 0 \qquad \forall\, c,\, u,\, t \tag{5}$$

$$\sum_{c \in C} \sum_{s=1}^{S_c} \sum_{u \in U} \left(\frac{f_{cl}}{S_c} x_{csu(t-t_{ul})} \right) - F_l \leq EF_{lt} \qquad \forall\, l,\, u,\, t{>}t_{ul} \tag{6}$$

$$x_{csut} \in \{0, 1\} \qquad \forall\, c,\, s,\, u,\, t \tag{7}$$

The objective function 2 sums the squared value of the excess outflow EF over all the load docks and time buckets. Constraint 3 ensures that each unloading dock is used by at most one container segment per time bucket, where a container is divided into S_c segments of equal size, such that in one time bucket each unload dock can unload exactly one container segment. Constraints 4 and 5 are similar to the ones proposed by McWilliams (2010), 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. The first term of constraint 6 incorporates the internal transport times t_{ul}. In order to measure the outflow at load dock l at time t the flows that were generated by the unloading docks at time $t - t_{ul}$ have to be used. The second term of constraint 6 ensures that outflows that exceed the capacity F_l force the value of EF_{lt} to be positive. Finally, Constraint 7 ensures that all decision variables are binary. This mathematical model merely describes the static form of the problem. In order to solve it, we need full knowledge about incoming containers which is unrealistic, and would lead to an intractable problem. We could also solve this model repeatedly, e.g., whenever a sufficient number of containers arrive, but this is impractical because the solution of already assigned containers may change, and delays in dispatching containers would occur due to waiting container arrivals and solving time of the model. As a result, a dynamic approach, i.e., the ADLBA, is preferred.

The main idea of the ADLBA is to extend the DLBA by incorporating internal travel times, but still does not incorporate possible traffic delays on the sorter. Moreover, we make both algorithms applicable to a baggage handling scenario. To do so, we need to update the system information according to arrivals occuring in real-time. This proceeds as follows (refer to Table 2 for an overview of notations): first, to process a checked-in item using the DLBA, we increase the counter for the item's destination by one as soon as it is identified in the system, i.e., *announced*. However, the ADLBA also determines the expected arrival time at the destined outfeed (TB_{start}):

$$TB_{start} = \left\lceil \frac{ac + t_{CI} + t_{io}}{tb} \right\rceil \tag{8}$$

where ac denotes the current time in seconds and a delay time t_{CI} is added, because items are announced before they actually arrive at the check-in infeed of the main sorter. 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 no traffic delays. Dividing this time by the size of a time bucket tb and rounding down then adding 1, provides the time bucket index at which the item would arrive. Finally, the variable $FLOW(TB, o)$, which keeps track of the expected outflow, is updated (Figure 3a).

Second, to request items from the EBS (when their destination is assigned to an outfeed), the DLBA only requires knowledge about which destination (d) is assigned to which outfeed (o). However, the ADLBA also requires information about the number of items in the EBS for this destination (EBS_d) and the travel time from the EBS to the outfeed. Figure 3b shows the procedure to update the expected outflows when items are requested from the EBS. For the travel time the notation t_{io} is used, as the EBS is modeled as 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 are determined using:

$$TB_{start} = \left\lceil \frac{ac + \frac{1}{F_i/3600} + t_{io}}{tb} \right\rceil \tag{9}$$

$$TB_{end} = \left\lceil \frac{ac + \frac{1}{F_i/3600} \cdot EBS_d + t_{io}}{tb} \right\rceil \tag{10}$$

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.

ac	current 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 2: Notation for ADLBA equations

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 storage, the time required to unload one item is $\frac{3600}{F_i}$ seconds. This, again, ensures that the numerator of equation 9 displays the exact time bucket in 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, the procedure assumes that the items arrive at the sorter homogeneously over the different time buckets. The number of items that arrive per time bucket (TB_{flow}) is therefore:

$$TB_{flow} = \frac{EBS_d}{TB_{end} - TB_{start} + 1} \tag{11}$$

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_{end}.

Another event common to both industries, is when the dispatcher decides to dock a specific container at an outfeed. In this case, an approach similar to the one used for the EBS is applied. For each possible destination, the assigned outfeed o and the values of

TB_{start}, TB_{end}, and TB_{flow} are determined (Figure 3c) using the following equations:

$$TB_{start} = \left\lceil \frac{ac + \frac{1}{F_i/3600} + t_{io}}{tb} \right\rceil \tag{12}$$

$$TB_{end} = \left\lceil \frac{ac + \frac{1}{F_i/3600} \cdot C_{tot} + t_{io}}{tb} \right\rceil \tag{13}$$

$$TB_{flow} = \frac{C_d}{TB_{end} - TB_{start} + 1} \tag{14}$$

The time required for unloading $\left(\frac{1}{F_i/3600} \cdot C_{tot} \text{ seconds}\right)$ thus depends on all items inside the container, whilst the flow arriving at a specific outfeed depends only on the destinations that are assigned to this outfeed.

