Supply chain optimization using an incremental approach: the improved WDScan model

Albert Heijn

UNIVERSITEIT TWENTE.

Robert Alexander Smit

Summary

Albert Heijn is a large supermarket chain in the Netherlands that supplies around 900 stores in the Netherlands from its four distribution centres across the Netherlands. Most of these stores are delivered on a daily basis for each flow of goods. The deliveries of the ambient and fresh goods are scheduled in delivery windows of one hour. In order to gain more insight in the total costs of a delivery time window plan (DTWP) Albert Heijn has developed a model with ORTEC: WDScan. However, this model does not give acceptable results. The DTWP WDScan creates is not acceptable, because (1) it does not imply lower overall costs than the manually generated DTWP and (2) the manual DTWP is considered infeasible by WDScan, while this DTWP is currently used and therefore feasible. Our goal is to improve the WDScan model to make sure that it can be used to generate feasible and improved delivery time window plans for the busy weeks. The busy weeks are chosen, because changing a DTWP in these weeks is accepted by the stakeholders, while in normal weeks it is not accepted to change the DTWP.

In literature, the DTWP problem is related to an Inventory Routing Problem (IRP). An IRP is a generalisation of the well-known vehicle routing problem (VRP), first introduced by Danzig and Ramser (1959) as the Truck Dispatching Problem. The problem most related to the DTWP problem is the periodic inventory routing problem introduced by Gaur and Fisher (2004). Gaur and Fisher (2004) introduce a method to cope with this problem at Albert Heijn. The difference between the periodic inventory routing problem and the DTWP problem is that Gaur and Fisher (2004) solved the actual truck routing while our problem mainly focusses on an estimation of the costs of transportation. In addition to the transportation costs, our problem considers the distribution centre and store costs. The DTWP problem also involves the importance of not bothering customers at the stores of Albert Heijn by restocking. Besides the difficulty of assessing the transportation, store, and distribution costs, the importance of not bothering customers by restocking should be taken into account.

In order to improve the WDScan model, we analyse the input and parts of the WDScan model. We identify three areas for improvement. The first area for improvement is the delivery size estimation. Currently, the values for the delivery size of a delivery to a store are fixed per day, based on historical data. Therefore, the delivery size is the same when the delivery time window is in the morning or in the evening. However, in practice the delivery size is based on the expected number of customers until the next delivery.

The second area for improvement considers the penalty functions used in WDScan. Our analysis shows that the penalty functions of WDScan need improvement. The main reason the penalty functions have to be improved is that they are a large portion of the target function, while this should not be the case for a feasible solution. In addition, the manual feasible DTWP yields a lot of penalty costs, while feasible solutions should not yield high penalty costs.

The third area for improvement is the order picking model. In WDScan, the order picking model is based on just-in-time order picking. The order picking model assumes order picking is an immediate predecessor of loading the order into the truck at the distribution centre. However, in practice orders

are picked in advance. Order picking in advance allows solutions that are infeasible with just-in-time order picking.

We propose three solutions to improve the above mentioned areas for improvement. The first solution is a method to estimate the change in delivery size when changing a *heartbeat moment*. A heartbeat moment is the delivery time window throughout the week. When changing such a heartbeat moment, all of the delivery windows are set at the new heartbeat moment. Based on the amount of customers a delivery supplies, we calculate the amount of goods for each sales hour. When changing the heartbeat moment the amount of goods of the delivery is calculated by the sum of the goods for the hours until the next delivery. We show that this gives a good estimate of the delivery size when changing the heartbeat by testing the method for different stores with similar customer patterns, but different heartbeat windows.

The second solution is the improvement of the penalty functions of WDScan. We propose changing the penalty functions based on the guidelines by Smith and Coit (1997). Smith and Coit (1997) argue that penalty functions should be severe enough to penalize infeasible solutions, but not too high to not allow infeasible solutions that are close to the optimal region. We introduce upper bounds for the penalty functions that are severe enough to penalize infeasible solutions.

The third solution involves changing the just-in-time order picking model in WDScan. We propose to use a method that supports order picking in advance. This method is based on the method currently used by the SCCP department of Albert Heijn to assess whether the manual DTWP is executable in the distribution centres. The main assumption is that if the orders can be picked in advance by a shift of order pickers and does not surpass the maximum loading dock buffer area, a valid order picking plan can be created. Our method can determine the minimum costs and use of buffer capacity given the model assumptions.

In addition to our three improvements, we recommend additional steps to improve WDScan. The first step we advise Albert Heijn to take after implementing our improvements is to verify the changes to a DTWP. We propose to determine the difference in costs of a DTWP before and after a periodic DTWP change. The second step that we propose is that Albert Heijn assesses the penalty values based on our upper bounds. The last additional improvement is determining how much not bothering a customer is worth to Albert Heijn. Determining how much not bothering a customer is worth to Albert Heijn is required to assign the weights of the real costs on the one hand and the customer bothering component on the other hand. Future research includes the bothering of customers model and how WDScan copes with Sundays.

Preface

This thesis is the final stage of my master Industrial Engineering and Management, track Production and Logistics Management. I worked with great pleasure over the last six months at Albert Heijn. I would like to thank the whole Supply Chain Capacity Planning department for the pleasant time working there. Especially, I would like to thank Adze Spoelstra and Martijn Beerepoot for their feedback, guidance and support. I would also like to thank Liesbeth Brederode and René Baks for giving me the opportunity to conduct this research project at Albert Heijn.

In addition to my supervisors at Albert Heijn, I would like to thank Marco Schutten and Leo van der Wegen for their feedback and counselling. You have made sure the level of this thesis has been greatly improved.

Finally, I would like to thank my friends and family for their unconditional support during my studies. You have given me the strength and courage to finish this project.

Robert Smit

31st of august 2014

Contents

List of Figures

List of Tables

Glossary

AH (Albert Heijn B.V., a subsidiary of Ahold N.V.): the company this research is conducted at.

BW (Busy Weeks): the busy weeks are for example the weeks before Christmas or Easter where a lot more goods have to be delivered to the stores.

Current DTWP: the manually constructed current DTWP. The current DTWP is often used in this research as a benchmark for the results of WDScan.

DC (Distribution Centre): one of the regional distribution centres of AH.

DTW (Delivery Time Window): the time window a delivery to a store for one of the flows (ambient or fresh) is made in. The DTWs of all of the stores of a distribution centre is called a Delivery Time Window Plan for this distribution centre.

DTWP (Delivery Time Window Plan): the plan of the delivery time windows for all of the stores connected to one distribution centre. This DTWP gives the delivery times for all of the stores and days of the week. The delivery time window is at the heartbeat moment, the same moment for every day. In addition, if we speak of a DTWP of a store we mean the part of the DTWP considering this single store.

Franchise: the term used to indicate that a store is not owned by Albert Heijn but rather is owned by an entrepreneur who is the franchisee of the Albert Heijn brand.

Heartbeat moment: the timeslot a store is delivered. This timeslot is the same every day of the week, if a delivery is made to this store on this day.

Infeasible DC Closed Loading: a penalty for each trip that leaves the Distribution Centre (DC) outside of the opening hours of the Distribution Centre loading time of the DC

Infeasible DC Closed Production: a penalty for each trip that has its order picking conducted outside of the order picking opening hours of the DC

Infeasible DC Prod Spread: a penalty per roll cage (RC) outside of the given bandwidth for a set amount of hours (parameter). The average over these hours is calculated and all the RCs outside of the given bandwidth result in a penalty per RC. The penalties are calculated for all of the flows, and the sum over all of the flows.

Infeasible DC Trip Departure: a penalty when less than a given threshold value or more than the capacity of the DC amount of trucks depart from the DC for all the hours a trip is made. The amount of trips can be a non-integer.

Infeasible Store Closed: a penalty for if the start restocking is after the store is closed. Per store a parameter for the time after closing restocking is allowed.

Infeasible Store Interval Time: a penalty if two deliveries to the same store are too close to one another. For two deliveries of the same flow, four hours is required between to deliveries. For two deliveries that are not of the same flow, two hours is required. Violating this restriction gives a penalty per minute violation.

Infeasible Truck Type Spread: a penalty per truck outside of the given bandwidth for a set amount of hours (parameter). The average over these hours is calculated and all the trucks outside of the given bandwidth result in a penalty per truck. The penalties are calculated for all of the truck types, and the sum over all of the trucks.

Infeasible Truck volume: a penalty for using more than the capacity of a truck per RC. Per RC over the truck capacity a penalty is inflicted.

RCs (Rolcontainers, Dutch): load carriers used in the process of transport of goods between the distribution centres and the stores.

WD (Winkeldistributie, Dutch): the project initiated to improve the delivery time window plan (DTWP) for the stores, distribution centres, transportation and customers in the store. One of the outcomes of this project is the WDScan model.

WWM: The stores owned by Albert Heijn. Contrary to the franchise stores that are not owned by Albert Heijn but are part of the Albert Heijn brand.

1. Introduction

This chapter introduces the research project. This thesis is the final part of my master program in Industrial Engineering and Management, conducted at the Supply Chain Capacity Planning (SCCP) Department of Albert Heijn (AH), introduced in Section 1.1. In addition, Section 1.1 presents the motivation for the research. Section 1.2 gives the problem definition based on the initial problems as stated by Albert Heijn. Section 1.3 presents the research goal and the sub questions. This chapter finishes with the outline of this thesis in Section 1.4.

1.1 Introduction to Albert Heijn and the SCCP Department

Albert Heijn (AH) is a subsidiary of Ahold and is situated in the Netherlands. Its headquarters are located in Zaandam and it supplies around 900 stores in the Netherlands and a few stores in Belgium, from different distribution centres across the Netherlands. The deliveries are made on a daily basis and for different flows of products. An example of such a flow is the flow that contains the fresh (cooled) goods, which are delivered separately from the ambient goods.

The delivery time windows for the stores are based on past practice, and revised occasionally if for example a new store has to be added or a delivery time window of an existing store has to be modified. The creation and modification of the delivery time window plan (DTWP) is a lengthy process that involves a lot of communication with the store managers, the operational managers and the supply chain partners. If a change in such a window is approved, a process of trial and error in changing time slots has to result in a feasible solution. Albert Heijn would like to use a better solution and make the process of determining these new delivery windows less time consuming.

In the current situation these delivery times are used as input for the transportation department of Albert Heijn that creates the vehicle routes based on these delivery windows. Based on this transportation plan, the distribution centres create their capacity plan. This process is based on optimizing small parts of the total distribution chain. In order to get a better overall output, Albert Heijn would like to have the delivery windows determined based on costs in the stores, the amount of customers that are bothered by the restocking of the shelves in the stores, the costs in the distribution centres and the total transportation costs. Taking all of these criteria into account, Albert Heijnhopes to get a better overall performance.

In the process of trying to improve the timeslots, Albert Heijn and ORTEC have developed a model that should take into account the above mentioned criteria. This model is known as WDScan and creates delivery windows based on minimizing the total costs and amount of customers bothered by restocking. However, the delivery windows created by this model are difficult to implement, because just a small change in the current delivery window is accepted by the store managers of Albert Heijn. In the weeks that there is more pressure on the supply chain (‘uitvalsweken’ in Dutch), it is accepted by the stores that delivery times change. Examples of these weeks are the weeks before Christmas and Easter, where a lot more goods have to be delivered to the stores. In these weeks changes are accepted, because it is necessary to change the delivery times to allow the larger volumes to go through the supply chain.

The main problem as indicated by Albert Heijn is the bothering of customers when restocking the shelves during peak times in the stores. The goal is to reduce the amount of customers bothered by restocking, without implying additional costs in the total value chain. This includes the store, transportation and DC costs. The project “Winkeldistributie” (WD) has tried to facilitate this; however because of a lack of support from different departments among Albert Heijn this has not reached its potential. The future state desired by Albert Heijn is a centralised approach aimed at reducing overall costs, while creating additional value for their customers.

The problem WD has to solve is a typical multi-disciplinary problem, as it involves a lot of different actors. In this case the impact of a change in the delivery windows involves a lot of different departments of Albert Heijn. These departments all optimize their plan in a certain way. The project WD has tried to *optimize*[1] the total value chain costs, instead of optimizing the costs for each department separately.

The assignment is to see how we can improve the delivery times created for these more busy weeks where changes are permitted. The current situation is that the DTWP is created ‘by hand’, building the DTWP from scratch in Excel. Albert Heijn wants to see if this can be improved by using WDScan to determine the delivery windows. However, the WDScan application is not creating acceptable delivery times yet, and a part of the solution is to make changes to the model. If there is a way to get these busy weeks improved, Albert Heijn likes to see how this can be used for the normal weeks. However, to assess how WDScan can be used for the normal weeks is outside the scope of this research.

1.2 Problem definition

This section describes the problem identification and analysis. Identifying the problem is done based on interviews with supply chain officers of the SCCP Department of Albert Heijn. The initial goal of project WD was to reduce the amount of customers bothered by restocking and optimizing the sum of transportation, DC and store costs. The DC costs are the sum of the costs of distribution centres of Albert Heijn in the Netherlands.

The problems as stated by Albert Heijn are (Beerepoot & Spoelstra, Interview 1, 2014):

- Customers are bothered by restocking during customer peak time
- It is not possible to gain insight in the total costs of a DTWP and customers bothered
- The process of determining the delivery time window plan (DTWP) is a manual process
- A lot of departments and stores (stakeholders) all have a say in the process

The main issue in having a manual process and the urge to reduce the amount of customers bothered by restocking, is evaluating what effect a change to the DTWP has. In order to reduce the amount of

[1] *Within Albert Heijn the term ‘optimize’ is used for improving, rather than getting to an optimal value.*

customers bothered and total costs, we have to know what costs a change implies. In the current situation it is not known in advance what a change inflicts. In addition, the way the DTWP changes are evaluated should be accepted by all of the stakeholders that are affected by such a change.

As described in Section 1.1, Albert Heijn has tried to facilitate this with the project WD. However, this project finished without a proper implementation of a model to facilitate this for the SCCP department. This year a new project 'WDScan' was initiated to implement the model built during the project WD at the SCCP department. The main question posed by the SCCP department is:

'How can WDScan be used to improve the delivery time window plan process?'

Section 1.3 rephrases this question to a research goal. The SCCP department thinks it is best to use WDScan for the busy weeks (BW), because they do not have to take all of the stakeholders into account when creating the DTWP for these weeks. We do not have to take all the stakeholders into account, because the stakeholders accept changes in the DTWP for the BWs. Contrary to the regular weeks, weeks where the stakeholders only accept minor changes to the DTWP. In addition, Section 1.3 explains what research questions we use to reach the research goal.

1.3 Research goal and sub questions

The research goal of this thesis, as explained in Section 1.2, is based on the question of the SCCP department to see how we can use WDScan to improve the DTWP process. At the SCCP department there is reasonable doubt about the outcomes of the model and they would like to gain more insight in how the model works and what parameters influence the outcome of the model most. The research goal is therefore:

"Provide a plan to use WDScan to make better decisions on store, distribution centre and transportation costs versus the amount of customers bothered for the busy weeks"

This research goal implies we improve the current situation into a more desired situation. To achieve this goal we have created several sub questions. In order to change the current situation to the desired situation, we have to investigate the current situation. Therefore our first research question is: *'How are the time windows determined in the current situation?'* Once we know the current situation, we can find improvements to the current situation. Our second research question investigates the model built by ORTEC and its possible drawbacks and benefits. Therefore, our second research question is: *'What does the WDScan method look like, and what are its benefits and drawbacks'*

Based on the current situation and the WDScan model, we assess what literature is available on these subjects. We focus on two areas: the problems in literature that are comparable to our problem and the techniques that are available in literature to solve these problems. The study includes literature about parts of the WDScan model. In order to evaluate the parts of the model of WDScan, we need literature to assess the validity of the choices made. Therefore, our third research question is: *'What are similar problems to the WDScan problem in literature, and what are possible solutions for these problems in literature?'*

After assessing the WDScan model, we give suggestions to improve parts of the model. The suggestions are based on the literature review and interviews with experts of the SCCP department. As stated in Section 1.1, Albert Heijn wants to use the improved model for the busy weeks (BW). The reason to investigate the BWs is that the SCCP department thinks implementing WDScan for these weeks is easier, because in the BWs the stakeholders accept changes to the DTWP. For example, the store managers of Albert Heijn accept that their delivery times change in these weeks. However, in order to use the model for these weeks we have to assess if the method is able to cope with the busy weeks. The research question is therefore as follows. *'How can we improve the WDScan model and what is the performance of the improved model?'*

Summarised, we have formulated the following research questions:

1. How are the time windows determined in the current situation?
2. What does the WDScan method look like, and what are its benefits and drawbacks
3. What are similar problems to the WDScan problem in literature, and what are possible solutions for this problem in literature?
4. How can we improve the WDScan model and what is the performance of the improved model?

1.4 Outline

The research questions mentioned in Section 1.3 are the guidelines for the chapters in this thesis. Hence, the thesis first explains the problem and the background of the problem, and then continues to illustrate the current situation and the solution proposed by ORTEC. The following chapters analyse the WDScan model, suggest improvements and test these improvements. The final part of the thesis considers conclusions and future research.

Chapter 2 concentrates on explaining the current situation and the solution proposed by ORTEC: WDScan. By explaining the current situation and demonstrating how the WDScan model works we lay the base for the analysis of the WDScan model. This chapter illuminates several drawbacks and advantages of the WDScan model. In addition, this chapter selects items to improve to the WDScan model. Chapter 3 answers the third research question, by assessing what literature is available to tackle the problem WDScan tries to solve. To do this, we first have to classify this problem. This clarifies what solutions are readily available in literature and what the gaps are in the available literature. In addition, it presents literature on parts of WDScan that require further investigation.

Chapter 4 focusses on the improvements that we advise for the WDScan model. This chapter presents improvements to the ORTEC model based on the drawbacks of the model as explained in Chapter 2. The chapter continues by giving the validation of the improvements. Here, quantitative and qualitative validation shows why the improved methods should be implemented. In addition, additional steps that have to be taken by Albert Heijn to use the WDScan method for the busy weeks are presented.

This thesis finishes with the conclusions in Chapter 6. In addition to the conclusions, Chapter 6 elaborates on future research.

2. Current situation and WDScan model

This chapter answers the first two research questions: '*How are the time windows determined in the current situation*' and *'What does the WDScan method look like, and what are the benefits and drawbacks'*. To answer these questions, Section 2.1 first illustrates the manual process of determining the delivery time window plan (DTWP) for the busy weeks (BW). It describes the process and gives a schematic overview of the process under investigation. Section 2.2 focusses WDScan, the solution proposed by ORTEC to solve this problem. This section is divided in the input, model choices and output of WDScan. Section 2.3 analyses WDScan and identifies what the advantages and drawbacks are of WDScan, and presents the selection of the parts of WDScan to improve. Finally, Section 2.3.3 gives the conclusions of this chapter.