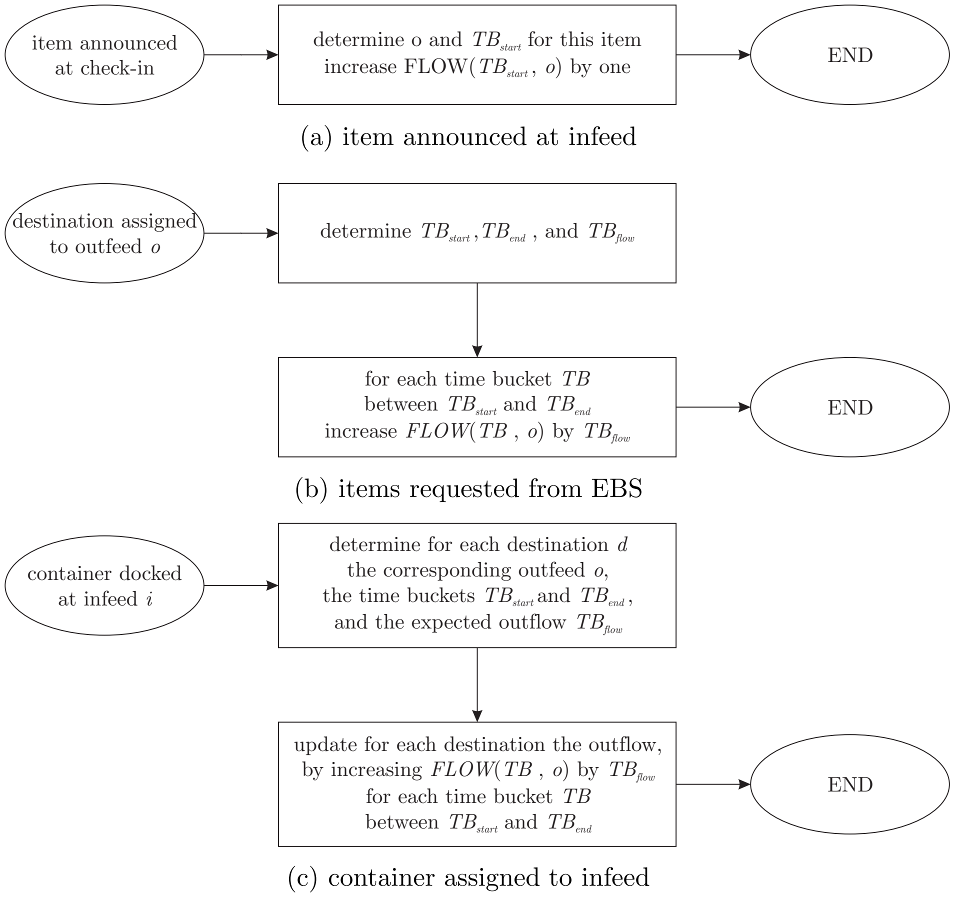


Figure 3: ADLBA procedures for updating system information

Ideally, the exact solution requires enumerating the expected excess outflow for each container-infeed combination, and then selecting the set of container-infeed combinations that minimise the total excess outflow. Since this approach is computationally expensive, a constructive heuristic is proposed: 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 value for an assignment decision of a specific container to a specific infeed, is relatively simple and uses the system information from the $FLOW$ variable. However, time buckets in past are irrelevant, and information about time buckets that are relatively far in the future are not reliable, because recirculation and merging difficulties may alter these predictions. We focus 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 about these time buckets is 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 12 –14 and the procedure for assigning containers to infeeds explained earlier. Finally, the objective value for each decision can be determined using equation 15. 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. Based on our discussion, Figure 4 presents the ADLBA.

$$EF_{ic} = \sum_{TB \in TB_{horiz}} \sum_{o \in O} \left(\max \left\{ 0, FLOW_{ic}(TB, o) - \frac{F_o \cdot tb}{3600} \right\} \right)^2 \tag{15}$$

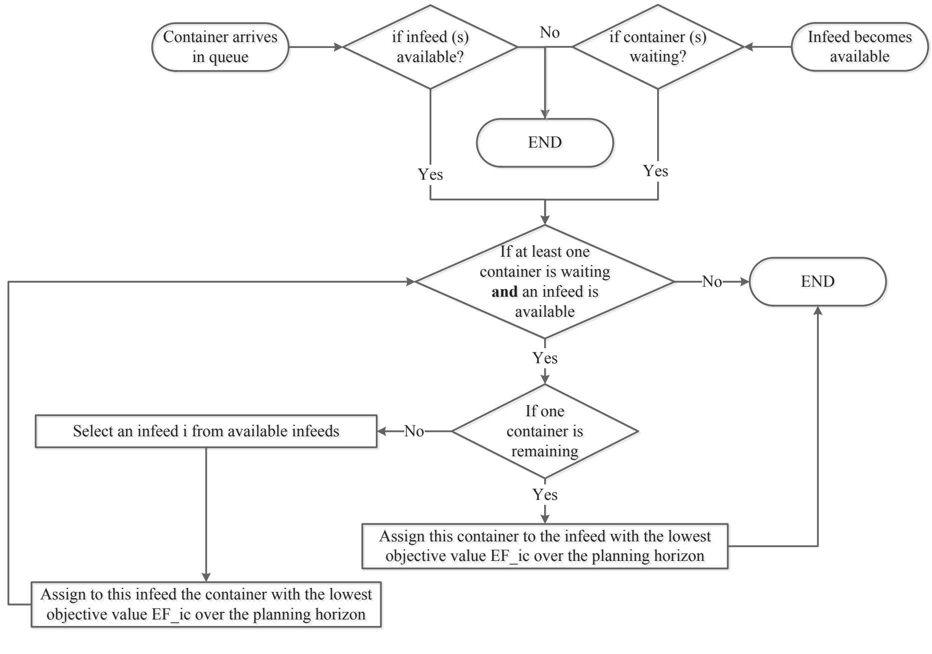


Figure 4: Main logic of the ADLBA

4 Extensions: Urgency & Delayability

Section 3 focussed on the integration of internal travel times in the DLBA. There are, however, two other issues that should be taken into account when assigning containers to available infeeds, particularly in baggage handling systems.

4.1 Urgency

Hallenborg (2007) provides an approach to determine a bag's urgency (Equation 1). We build on this approach to calculate the urgency of a container of bags. Note that at a certain point in time, it is physically impossible to transport items through the sorter system to the correct outfeeds before they close. For our problem, items for destination d become non-urgent if they have less than a duration of time U_{end}, remaining before cutoff time. We set this time duration equal to the internal transport time t_{io}, where i is the infeed under consideration, and o is the outfeed destination d is assigned to. That is, an item becomes non-urgent when it cannot be on time, even if it was the first to be unloaded from a container. Unfortunately, this extension does not suit scheduling approaches that do not keep track of internal transport times (e.g., DLBA, FCFS, and ARB), for which we use a fixed value of 5 minutes that is an estimation of the average internal transport times in our experimented system layouts. Another relevant time threshold is U_{start}, where bags become urgent if they have less than this threshold remaining before cutoff time. Figure 5 shows these time indicators at a certain moment in time t, where t_d is the remaining time until the outfeed assigned to destination d closes.

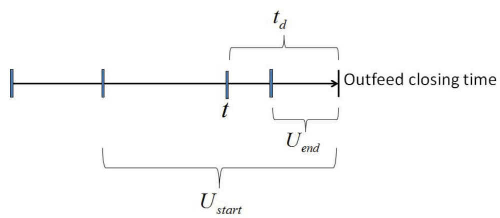


Figure 5: Important time indicators.