2.1 Manual process of determining the delivery time window plan

To answer the first research question: *'How are the time windows determined in the current situation?',* this section explains the current manual process of determining the delivery time window plan (DTWP). This process is a manual process and starts 20 weeks before the busy weeks (BW). Based on the most recent DTWP of a regular week, indexed for the forecasted volume increase, a tactical plan is created. Based on the forecast and the regular DTWP, the supply chain specialists determine what plan should be used as a base for each of the days in the busy weeks. The amount of goods of these deliveries is adjusted for the volume increase expected by Albert Heijn for these days. Because the amount of goods that has to be delivered to the stores is higher than in a regular week, the regular allocation of trucks is not valid anymore. To assess whether more deliveries to a store are necessary, the amount of load carriers (RCs) for a delivery to a store is compared to the RC capacity of a truck. If the amount of RCs plus a safety margin is larger than the truck capacity, an additional delivery to the store is scheduled. Additional deliveries are scheduled, because the truck scheduling algorithm, Shortrec, does not generate additional deliveries. The truck scheduling algorithm solves a Vehicle Routing Problem (VRP) with time windows (Sections 3.2. and 3.3 explain the VRP in more detail).

The additional delivery to a store is preferably placed on the heartbeat moment of this store, a time window equal to the delivery rhythm of this store throughout the week. If the first step is completed for all of the deliveries to all of the stores, the next step is initiated. This next step is called 'simulation'; here a rough transportation plan based on the DTWP is created. The transportation plan shows the routes of the trucks and what stores are combined in a truck. In addition, this transportation plan visualises how many trucks have to leave the Distribution Centre (DC) each fifteen minutes. Based on the amount of trucks that leave the DC each fifteen minutes, the SCCP department determines whether the DTWP is 'executable' for the four DCs. The following points are evaluated, and explained in more detail in the remainder of this section:

- Is enough buffer space available to gather the orders that have to leave in a certain hour?
- Is it possible to order pick the amount of goods before the truck has to leave the DC?
- Is the amount that has to be order picked evenly spread over the hours?

- Is the percentage of goods that has to be picked at night below a certain (predetermined) threshold value?

Based on the above mentioned points the DTWP is evaluated. If one of these restrictions is not met, the SCCP department is requested to make changes to the DTWP. Typical changes are moves of a certain amount of volume to the previous or next hour. Based on these requests a number of delivery time windows are rescheduled. Figure 1 illustrates how changing these time windows can help to ensure that the amount that has to be loaded is within the maximum capacity. The dashed line in Figure 1 shows the order picking amounts per hour that had to be picked before changing the DTWP. The dotted line shows the amount that has to be picked after the changes have been made. Where the dashed line surpasses the solid capacity line more than once, the dotted line is completely below the maximum capacity line.

Figure 1: Order picking amount, adapted from Albert Heijn (2014)

In addition to the order picking amount, the usage of the buffer area has to be below the capacity threshold. The buffer area is the area where the RCs are waiting before they are loaded into the trucks. In front of each loading area there is a buffer area that can buffer at most two truckloads. Figure 2 illustrates the total buffer usage, before the changes to the DTWP and after the changes. The dashed line, the buffer usage of the base DTWP, surpasses the maximum of the buffer several times. While the dotted line, the buffer usage of the new DTWP, never surpasses the maximum buffer capacity line.

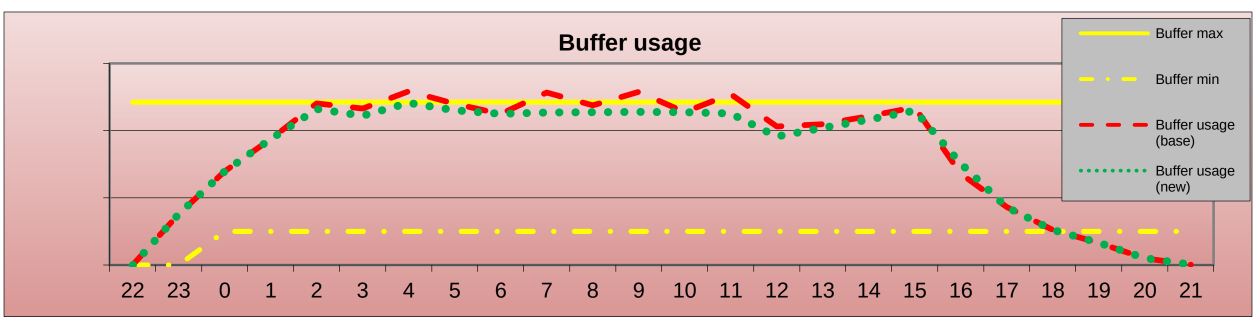

Figure 2: Buffer usage, adapted from Albert Heijn (2014)

Figures 1 and 2 are based on the transportation plan created from the DTWP. This transportation plan is created using Shortrec, a software package supplied by ORTEC. Based on the departure times of trips from the DCs of Albert Heijn the amount of goods that has to be loaded is calculated. The dotted green lines in Figure 1 and 2 show the improved DTWP, while the red dashed line shows the starting point of the improvement. After changing the delivery times of different stores, another rough transport plan is generated. Because the transport algorithm only does two iterations, while it would do at least twelve to create the final plan, the result is called a rough transportation plan. The reason for doing another iteration is that the route planning algorithm has different delivery time windows for a significant amount of stores. Different delivery time windows could result in a different amount of goods that has to be picked compared to the first plan. In addition, doing a second run is a check to see if the changes that are requested by the DCs are honoured and to filter possible errors made in the manual process. Doing more than two runs would not increase the performance substantially, because most of the issues are resolved after one run (Beerepoot & Spoelstra, Interview 1, 2014). This process is done for all of the days that have a high volume (as indicated in the tactical plan), for all of the DCs. The process of building the DTWP, running all of the individual high volume days with Shortrec for all distribution four distribution centres, updating the DTWP and run the new DTWP in Shortrec takes two fulltime weeks of four employees for the busy weeks around Easter.

This method makes sure that the goods flow through the supply chain without hiccups. However, this does not take into account the total transportation, DC and store costs or the customers that are bothered by restocking in the stores. The ‘simulation’, the process of improving the amount of goods that has to be loaded as explained in Figure 1, improves the DTWP. This makes sure the DC is able to order pick the amount of goods requested by the stores. In this ‘simulation’ choices have to be made, such as what deliveries to change and if there is not enough order picking capacity how to facilitate this. The trade-off has to be made between costs made on the distribution centres (DC), store and transportation costs while not being able to evaluate these costs properly. The employees determining the DTWP for the BWs are from different disciplines and have a good idea what a certain change implies to their disciplines’ costs, but an incremental approach is needed to assess the overall costs properly (Beerepoot & Spoelstra, Interview 1, 2014). Figure 3 on the next page, schematically shows the current process of determining the DTWP for the BWs.

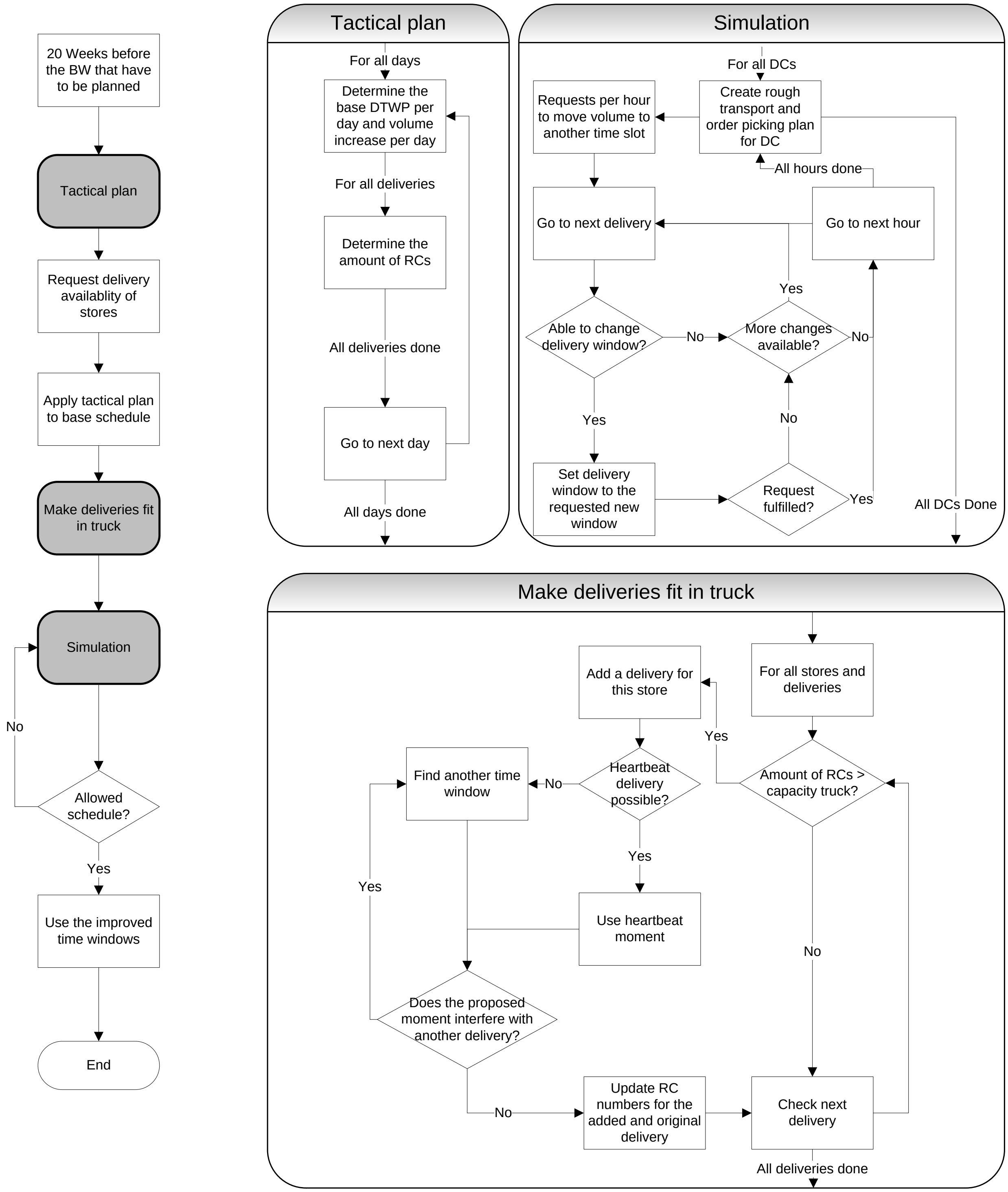

Figure 3: Current situation of determining DTWP for the busy weeks

2.2 The ORTEC Solution: WDScan

To answer the first part of the second research question '*What does the WDScan method look like and what are its advantages and drawbacks'*, this section provides insight in the WDScan model built by ORTEC and its advantages. This section gives a general impression of the model. Section 2.3 provides a more in depth view of the model and displays its drawbacks by giving a thorough analysis of the model. Section 2.2.1 gives an overview of WDScan and its advantages. Section 2.2.2 describes the input part of WDScan. Section 2.2.3 explains the WDScan model. Section 2.2.3 elaborates on the output of WDScan.