We use a relatively simple approach from practice, where each destination is urgent for 30 minutes and U_{start} is therefore equal to $U_{end} + 30$ minutes. Finally, the urgency of a destination is modelled in such a way that it starts at zero when a bag has U_{start} time remaining and equals one when it has U_{end} time remaining. Urgency of a destination d at some point in time t is thus determined using equation 16.

$$u_d(t) = \begin{cases} \frac{t_{io}}{t_d} \cdot \frac{U_{start} - t_d}{U_{start} - t_{io}} & U_{end} \leq t_d \leq U_{start} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

To determine the urgency of a container, we propose one of three simple approaches. 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. The 'maximum' measure would often fail to correctly differentiate between the urgencies of containers. For instance,

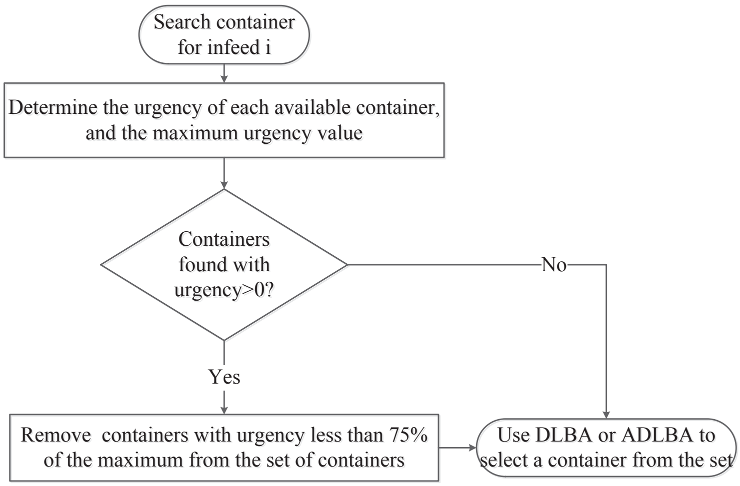


Figure 6: Flowchart for Priority Scheduling

if there is one urgent destination, then a container that holds one item for this destination is as urgent as a container that holds ten. The 'average' measure 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. However, this assumption is the drawback of this approach, as 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 'sum' approach solves this issue, and so it is used in this research.

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 the sum of individual urgencies, as follows:

$$u_{\text{sum}} = \sum_{j \in J_c} u_{d(j)} \tag{17}$$

In the context of the load balancing, we use the priority algorithm to select a subset of available containers, to which we apply existing scheduling approaches. In collaboration with our industrial partner, we decided to disregard containers with urgency less than 75% of the maximum urgency container. This ensures that a priority container is scheduled, whilst also balancing the workload over the outfeeds. The DLBA or ADLBA may accommodate the priority extension by calling the priority algorithm (Figure 6) when they start searching for a suitable container for infeed i.

4.2 Delayability

So far, we implicitly assumed that every arriving container joins the queue of waiting containers that are announced to the dispatcher, who decides which container to unload

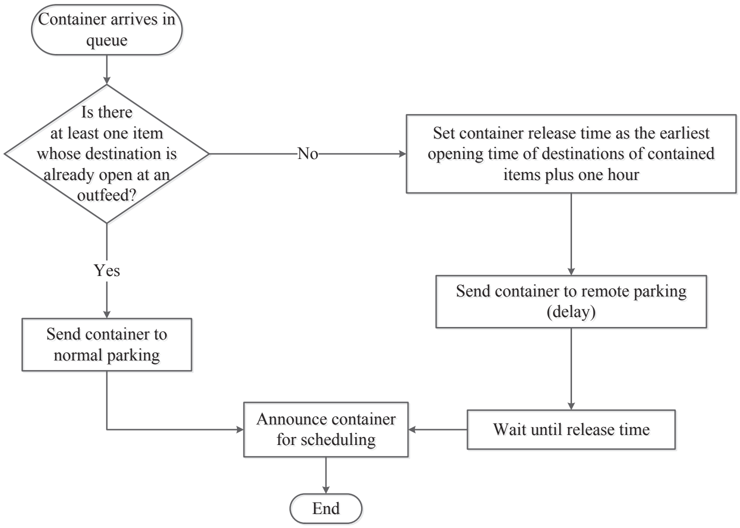


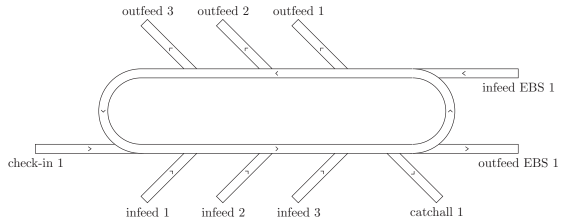
Figure 7: Flowchart for Delayable Scheduling

at which infeed. However, it is possible to temporarily *delay* specific containers and not announce them to the dispatcher. This can be advantageous for two reasons: First, many airports lack infeed capacity during peak hours (usually workdays between 6am – 9am). To reduce these peaks, we may temporarily park ULDs on a remote ULD yard. If ULDs that contain only early baggage are delayed, 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, early baggage items are now stored in relatively expensive EBS systems. Storing them in a container on the yard is a much cheaper solution. The decision to delay a container is made upon arrival and is fairly simple. However, to decide when to bring a container back to the dispatcher, we consider the fact that in the baggage handling industry, each outfeed is assigned to a destination for approximately three hours. Therefore, according to experts opinion, we propose to make a container available an hour after the destination of one of the contained items is assigned to an outfeed, leaving two hours to sort the item(s) that triggered our decision. The algorithm to delay containers is executed as soon as a container arrives in the queue (Figure 7).

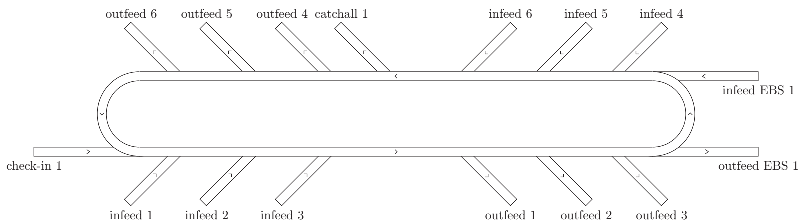
5 Computational Studies

5.1 Experimental setup

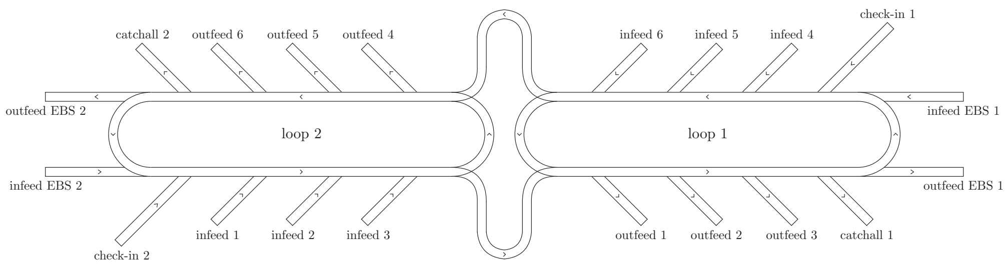
For our experiments, we test the performance of four algorithms: First Come First Served (FCFS) as a common current practice, arbitrary scheduling (ARB) merely as an academic benchmark, the DLBA, and the ADLBA. We use the Applied Materials® AutoMOD™ software package, to apply the scheduling approaches on sorter systems. Based on layouts that are frequently delivered by our industrial partner, we developed three simulation models of sorter systems with simple traffic control rules implemented to conform to a



(a) Sorter Layout 110



(b) Sorter Layout 120



(c) Sorter Layout 22c

Figure 8: Layouts of Three Test Models

realistic situation. However, we do not invest further in the control logic of sorter systems, as this study is concerned with inbound operations scheduling and not the system itself. The simulation models we use are as follows:

- A single sorter in loop configuration with one infeed and one outfeed area, each consists of three conveyors, one infeed for check-in baggage, and one EBS (Figure 8a).
- A single sorter in loop configuration with two infeed and two outfeed areas, each consists of three conveyors, one infeed for check-in baggage, and one EBS (Figure 8b).
- 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. Crossovers, with limited capacity, connect the two sorters (Figure 8c).

We use a tuple notation to identify layouts: number of loops, number of infeed and outfeed areas, special transport routes. 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. Table 3 provides an overview of the transport capacities of the simulation models. The *catchall* outfeed collects items that cannot be sorted due to, e.g., a missed flight.

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 3: Capacities per Simulation Model

Regarding datasets, we distinguish both industries based on their specific characteristics, e.g., the presence or absence of check-in flows or the size of the containers. A second distinction is based on the quantites of items going to certain destinations inside one container. A homogeneous distribution means that inside one container, the number of items for a specific destination is nearly the same for all destinations. A heterogeneous distribution means that some containers hold significantly more items for destination a and others hold more items for destination b. Based on these classifications, four scenarios are constructed:

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

Table 4 provides an overview of the selected values for the scenario parameters. Note that we generate unrealistically high loads on baggage handling sorters, which can occur only in high peak hours. We do so to better test the impact under hard operational conditions. For our simulation model setup, i.e., the size of confidence interval and number of replications for each simulation experiment, we follow the *sequential procedure* for terminating simulations, proposed by Law and Kelton (2000).

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. Airports are mainly interested in one aspect of baggage handling systems: the number of bags that do not catch their flight as a result of a failing sorter system, also known as *missorted bags*. In parcel & postal, however, focus is on *throughput*. Throughput is generally defined as the number of correctly sorted parcels per hour. In addition to the described KPIs, we report on other performance indicators (PIs) that are of interest. Table 5 provides a full overview of the (key) performance indicators

		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 22*c*

Table 4: Scenario Parameters per Simulation Model

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 5: (Key) Performance Indicators

and their measurement units. Furthermore, it shows the relevance of the performance indicators for each of the industries.

5.2 Results

We distinguish between statistical significance and operational significance when comparing results of different algorithms. A difference is statistically significant, when we can statistically prove it exists with 95% confidence. However, a statistically significant measure, may not be relevant from an operational perspective, e.g., a statistically significant difference of 1 item per hour (iph) on throughput in a system sorting thousands of iph is operationally unimportant. In this research, we say that a difference is 'operationally significant' when it is both statistically significant and large enough to be of interest from an operational point of view.

In the boxplots used to show results, the central rectangle spans the first quartile to the third quartile. The segment inside the rectangle shows the median and the two 'whiskers' indicate the limits of the 95% confidence interval. Finally, '+' symbols indicate outliers.

5.2.1 Parcel & Postal

For the evenly distributed scenario in parcel & postal, the simulation studies show that for model 110 and 120 there is no statistical difference between any of the scheduling approaches (Figures 9a and 9b). However, Figure 9c shows that for model 22c the ADLBA approach outperforms all others, with a throughput that is approximately 25 items per hour (1.5%) higher. The original DLBA performs statistically just as well as FCFS and ARB and is therefore not considered to be an improvement. A possible explanation to this behaviour is related to the infeed assignment problem. FCFS, ARB, and DLBA assign a container to an available infeed, irrespective of its location on the sorter system, and when containers are homogeneous there is nothing to optimize or to balance for an approach like the DLBA. However, the ADLBA assigns the containers to infeeds that are selected based on the load balancing criterion. Therefore, although containers look similar, the ADLBA still tries to balance the workload over the two separate sorters, and over time, giving more room for improvement.

For the unevenly distributed parcel & postal scenario, the simulation studies show that the load balancing algorithms (DLBA and ADLBA) do indeed outperform the FCFS and ARB for all the simulation models (Figure 10). Particularly, the DLBA proves to be an interesting approach, as 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. As now containers are differentiable, the original DLBA proves to be a better scheduling approach than the newly developed ADLBA for all models, although only 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 simulation studies also show that some interesting results can be achieved regarding the waiting containers. 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 models 110, 120 and 22*c* respectively. The results of the latter two models, in particular, suggest 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. Specifically, 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.