2.2.1 The purpose and advantages of WDScan

The WDScan model was built to improve a more general situation of the process described in Section 2.1. It was built to determine what the best DTWP in general is, when taking into account store, DC and transportation costs and the amount of customers bothered; the problem as described in Section 1.2. However, the DTWP that WDScan creates is not acceptable, because (1) it does not imply lower overall costs than the manually generated DTWP and (2) the manual DTWP is considered infeasible by WDScan, while this DTWP is currently used and therefore feasible. In order to gain insight in the model it is important to know how the parts are connected. Figure 4 gives an idea how the different parts of WDScan are connected.

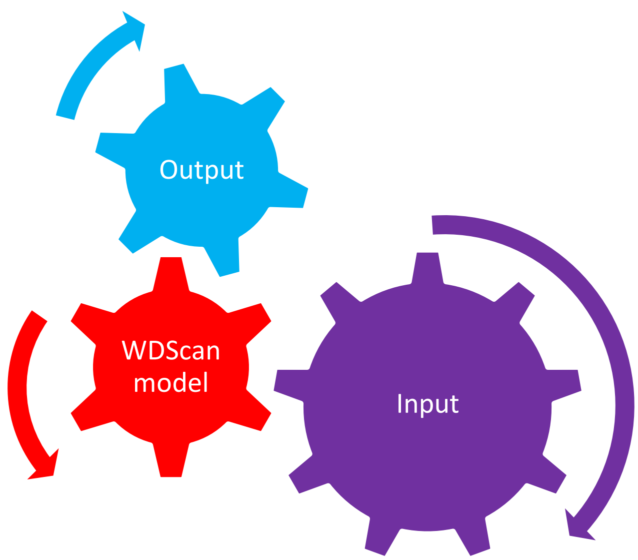

Figure 4: WDScan connections

As seen in Figure 4 the input gear is relatively large; this represents the importance of the input. In the WDScan model, input is not just raw data. In addition to raw data, output of other models is used. An example of such output is the output of a congestion model. However, some of these input parameters are based on simple assumptions. Based on this input data, WDScan evaluates the costs of a certain DTWP. The WDScan model calculates the costs and customers bothered, and uses simulated annealing to improve this costs function incrementally.

2.2.2 The input of WDScan

The input has a major influence on the results of WDScan. A lot of data is used in the model to calculate the costs of a certain DTWP, for example for every store the amount of customers for each hour of each day. In addition, there are some generic input data entries, for example the transportation costs per km and transportation costs per hour. These parameters do not change over time or when driving to store X instead of Y. However, they can be changed for the different kind of vehicles that are available to Albert Heijn. Based on the characteristics of a distribution centre (DC) input parameters vary, for example the order picking costs for each hour, DC, day and flow (ambient or fresh goods).

The amount of goods that has to be delivered to the stores and the amount of customers are input data. The value of these input parameters is determined for each store, day, hour and flow. Several input parameters are based on choices rather than on raw data. In addition, for some input parameters store specific values could be defined, but rather a general value is used. This creates a lot of freedom, because we can change the output of the model by modifying input data. Increasing the level of detail of these parameters changes the output significantly.

2.2.3 The model choices of WDScan

One of the key assumptions of the WDScan model is that the flow of a delivery is as shown in Figure 5 and Figure 6. WDScan assumes that all these activities are direct predecessors and are started when the former is completed. Figure 5 shows the process at one of the DCs (distribution centres) of Albert Heijn. Based on the delivery time window chosen, the loading of the truck at the DC is finished at the beginning of the delivery time window, minus the time to travel to the store. The activities are therefore considered to be direct predecessors of each other, and calculated from the decision variable: the delivery time window of a single store. Figure 6 shows the store process, where the same assumptions are made as for the DC process: all activities are direct predecessors of each other. Here the unloading and allocation of goods to sections starts in the middle of the determined delivery time. The restocking starts when the unloading and allocation of goods to sections is finished.

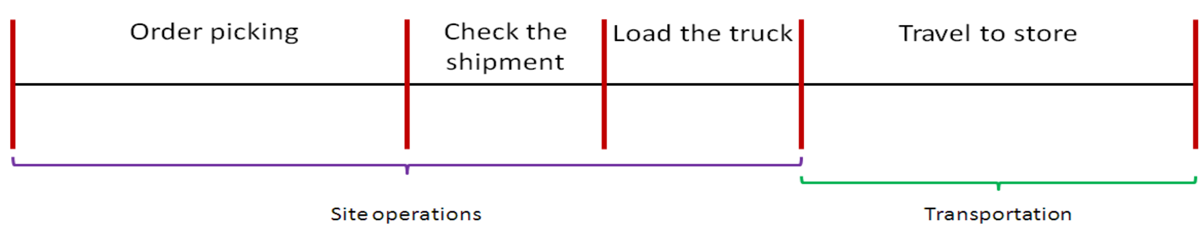

Figure 5: Distribution Centre process, adapted from ORTEC (2013)

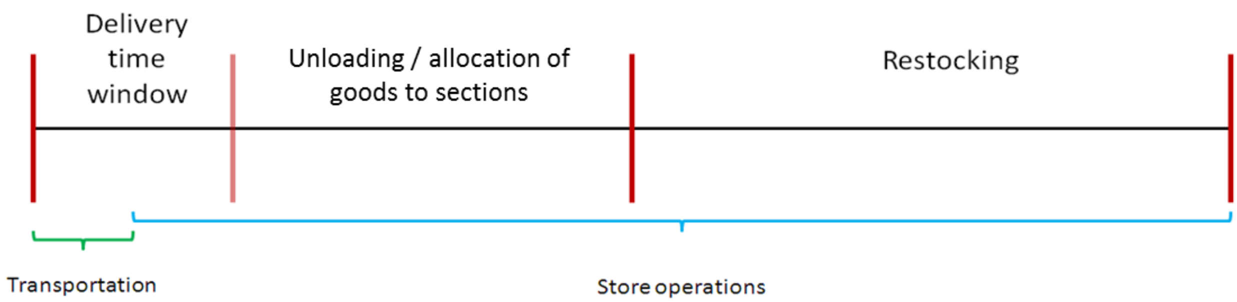

Figure 6: Store process, adapted from ORTEC (2013)

All of the operations mentioned in Figure 5 and Figure 6 incline certain costs. The WDScan model calculates these costs for all of the stores and sums over all of the stores. In addition to the store, distribution centre and transportation costs, the amount of customers bothered is taken into account. The amount of customers bothered is the sum of the amount of customers bothered for each delivery made to store i. To be able to compare the amount of customers bothered among stores, a method is created to compare this among different stores. Based on the model used by Albert Heijn to calculate the amount of customers bothered, ORTEC (2013) has formulated the following equation to calculate the customer bothered part of the target function:

$$Customers\ bothered_i = \sum_d \frac{customers\ in\ store\ during\ restocking_{di}^2}{floor\ space\ of\ store_i}\ \forall i\ ,$$

where each d is a delivery and the customers bothered for a store is the sum over the customers bothered of all of the deliveries to store i. The customers bothered is a part of the target function. The real costs in the target function are the distribution centre, store and transportation costs. In addition to these costs, penalty costs are present in the target function. Penalty costs occur if one or more restrictions are violated. However, these penalty costs increase the solution space for simulated annealing, and allow it to find more neighbour solutions. These penalty costs are also introduced, because the current DTWP used by Albert Heijn would not show a feasible solution in WDScan, because multiple restrictions are harmed in the current DTWP. Combining these components gives the following target function (ORTEC, 2013):

$$\begin{aligned} Min\ z = \sum_i \{ & A(Customers\ bothered_i) + B(Transportation\ costs_i) + C(\,Store\ costs_i) \\ & + D(Distribution\ centre\ costs_i) +\ penaltycosts_i\} + global\ penalty\ costs \end{aligned}$$

In this equation, i represents the index for individual stores. All of the costs are allocated to individual stores and summed over all of the stores. The exceptions are the global penalty costs, which are not allocated to a single store. The customers bothered are multiplied by value A to compare it to the real costs and penalty costs. In addition, the real costs are multiplied by their relative importance value B, C and D. These relative importance values allow Albert Heijn to influence the target function to their preference.

The store costs are calculated based on the amount of load carriers (RCs) delivered, and the time of the delivery. Per store, the input data states a costs value for restocking at a certain hour. These costs are multiplied by the amount of RCs delivered to the store at the delivery time window and divided by the restocking rate per store. In the WDScan input the restocking parameters are generic, not differentiated per store or hour.