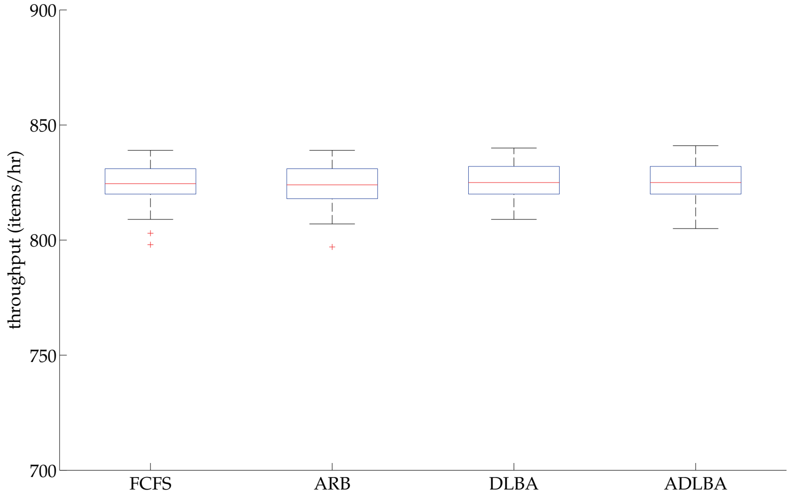
5.2.2 Baggage Handling

In baggage handling we also test the extensions of the algorithms, where we use the suffixes 'p' and 'd' to indicate the priority and delayability extensions respectively in Figures 11 and 12 and the discussion below.

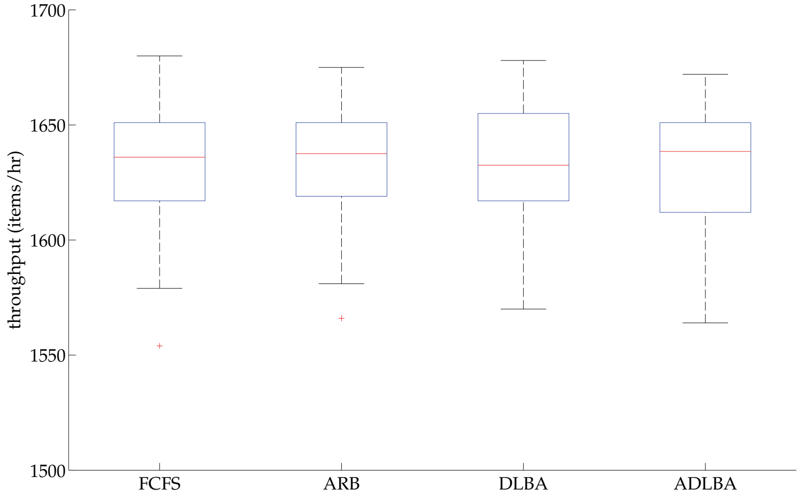
There are main points that make results here incomparable to parcel & postal: First, the contents of the containers differ in destinations of the items, the number of items contained, and more important in priority (for baggage handling containers). Second, focus is now on missort rate, which is a different KPI. Third, in a parcel sorter system, the impact of assignment decisions are directly realized because parcels are sorted immediately to outfeeds, but in baggage handling there is the storage function and flights schedules that heavily influences the flow.

For the evenly distributed baggage handling problem, the ADLBA approach is preferred in model 110, as it outperforms DLBA by 0.4 permillage point (7.0%). In model 120, FCFS is the best approach: it outperforms DLBA by 1.7 permillage point (2.5%). Finally, in model 22*c*, 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. The explanation used for the parcel & postal industry might be applicable in this case as well. The ADLBA appears to bring some benefit for problems where there is no clear differentiations in the data. It is, however, important to realise that the differences in model 110 and 120 are hardly operationally significant. In other words, although the ADLBA and FCFS provide better results for these models, it is not worth the effort to switch approaches.

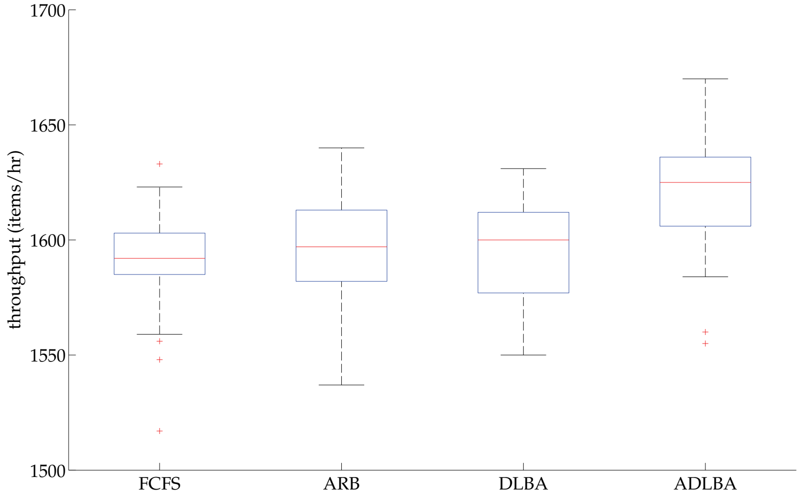
The priority extension improves the results of the DLBA and ADLBA approach. The differences between approaches with and without the 'p'-extension are often only statistical



(a) model 110

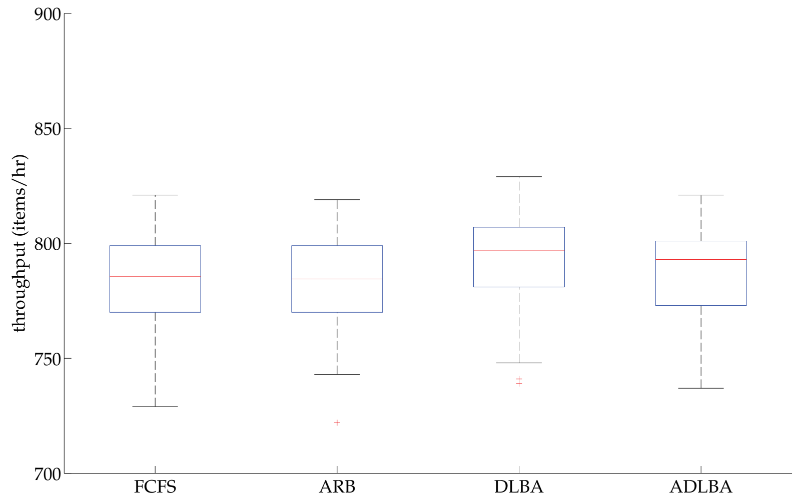


(b) model 120

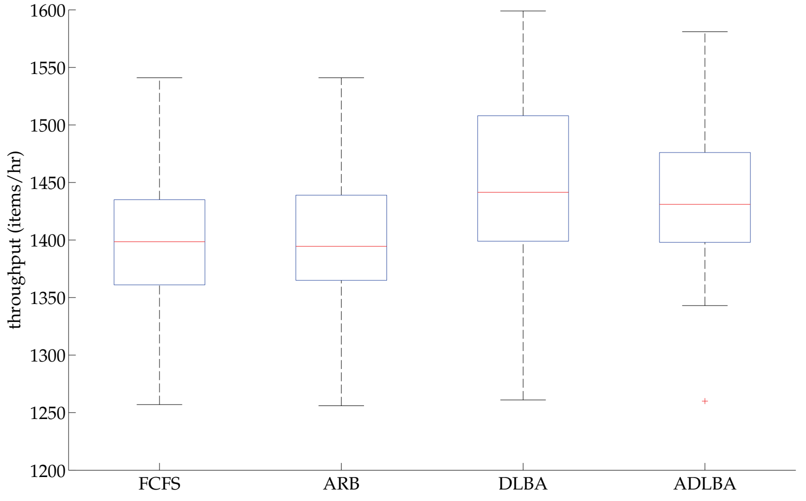


(c) model 22c

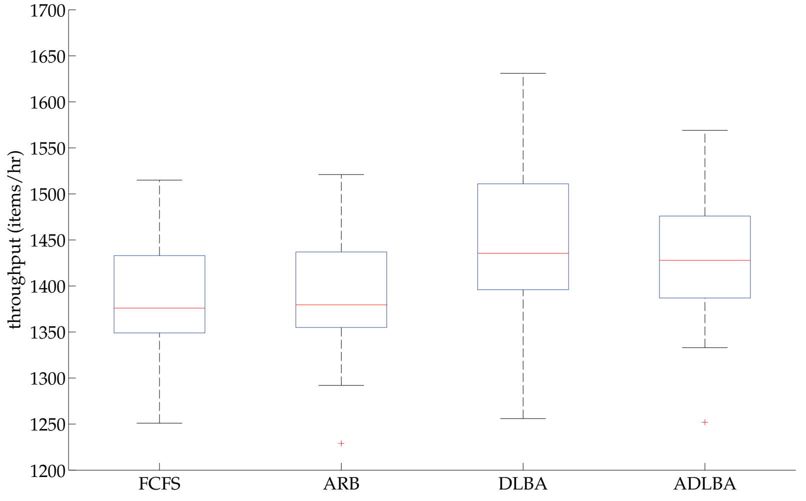
Figure 9: Throughput using the PP-even Scenario



(a) model 110

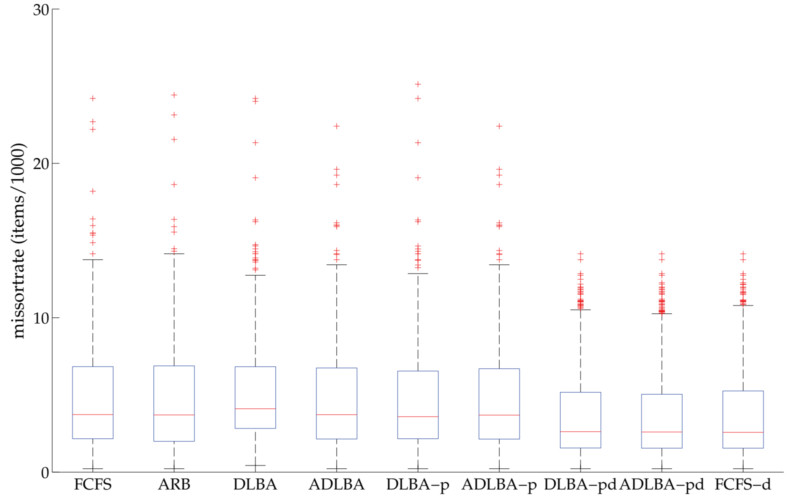


(b) model 120

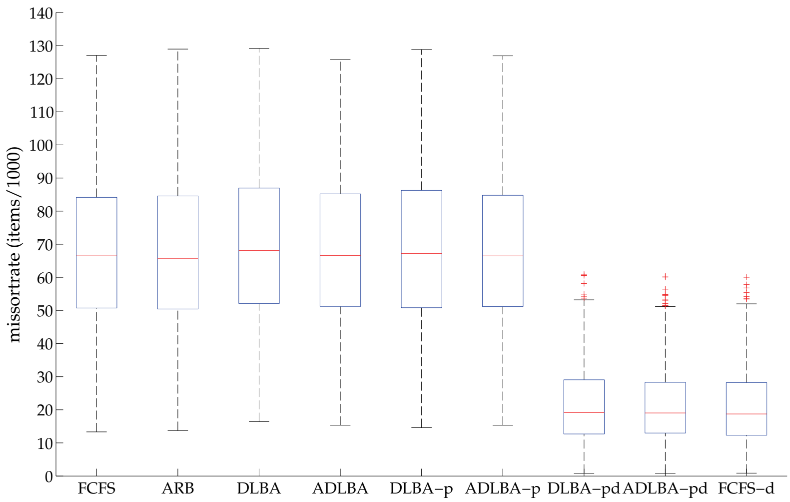


(c) model 22c

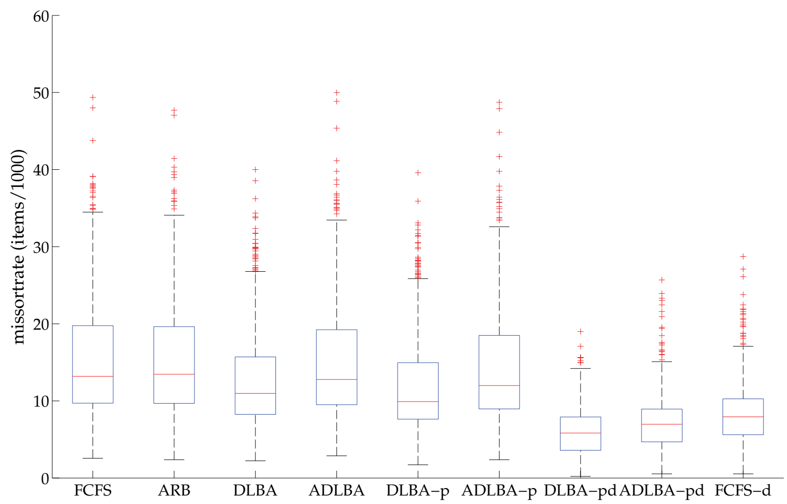
Figure 10: Throughput using the PP-uneven Scenario