The transportation costs are estimated by an model constructed by ORTEC that starts by selecting stores that could be delivered in one trip. It evaluates the costs of transportation using the delivery times of stores that are in the same cluster. The WDScan model creates clusters the following way (ORTEC, 2013):

- Create a number of base stores that are the start of a cluster
- Create a list of stores that are within a certain range of a base store that have the same DC and the same truck type
- Clusters of stores can only be created between stores within this list
- Based on a delivery time difference of less than 60 minutes, WDScan selects stores until the capacity of the truck is reached or the maximum number of stores in a cluster is reached.
- If there are no more combinations to be made the remaining stores are delivered in a separate trip

The transportation costs are the sum of the time a trip takes, times the costs of a driver per hour and the km of a trip, times the costs per km. The transportation costs are allocated to the store that is delivered by this trip. If two or more stores are delivered in one trip, the costs are divided using a formula designed by ORTEC.

Finally, the DC costs are based on the time the orders have to be picked, loaded and unloaded. These tasks have certain fixed and variable amount of time. The different times are multiplied by the DC costs for the hours of the order picking and loading at the distribution centre. The distribution centre costs are allocated to individual stores. The costs in the distribution for order picking, checking the shipment and loading the truck for a delivery are allocated to the delivered store. If two or more stores are delivered in one trip, the formula used to divide the transportation costs is used.

2.2.4 Output of WDScan

The purpose of WDScan is to provide a better DTWP. Hence, the main output of WDScan is a new DTWP. In addition, WDScan provides performance indicators to analyse the solution. The output is written to an Excel file for convenience. This Excel file consists of:

- A worksheet containing the new delivery windows of all of the stores for each flow (ambient and fresh). The amount of load carriers and goods (size of the delivery) are displayed for all of the delivery windows. The new DTWP is the output of the decision variable.
- A worksheet containing store costs. This tab displays the transportation, customer, distribution centre and store costs allocated to each store. In addition, the penalty costs that are calculated

per store are shown for each store. The sum of these costs is the total costs associated with the DTWP[2].

- A worksheet containing the calculated time intervals for the total delivery. This worksheet states the time for every delivery: order picking starting time, the loading at the distribution centre start time and the time all of the other activities displayed in Figure 5 and Figure 6 start.
- Worksheets containing the amount of trucks in use, the amount of order picking at the DC and the amount of outgoing trucks at the DC. These tabs give insight in the calculation of the global penalty costs. These penalty costs are explained in more detail in Section 2.3.3.
- A worksheet containing the original DTWP, which is used as the start of the optimization process WDScan performs.
- Worksheets containing the overview of the optimization process, displaying the objective value in the stages of the simulated annealing process. In addition, the parameter settings and penalty costs functions weights are displayed.

The output gives a good overview of how the model works and helps understand how the specific model choices actually work. A large part of the analysis is done modifying or structuring the output data to gain insight in the model. The analysis is explained in Section 2.3.

2.3 Analysis of WDScan

This section provides a more in depth view of the WDScan model. In order to answer the second research question: *'What does the WDScan method look like and what are its advantages and drawbacks'*, Section 2.2 introduced the WDScan model. This section elaborates more on the model and provides an analysis of the model to show the major drawbacks to improve. This section starts by explaining the approach used to analyse the model and then gives the outcome of the analysis. We choose this approach to better structure the analysis. The remainder of this section describes the analysis per group as described in the section containing the approach, Section 2.3.1.

2.3.1 Approach

In order to analyse the WDScan model we use several techniques: Interviews with the supply chain officers at the SCCP department, analysis of the output of WDScan, an in-depth analysis of the input variables and a quantitative analysis of the model choices in WDScan. In general we can split the non output part of WDScan model into input data and model choices, as explained in Section 2.2. We therefore use the same split in the analysis of the model. The approach used to analyse the validity of the model is to validate the individual input and model choices, in order to find flaws. To find flaws we investigate the output and investigate what causes the not expected output. Once the individual parts are validated or improved, the output should be re-validated to see if the improvements to the individual parts have resulted in improved output. The re-validation of the output is outside of the scope

[2] These do not include the global penalty costs. Section 2.3.3 shows that in addition to the store specific penalty costs, there are global penalty costs that cannot be allocated to a single store.

of this thesis, because the improvements to the model first have to be implemented in order to test the output again.

Figure 7 schematically shows the approach used and outcome of the analysis. Based on the input parameters and the parts of the WDScan model, we have reduced the possible improvements to three groups to improve. Because there are a lot of input parameters, we reduce them to a few items that are interesting to improve. Based on interviews with supply chain officers and the analysis of the output we have reduced the huge list to 5 groups of items as possible improvements. Appendix A displays the list, containing the starting input data and considerations for improvement. We choose to exclude this from the body of the thesis due to the size of this list. However, this list is important, because it illustrates the process of selecting what input to consider neatly. Section 2.3.2 describes the analysis of the input part of the model. Here we describe in more detail the selection of the 5 groups of parameters. In addition we assess what of these parameters' improvements are within the scope of the research and have a big impact on the outcome of WDScan.

In addition to the input, the model parts and solution method is investigated. The model parts are the parts of the target function:

- The customer bothering model
- The transportation costs model
- The distribution centre costs model
- The store costs model
- The penalty costs model

We investigate these model parts and the choice of simulated annealing as a solution method. In addition, we investigate the weights of the target function. The selection of items to improve is based on the interviews with the supply chain officers and the analysis of the output. Section 2.3.3 describes the parts of the model in more detail. In addition, Section 2.3.3 selects parts of the model to improve based on impact on the validity of WDScan. Figure 7 provides an overview of the discussed selection process. This figure shows the selection of three items to improve: delivery size estimation per hour, the penalty costs model and the distribution centre costs model.

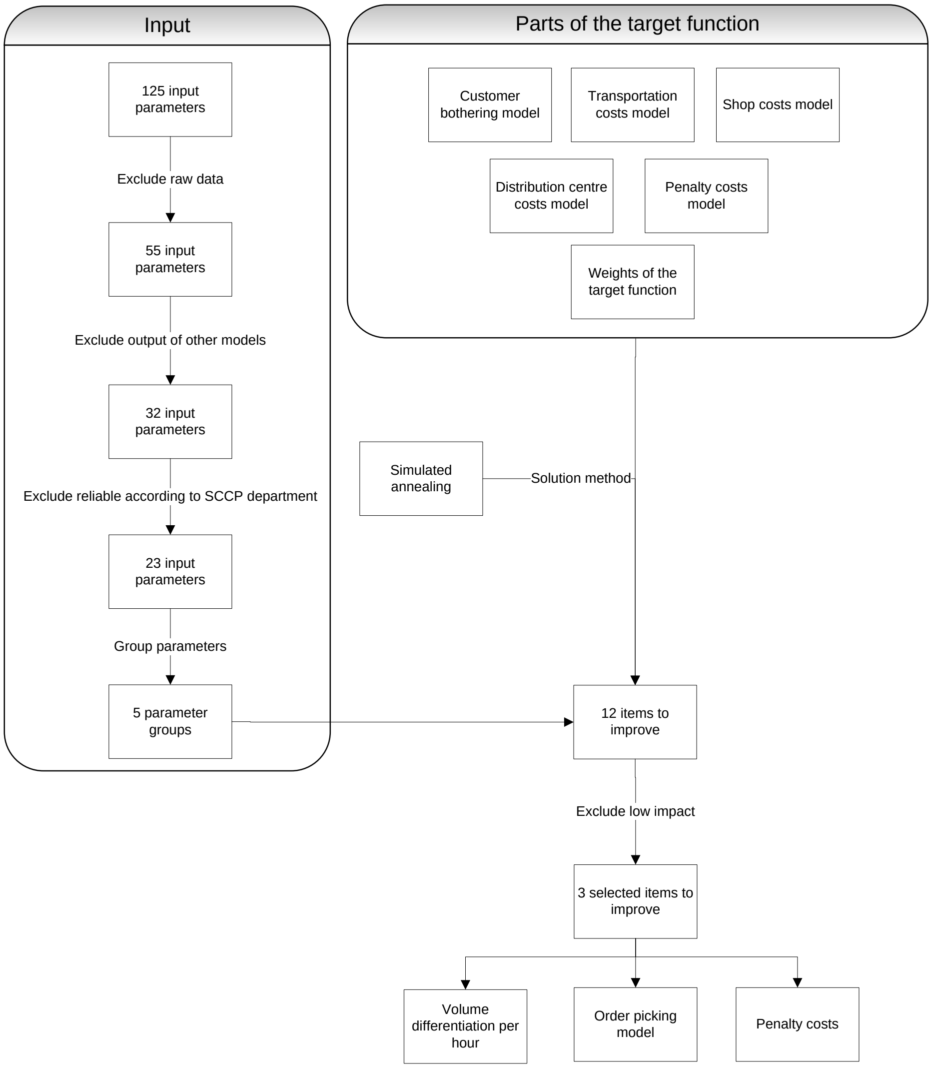

Figure 7: Selection of items to improve

2.3.2 Motivation for what input items to improve

This section describes the steps conducted in the process of selecting the input parameters to improve. An overview of the selection is shown in the Input box in Figure 7. In order to get good output, the input has to be solid. As stated in Section 1.2 the current output is not usable. Our approach to solve the problem of getting proper input data is to question the input data. We have characterised the input into different types of input. This segmentation is done to indicate what data are generated based on rule sets and what data is imported from other databases as raw data. In addition, some input data is the output of other models and some are choices that are generic for all entries of the input file. Appendix A shows the table that displays the overview of the input parameters. As explained in Section 2.3.1., this table is not displayed here, because of its size. We excluded the parameters that are imported directly from a database with raw data for improvement. Examples of raw data are the average amount of customers for each hour for a specific store. These values are extracted from a database and put (almost directly) into the input file for WDScan. Another example of raw data is the address information of each store. The amount of input parameters that we start with is 125.

In order to find parameters that are interesting to improve, we excluded from investigation all of the raw data as possible subjects to investigate. However, the data in the databases has to be correct. Because the data subtracted from these databases is not modified and based on historical or location data, the SCCP department thinks most of this data is reliable. We do not investigate the raw data, except for the data Albert Heijn thinks is unreliable (Beerepoot & Spoelstra, 2014). This selection of input data results in 55 parameters remaining to investigate. The output data of other systems, if used as input for other planning systems at Albert Heijn is reliable for WDScan. Using the input used for other planning systems of Albert Heijn is reliable, because based on these models the costs for Albert Heijn are evaluated. An example of such output is the output of the congestion matrix, which is used in the normal transport planning process. The data that are not used in the normal planning process, or Albert Heijn thinks is not reliable, we include for investigation. 32 parameters remain after this selection.

The remainder of the input data is specifically created for WDScan. In general these are based on practice, to model the current situation. Examples of this data are the restocking costs per hour. Parameters used to mimic decisions made in the current process are excluded from investigation. We keep the parameters that the SCCP department does not think are properly used on the list. The input rules are compared with the business rules used in the current (manual) process. The parameters that mimic the business rules of Albert Heijn properly are excluded. The 23 remaining parameters we group into five categories, which will be explained in the remainder of this section:

- Costs of transportation, distribution centre and store per hour
- Time between two deliveries to a store
- Amount of outgoing trucks per hour per DC per flow (ambient or fresh)
- Delivery size estimation between deliveries to a store
- Delivery size estimation per hour

The remainder of this section argues whether or not to improve these groups of parameters. In the order of the above list we motivate what groups to improve. The selection is based on the impact of an improvement to one of these groups, and whether it is within the scope of this thesis to improve the group.

The *Costs of transportation, distribution centre and store per hour* are not taken into account in this research, because the project WD has verified these costs for the current DTWP. However, it is not certain whether these costs will be accurate when changing the DTWP. The change in costs is especially uncertain when changing a lot of delivery time windows for individual stores. We think the changes should be evaluated after the basis of the model works. We think Albert Heijn should start by assessing the value of small changes, to investigate if these costs act as predicted. Once the small costs changes are verified, larger changes should be evaluated. The validation of costs changes is not within the scope of the project, because it is not possible to assess these costs changes within the time available to do this thesis.

The *time between two deliveries to a store* problem is solved when the penalty costs are properly assigned[3]. The *time between two deliveries to a store* is a restriction that we do not want to violate, except if there is no other possibility. Changing the penalty costs makes sure no restrictions are violated except if there is no other option. The *time between two deliveries to a store* restriction could be violated if a store has a very small delivery window or a lot of deliveries every day. The *time between two deliveries to a store* can be given per store, therefore changing this variable for the stores that have to violate the restriction solves the problem. Therefore, we solve this issue when improving the penalty costs.

The amount of outgoing trucks per hour per DC per flow is an input parameter to model the capacity restriction of the amount of trucks that can leave the distribution centre per hour. This parameter is correct in the current way of modelling the order picking process. However, because the SCCP department thinks the order picking model is not valid this parameter should be re-evaluated after improving the order picking model. Because we improve the order picking model[4], we should re-evaluate this parameter.

The delivery size estimation between deliveries to a store is not investigated. A project to improve this estimation is done simultaneously to this thesis. The outcome of this project should be used to improve the model, however improving this estimation not within our scope.

Delivery size estimation per hour is chosen for improvement. The delivery size estimation per hour is the change in volume delivered to a store when changing a *heartbeat moment*. A heartbeat moment is the delivery time for each flow, where this delivery time is the same for every day a store is delivered for this flow. The volume changes are not that big when changing a heartbeat moment, however, they are an estimate of what amount of goods a store will receive. In order to gain the support of the stores that get changes in their heartbeat moment, it is important that the estimate is accurate. The store managers create a personnel schedule based on these estimates, and prefer an accurate estimation. The model proposed in Section 4.1 is accurate in predicting the amount of goods a delivery has when the heartbeat moment changes. WDScan currently uses a fixed amount of goods that does not change when changing a heartbeat moment, which is a good estimate for the ambient goods. However, for the fresh goods this can vary. In order to get a more accurate estimate of the delivery size, this has to be improved.

---

[3] Section 2.3.3 explains the motivation for improving the penalty costs.
[4] Section 2.3.3 explains the motivation for improving the order picking model.

To conclude, we summarize the three input parameters that are selected for improvement. *The delivery size estimation per hour* is selected for improvement. The second input parameter we select for improvement is the *amount of outgoing trucks per hour per DC per flow*. The final input parameter selected for improvement is the *time between two deliveries to a store*. We note that merely the first selected improvement is stand alone, the latter two are both improved when improving parts of the model. The selection of the model choices to improve is described in Section 2.3.3.

2.3.3 Motivation of the model parts to improve

This section describes the motivation of what model parts to improve. We first explain the target function in detail and analyse the output of WDScan for a current delivery time window plan of Albert Heijn. The second part of this section explains what parts of the model we improve.

The target function of WDScan compares the real costs to the artificial costs. The real costs are the transportation, distribution centre and store costs. The artificial costs are the customers bothered (times a value for the customers bothered) and the penalty costs, as seen in the target function:

$$\begin{aligned} Min\ z = \sum_i \{&A(Customers\ bothered_i) + B(Transportation\ costs_i) + C(\ Store\ costs_i) \\ &+ D(Distribution\ centre\ costs_i) +\ penaltycosts_i\} + global\ penalty\ costs \end{aligned}$$

The parameters A, B, C and D are used to display the relative importance of costs of different departments or costs of bothering customers. Balancing the real costs and the artificial costs is difficult, because in the penalty costs, some costs are implicitly modelled. For example, the truck spread penalty is used, because the transportation department wants to offer long journeys to the logistic service providers to get a better price per km. In order to get longer journeys, the amount of trucks should be evenly spread over the day, therefore a penalty is implied when the amount of trucks is not evenly spread. Figure 8 shows the contribution of the different parts of the target function to the objective value for a current delivery time window plan of Albert Heijn calculated by WDScan.

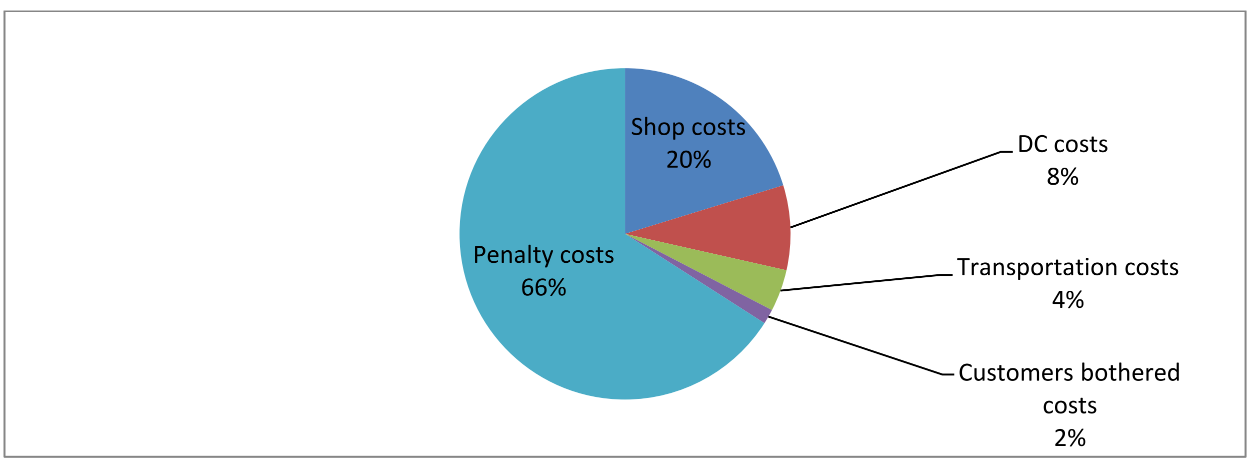

Figure 8: WDScan target function parts for the current DTWP

Figure 8 shows that the majority of the target function are penalty costs. The parameters and the penalty costs are the key to get good output of WDScan. The preference of Albert Heijn should be displayed in the parameters. However, to be able compare the real costs and customers bothered, the penalty costs should not be such a large part of the target function for a feasible schedule.

Based on interviews and analysis of the output, as described in this section, we motivate what parts of the model should be improved. The parts of the WDScan model are: *the customer bothering model, the transportation costs model, the store costs model, the distribution centre costs model and the penalty costs model*. In addition we investigate *the weights of the target function and the use of simulated annealing*. We analyse the parts of the model by getting an in depth view of the output of the model and how these costs are constructed. We first explain the parts of the model we select for improvement: *the penalty costs model* and *the distribution centre costs model*. We continue with explaining why we did not the select the other parts of the model for improvement.

We select the *penalty costs model* for improvement. The penalty costs are divided in different aspects: penalties for violation hard restrictions and penalties for violating soft restrictions. The hard restrictions are the restrictions we do not want to violate, except if there are no other options. An example is that Albert Heijn wants to have at least 2 hours between two deliveries to a store. However, some stores can only be delivered three hours a day and receive three deliveries. In this case it is not possible to have two hours between deliveries. In addition to the hard restrictions, soft restrictions are modelled in WDScan. The truck spread mentioned earlier in this section is one of the soft restrictions, the other one is the order picking spread. These restrictions have to make sure the truck usage or DC order picking amounts do not fluctuate too much over the days. Figure 9 shows the costs of the three different kinds of penalties: the order picking spread penalty, the truck spread penalty and the penalties for violating the hard restrictions. This figure shows that the majority of the penalty costs are order picking spread penalty costs and that the hard restriction violation penalty costs are just a small part of the costs in the current DTWP. The violation of hard restrictions is mostly due to restrictions that have to be violated, such as the time between two deliveries to a store, as mentioned earlier.