(a) model 110



(b) model 120



(c) model 22c

Figure 11: Missort rate using the BHS-even Scenario

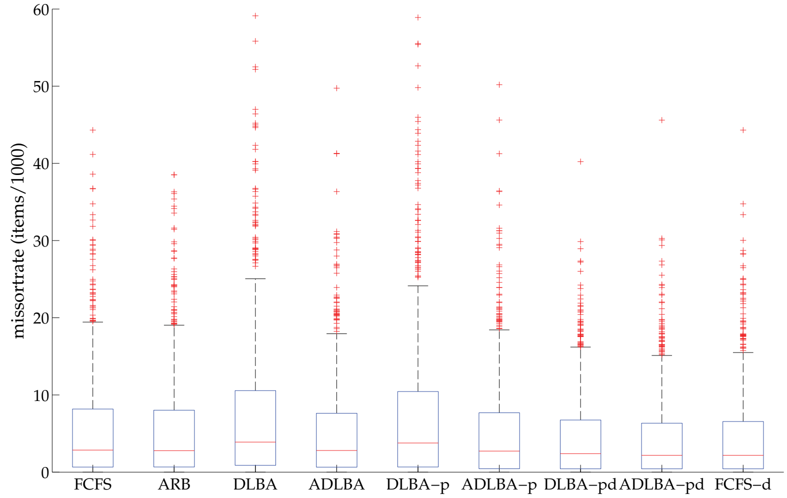
and not operational significant though. On the other hand, the ‘d’-extension tremendously improves the performance of the sorter systems. Generally speaking, this extension reduces the missort rate by 1.0, 47.6, and 7.3 permillage point (20.8, 68.7, and 50.1%) on model 110, 120, and 22*c* respectively, 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 caused by a relatively high workload (70% of outfeed capacity), which we used on purpose.

The ‘d’-extension also affects the other PIs. For instance, it reduces the required EBS space by 96, 230, and 204 items (approximately 30% in all cases) for models 110, 120, and 22*c* 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 22*c* to the remote parking.

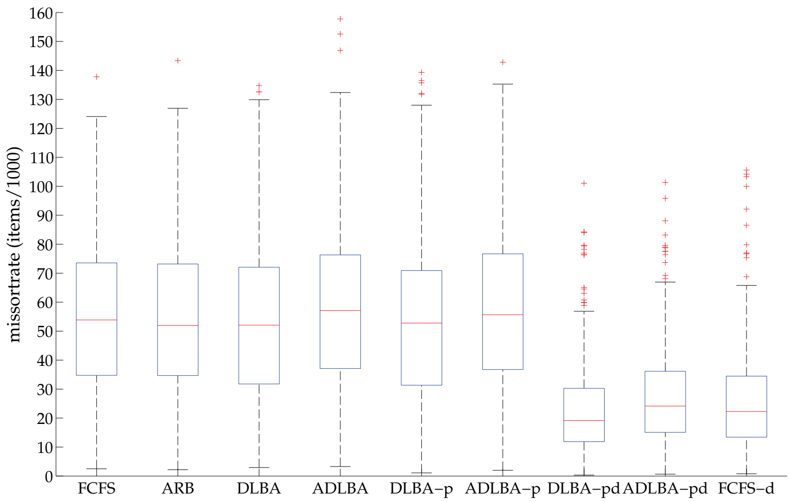
The analysis suggests that applying the ‘d’-extension in combination with FCFS might provide results that are comparable to the more complicated scheduling techniques. The simulation studies show that this is indeed the case for model 110 and 120. In model 22*c*, however, the DLBA-pd outperforms FCFS-d 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 more evident. For models 120 and 22*c* 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 operationally by FCFS. All other PIs show no operationally significant difference, except in model 22*c*. 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. We observe that the realizations of internal transport times are not in line with the estimations made by the ADLBA, especially for larger baggage handling systems. The highly stochastic and dynamic environment, in addition to the occasionally used storage function, seem to make the estimations less reliable.

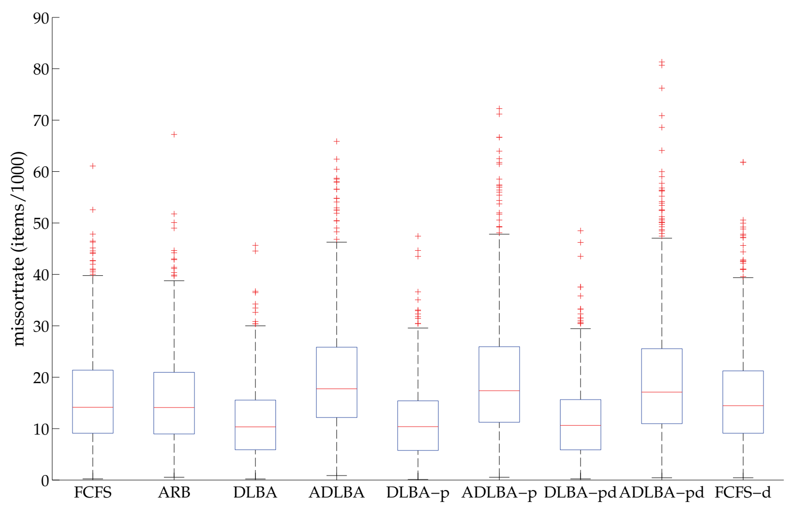
Including the ‘p’-extension does not bring significant improvements. However, the ‘d’-extension has a positive effect on the performance of the scheduling approaches. In model 110 the ADLBA-pd is clearly the best approach, and outperforms FCFS by 1.7 permillage point (19.6%). In model 120 and 22*c* the 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 PIs are affected by the delayability extension. Even with FCFS the delayability extension 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. The DLBA-pd is the best approach for models 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.



(a) model 110



(b) model 120



(c) model 22c

Figure 12: Missort rate using the BHS-uneven Scenario

6 Conclusion

The results show that the newly developed ADLBA is only interesting for parcel & 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. However, in relatively simple sorter systems, the ADLBA is not likely to contribute to the performance, probably because in such systems internal transport times are also similar. As soon as containers become more differentiable, the original DLBA outperforms not only the current practice FCFS, but also the newly developed ADLBA in all simulation models.