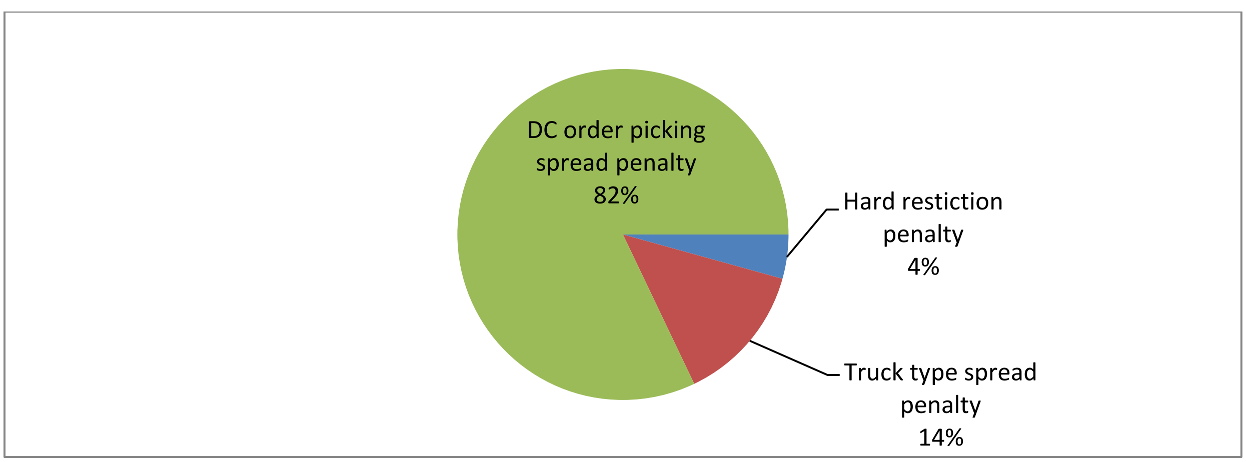

Figure 9: Penalty costs of the current DTWP

The *penalty costs model* needs improvement. We divide the penalty costs model in the hard penalty costs and the soft penalty costs. We improve the hard penalties, because WDScan violates hard restrictions in favour of soft restrictions when improving the target function in the current penalty settings. This means WDScan moves from the feasible area when optimizing. We think the DC order picking spread penalty should also be improved, because it is such a large penalty in the current DTWP. We improve this penalty while improving the whole *distribution costs model*.

Improving the *distribution centre costs model* and its penalty costs is required, because a lot of penalties occur for the penalty related to the distribution centre costs model. The related penalty to the distribution centre costs model costs constitutes a large part of the penalty costs, as seen in Figure 9. Based on the output of the analysis we find that the order picking model, which is the main part of the *distribution centre costs model*, is very different to the order picking model in practice. The order picking model in WDScan is based on just-in-time order picking, while the distribution centres of Albert Heijn order pick in advance. Therefore, the *distribution centre costs model* is selected for improvement by improving the order picking model.

We do not select the *customer bothering model* for improvement, because the model is similar to the current evaluation method of customer costs. The current method is based on whether the amount of customers is above the average amount of customers per m^2 over all of the stores of Albert Heijn. If the amount is higher than the average amount plus a given amount, the amount of transactions per m^2 is considered too high to restock at this moment. The model ORTEC built to measure the customer different; they count every customer during the restocking process as a customer being bothered. However, they square the amount of customers per m^2. This means a larger amount of customers compared to the square meters is considered less favourable. This method is similar, however not the same. The fixed values in the original model pose another problem: they have to be indexed. The average amount of customers per m^2 is used in the current model, based on the average over a certain period of time. The model used by ORTEC is robust to changes in the amount of transactions per m^2, which makes it favourable over the current method used. We think this give a good indication of the amount of customers bothered by restocking. However, because the amount of transactions is used to measure this, e.g. the amount of payments at the till, one could doubt whether the amount of transactions is a good indication of the amount of customers in the store. The amount of customers in the store per m^2 is assumed to be the factor that bothers customers. A statistical study conducted by Albert Heijn shows that the amount of transactions is not the best estimator for the perception of crowdedness by customers. The amount of transactions per m^2 combined with the amount of goods sold per m^2 gives a better view of the crowdedness in a store. This assumption is verified by a field study among store managers (Van Lunteren, 2007), where they had to indicate what hours are their most crowded hours and matched with what hours would be suggested by the indicators. We suggest improving the customer bothering model by using in addition to the transactions per m^2 per hour, the amount of goods per m^2 per hour. However, to develop and test a new customer bothering model is outside of the scope of this thesis.

We do not select the *transportation costs model* for improvement. The model used by Albert Heijn to create the transportation plan is built by ORTEC, the manufacturer of WDScan, therefore it seems reasonable to assume they share the same principles. After requesting more information about the model at ORTEC to investigate this, the consultant at ORTEC replied that the model used is *'based on their own method, that has proven to be sound over a long time'* (ORTEC, 2014). This means we cannot get a measure from ORTEC about the accuracy of the heuristic, compared to the actual planning method. However, the transportation planners have validated that the output of this model seems reasonable in a previous version of WDScan. We do think the transportation model should be validated properly by comparing the output of WDScan to the output of the transportation algorithm used by Albert Heijn. We do not investigate this further, because Albert Heijn does not think the transportation model is flawed (Beerepoot & Spoelstra, Flaws in inputdata, 2014) . In addition, the impact on the target function is not as big as the other improvements, therefore we put this outside of the scope of this research.

We do not select the *store costs model* for improvement. The main reason for not improving the *store costs model* is that this is not yet applicable, because the penalty costs and order picking model are not valid yet. Changes to the *store costs model* will not have a large impact on the solution and target function value. We therefore suggest investigating this after improving the order picking model and *penalty costs model.*

The use of *simulated annealing*[5] is a good way to get to a better solution to our problem. Because a large part of the WDScan target function is based on penalty costs in the current situation, simulated annealing can help to get out of a local optimum (ORTEC, 2013). Schutten (2013) claims an implementation of simulated annealing in practice requires a well chosen problem representation, an incremental costs calculation and a proper cooling schedule. We think the problem representation is usable for simulated annealing, because we can compare different costs. However, some of our costs are not actual costs. The choice of penalty costs and parameters should represent the preferences of Albert Heijn. This means the problem definition is a bit unclear at the moment, and should be improved. However, as explained in Section 2.3.3, this can only be done after the other improvements of Section 2.3.3. WDScan provides an incremental costs calculation, because neighbour solutions are created by changing a delivery time window of a store and evaluated on costs. However, in general, simulated annealing starts from a random initial solution (Henderson et al., 2003), while WDScan starts from the current DTWP. In addition, WDScan facilitates to lock some delivery time windows, because it is not accepted by Albert Heijn that these windows change. If a random initial solution is used, the locking of certain delivery time windows could not be facilitated.

We do not select *the weights of the target function* for improvement, because it is not yet applicable. This involves the weights of the parameters described in the start of this section. This has no use when

[5] Section 3.4 explains simulated annealing in more detail.

the penalty costs and order picking model is not valid yet. In addition, the soft penalty costs are closely related to this part, because they are not direct costs and should represent what direction the model should take. We therefore split the penalty costs into the hard restrictions and the soft restrictions. The soft restrictions should be taken into account when comparing the costs and customers bothered.

To conclude, we summarize the model choices selected for improvement: *the distribution costs model* and the *penalty costs model*. Where the hard penalty costs are improved and the soft penalty costs are not, because the soft penalty costs have to be evaluated when assessing the weights in the target function. However, we do improve one of the soft penalties: the order picking spread penalty, which is closely related to the distribution costs model. Assessing *the weights of the target function* and *the customers bothered model* are not within the scope of the research. *The customer costs model* is not selected for improvement, because the model is similar to the customer costs calculation in the current situation. *The transportation model* is not selected, because it was validated previously. We think *simulated annealing* is a proper solution method for WDScan, mainly because of the use of penalty costs in WDScan.

2.4 Conclusion

This chapter answers the research questions '*What does the current situation look like*' and *'What does the WDScan method look like, and what are its benefits and drawbacks'*. We have shown that the in the current situation a delivery time window plan (DTWP) is based on manual process and for the busy weeks (BW) an optimization is done to see if the plan is feasible. The WDScan method should be able to automate this process and reduce the amount of customers bothered. WDScan uses an incremental approach to determine what the 'best' delivery time window is for a single store, based on real distribution centre, store, transportation costs, and artificial customer bothering and penalty costs.

However, according to the SCCP department of Albert Heijn there are some drawbacks to this model. The current model is not able to evaluate the current manual DTWP properly and the DTWP created by WDScan after optimization is actually worse than the starting point. Based on interviews with supply chain officers at the SCCP department and analysis of the output of WDScan we have selected three improvements to the WDScan model. First, the input parameter: delivery size estimation per hour. This parameter describes per store what the amount of goods (volume) of a delivery to a store would be if the heartbeat moment is at a different time than the original heartbeat. This improvement is selected, because of the impact on the stores of Albert Heijn. Second, the order picking model is selected for improvement. The order picking model simulates the order picking at the distribution centres of Albert Heijn. Improving the order picking model is selected, because a lot of penalty costs occurred when evaluating the current DTWP in the order picking part. Improving the order picking model would have a major impact on the target function and solution WDScan calculates. The last improvement selected is the penalty costs model, because WDScan violates a lot of hard restrictions in favour of soft restrictions. As a result, WDScan moves from the feasible region. Moving from the feasible region makes the new DTWP not executable. Chapter 4 elaborates on the actual improvements of the improvements selected in this chapter.

3. Literature review

The goal of this chapter is to answer the third research question: *'What are similar problems to the WDScan problem in literature, and what are possible solutions for this problem in literature?'*. To illustrate what similar problems are, we present the base problem, and show what kind of special case our problem is. Section 3.1 explains the general combinatorial optimization (CO) problem. Section 3.2 presents an example of a CO problem: the vehicle routing problem (VRP). Section 3.3 shows that our problem is comparable to an inventory routing problem (IRP), a general case of the VRP. In addition, this section explains the general IRP and in particular an implementation of an IRP at Albert Heijn from literature. Section 3.4 focusses on solving a CO problem and presents the simulated annealing method, used by WDScan. Section 3.6 describes methods to compare costs and qualitative matters. Section 3.5 gives insight how to cope with penalty costs. Finally, Section 3.6 states the conclusions of this chapter.

3.1 CO problems

A Combinatorial Optimization (CO) problem is a problem of finding an optimal solution among a finite number of possible solutions (Papadimitriou & Steiglitz, 1982). In combinatorial optimization a *criterion* is measured by an *objective*, the objective is to maximize or minimize. Based on the *decision variables* a best alternative is chosen. The *solution space* covers all the possible solutions that are allowed by the *constraints* and *parameters*. There are many fields in which CO has improved performance. In logistics examples of these fields are (Schutten, 2012):

- Transport and Distribution
- Technical aspects of production
- Production planning and scheduling
- Allocation and location problems
- Warehousing
- Healthcare

Among these CO problems, the Travelling Salesman Problem (TSP) is probably the most well-known (Aarts & Korst, 1989). The TSP a problem where a salesman has to visit certain cities exactly once in a tour, starting and ending at his home city. The objective is to minimize the total tour length. The travelling salesman problem is a nice example, because it is simple to explain and find a feasible solution to the problem, but finding the best solution is very difficult (Garfinkel, 1985). In a CO problem we minimize or maximize an objective, specified by a set of problem instances. These instances can be formalized as a pair (S, f). The solution space S denotes the finite set of all possible solutions, and the costs function f is defined as (Aarts & Korst, 1989):

$$f : S \rightarrow \mathbb{R}$$

In the case of a minimization problem, the problem is to find a solution $i_{opt} \in S$ that satisfies

$$f(i_{opt}) \leq f(i), for\ all\ i \in S$$

In the case of maximization, i_{opt} satisfies

$$f(i_{opt}) \geq f(i), for\ all\ i\ \in S$$

Such a solution i_{opt} is called a 'globally- optimal solution' In this formulation$f_{opt} = f(i_{opt})$ gives the optimal costs and S_{opt} the set of optimal solutions. The possible solutions are bound to constraints as mentioned before.

In general CO problems can be divided in two different classes: hard and easy problems. The difference between these problems is that the easy problems can be solved in polynomial time and of most hard problems it is not yet shown that they can be solved in polynomial time. Polynomial time means that if the problem gets larger, (and therefore computation time increases), the computation time will increase polynomially with the problem size. An example of an easy problem is an LP-problem. However, most problems that require variables to be integer are harder than non-integer problems (Vanderbei, 2008). Most real word problems require variables to be integer. For example, one cannot hire half an employee.

Most practical and theoretical problems are NP-hard. NP-hard problems cannot be solved in polynomial time and therefore a trade-off has to be made between optimality at immense computation time and sub optimality in polynomial time (Aarts & Korst, 1989). Due to the fact that easy problems are solvable in polynomial time and most practical problems are NP-hard, Section 3.4 dives into the different methods to 'solve' NP-hard problems.

3.2 Basic VRP

Since the first paper on the Truck dispatching problem (Danzig & Ramser, 1959) there have been a lot of papers on the Vehicle Routing Problem. A Google scholar search gives over 150,000 results for the term 'Vehicle Routing Problem'. The problem that Danzig and Ramser (1959) describe is delivering gasoline to gas stations and they gave the first formulation of what we now call the VRP. Most papers are about how to solve the VRP or a problem closely linked to the VRP. The interest in the VRP is mostly because of its application in practice and the difficulty of the problem: the most efficient algorithms can solve VRP up to about 50 customers to optimality; some particular cases can be solved for bigger problem instances (Toth & Vigo, 2002)

The VRP is a problem of how to dispatch trucks to customers. The VRP problem is usually displayed as a graph. In this graph the roads are shown by the arcs and the customers by the nodes in the graph. In addition to customers the depots are also present in the graph. From a depot vehicles dispatch to deliver goods to customers. These trucks and customers can have different characteristics. In addition, the objective function can also have different characteristics, as shown in Table 1.

Characteristics		
Customers	**Vehicles**	**Objective function**
Location on the graph	Home depot of the vehicle	Balancing of routes on travel time or vehicle load
Demand of the customer	Capacity of the vehicle	
Delivery time the customer can be served	Devices available for the loading and unloading	Minimization of the penalty costs for partial deliveries
Time required to deliver (unloading and loading times)	Subset of arcs that can be traversed by this vehicle	Minimization of the amount of vehicles used
Subset of vehicles that can deliver goods to this customer	Costs when using this vehicle (distance units, time units, etc.)	Minimization of the global transportation costs

Table 1: Different characteristics of Customers, Vehicles and Objective functions for the VRP (Toth & Vigo, 2002)

The VRP is NP-hard, because it is a generalization of the TSP (Cordeau et al., 2005). The TSP is explained in Section 3.1.

3.2.1 Formulation of the general VRP

This section describes a formulation of the *symmetric* VRP. In this instance of the VRP it is assumed that the distance from customer i to customer j is the same as the other way around. The formulation we present is proposed by Laporte, Nobert and Desrochers in 1985. The symmetric VRP is defined on a complete undirected graph $G = (V, E)$. $V = \{0, \dots, n\}$ is defined as a vertex set. Each vertex $i \in V \setminus \{0\}$ represents a customer having a non-negative demand q_i. Vertex 0 represents the depot. Each edge $e \in E = \{(i,j): i,j \in V, i < j\}$ has a travel costs c_e. A number of m identical vehicles with capacity Q are available at the depot. To solve the symmetric VRP a set of m routes has to be determined, where the goal is to minimize the total travel costs such that (Cordeau et al., 2005):

1. Each customer has to be visited once in a route
2. The route has to start and end at the depot
3. The total customer demand served by a route may not exceed the vehicle capacity Q
4. The route length may not exceed a pre-set limit L

This can be formulated as an ILP in the following way (Laporte et al., 1985):

$$Minimize \sum_{e \in E} c_e x_e \qquad (1)$$

Subject to

$$\sum_{e \in \delta(i)} x_e = 2 \ \ (i \in V \ \{0\}) \qquad (2)$$

$$\sum_{e \in \delta(0)} x_e = 2m \qquad (3)$$

$$\sum_{e \in \delta(S)} x_e \geq 2r(S)\ (S \subseteq V\{0\}, S \neq \emptyset) \qquad (4)$$

$$x_e \in \{0,1\} \left(e \notin \delta(0)\right) \qquad (5)$$
$$x_e \in \{0,1,2\} \left(e \in \delta(0)\right) \qquad (6)$$

In this formulation $r(S)$ denotes the minimum number of vehicles needed to serve the customers of a subset S. This value can be determined by solving a to this VRP linked *Bin Packaging Problem* (BPP) with item set S and bins of capacity Q (Cordeau et al., 2005). Define $\delta(S) = \{(i, j): i \in S, j \notin S \ o: j \in S, i \notin S\}$. In this formulation constraint (2) makes sure each customer is visited once. Constraint (3) makes sure that *m* routes are constructed. The capacity constraints (4) make sure that the solution is a tour that starts and ends at the depot and that enough edges are allocated to each vehicle selected. The BPP that determines $r(S)$ is NP-Hard and may be approximated by the lower bound by for example$\lceil \sum_{i \in S} q_i / Q \rceil$ (Cordeau et al., 2005). Constraints (5) make sure all edges between two customers are traversed at most once. Constraints (6) make sure that each edge at the depots is traversed at most twice.

3.3 The inventory routing problem

The inventory routing problem (IRP) is one of the more difficult and important extensions of VRPs (Bertazzi et al., 2008), because not only the routes have to be taken into account but also the inventory at the customers. A supplier manages the inventory of a customer to make sure they have no stock-outs. In more difficult cases, if holding costs are included, this environment is called a *one warehouse multi-retailer system* (Cordeau et al., 2005). IRPs are commonly seen in retail and the oil industry. The goal of the IRP is to minimize the distribution costs over a longer horizon of time.

The minimizing distribution costs over a longer horizon of time can be interpreted in several ways. We can distinguish three ways of interpretation: the strategic IRP, the tactical IRP and the infinite horizon IRP (Gaur & Fisher, 2004). The strategic IRP is based on finding a fixed partition policy for the IRP and determines how many vehicles are needed for this (Bertazzi et al., 2008). Research has shown that if customers require significantly less than a full truck load, direct shipments are not a good policy (Cordeau et al., 2005). The tactical IRP determines what set of customers to visit on a short time horizon and how many products to deliver to these customers (Cordeau et al., 2005). The infinite horizon IRP considers the long-run average transportation, ordering and inventory holding costs (Gaur & Fisher, 2004).

A study conducted at Albert Heijn by Gaur and Fisher (2004) addresses a similar problem to the one we are facing. They have qualified this as an IRP and found good solutions to this problem. This study is very interesting to us, because the study is conducted at Albert Heijn and the WDScan problem is