For both baggage handling scenarios the results show that the workload balancing approaches DLBA and ADLBA do indeed improve the performance of sorter systems. Again the ADLBA is the preferred solution when the differences between data instances 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. heterogeneous containers in larger and more complicated sorter systems. This makes us recommend the original DLBA as we adapted it for baggage handling sorter systems, because in this industry containers are more likely to be heterogeneous, and the need for smarter scheduling approaches is for the more complex systems, i.e., at air hubs. In general, the ADLBA uses an approach that is too detailed in a highly stochastic environment, and so when the containers are heterogeneous the DLBA performs well while the ADLBA only *overschedules* the problem, at least in the layouts we tested. Therefore, we find it wise not to schedule inbound containers based on detailed modeling of sorter systems (e.g., ADLBA), but to apply simpler approaches (e.g., DLBA) and invest more in the control rules and algorithms of the sorter system itself (and the EBS), which will be part of our future research.

Actually, the major improvements are achieved by the delayability extension we developed. The effects of the priority extension are often operationally insignificant, but the delayability extension shows that impressive improvements on all PIs are possible. Hence, we recommend applying the delayability extension in practice, it is interesting that it is applicable as an add-on to current scheduling tools, since we were able to get significant improvements from implementing delayability even with the FCFS approach.

References

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

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.

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.

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.

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.

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.

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.

Working Papers Beta 2009 - 2012

nr.	Year	Title	Author(s)
376	2012	Improving the performance of sorter systems by scheduling inbound containers	K. Fikse, S.W.A. Haneyah, J.M.J. Schutten
375	2012	Strategies for dynamic appointment making by container terminals	Albert Douma, Martijn Mes
374	2012	MyPHRMachines: Lifelong Personal Health Records in the Cloud	Pieter van Gorp, Marco Comuzzi
373	2012	Service differentiation in spare parts supply through dedicated stocks	E.M. Alvarez, M.C. van der Heijden, W.H.M. Zijm
372	2012	Spare parts inventory pooling: how to share the benefits	Frank Karsten, Rob Basten
371	2012	Condition based spare parts supply	X.Lin, R.J.I. Basten, A.A. Kranenburg, G.J. van Houtum
370	2012	Using Simulation to Assess the Opportunities of Dynamic Waste Collection	Martijn Mes
369	2012	Aggregate overhaul and supply chain planning for rotables	J. Arts, S.D. Flapper, K. Vernooij
368	2012	Operating Room Rescheduling	J.T. van Essen, J.L. Hurink, W. Hartholt, B.J. van den Akker
367	2011	Switching Transport Modes to Meet Voluntary Carbon Emission Targets	Kristel M.R. Hoen, Tarkan Tan, Jan C. Fransoo, Geert-Jan van Houtum
366	2011	On two-echelon inventory systems with Poisson demand and lost sales	Elisa Alvarez, Matthieu van der Heijden
365	2011	Minimizing the Waiting Time for Emergency Surgery	J.T. van Essen, E.W. Hans, J.L. Hurink, A. Oversberg
364	2011	Vehicle Routing Problem with Stochastic Travel Times Including Soft Time Windows and Service Costs	Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok
363	2011	A New Approximate Evaluation Method for Two-Echelon Inventory Systems with Emergency	Erhun Özkan, Geert-Jan van Houtum, Yasemin Serin

		Shipments	
362	2011	Approximating Multi-Objective Time-Dependent Optimization Problems	Said Dabia, El-Ghazali Talbi, Tom Van Woensel, Ton de Kok
361	2011	Branch and Cut and Price for the Time Dependent Vehicle Routing Problem with Time Window	Said Dabia, Stefan Röpke, Tom Van Woensel, Ton de Kok
360	2011	Analysis of an Assemble-to-Order System with Different Review Periods	A.G. Karaarslan, G.P. Kiesmüller, A.G. de Kok
359	2011	Interval Availability Analysis of a Two-Echelon, Multi-Item System	Ahmad Al Hanbali, Matthieu van der Heijden
358	2011	Carbon-Optimal and Carbon-Neutral Supply Chains	Felipe Caro, Charles J. Corbett, Tarkan Tan, Rob Zuidwijk
357	2011	Generic Planning and Control of Automated Material Handling Systems: Practical Requirements Versus Existing Theory	Sameh Haneyah, Henk Zijm, Marco Schutten, Peter Schuur
356	2011	Last time buy decisions for products sold under warranty	M. van der Heijden, B. Iskandar
355	2011	Spatial concentration and location dynamics in logistics: the case of a Dutch provence	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
354	2011	Identification of Employment Concentration Areas	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
353	2011	BOMN 2.0 Execution Semantics Formalized as Graph Rewrite Rules: extended version	Pieter van Gorp, Remco Dijkman
352	2011	Resource pooling and cost allocation among independent service providers	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
351	2011	A Framework for Business Innovation Directions	E. Lüftenegger, S. Angelov, P. Grefen
350	2011	The Road to a Business Process Architecture: An Overview of Approaches and their Use	Remco Dijkman, Irene Vanderfeesten, Hajo A. Reijers
349	2011	Effect of carbon emission regulations on transport mode selection under stochastic demand	K.M.R. Hoen, T. Tan, J.C. Fransoo G.J. van Houtum

348	2011	An improved MIP-based combinatorial approach for a multi-skill workforce scheduling problem	Murat Firat, Cor Hurkens
347	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	Joint optimization of level of repair analysis and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Inventory control with manufacturing lead time flexibility	Ton G. de Kok
344	2011	Analysis of resource pooling games via a new extenstion of the Erlang loss function	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Vehicle refueling with limited resources	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger
342	2011	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Bilge Atasoy, Refik Güllü, TarkanTan
341	2011	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Analysis of a two-echelon inventory system with two supply modes	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
334	2010	Flexible Heuristics Miner (FHM)	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	The Effect of Workload Constraints in Mathematical Programming Models for	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan

		Production Planning	
330	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	Preventing or escaping the suppression mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
319	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Minimal Protocol Adaptors for Interacting Services	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
		Teaching Retail Operations in Business and	

317	2010	Engineering Schools	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
316	2010	Design for Availability: Creating Value for Manufacturers and Customers	Pieter van Gorp, Rik Eshuis.
315	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	Getting trapped in the suppression of exploration: A simulation model	S. Dabia, T. van Woensel, A.G. de Kok
313	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
	2010		
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo

298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
			B. Vermeulen, A.G. de Kok

278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	R. Seguel, R. Eshuis, P. Grefen
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl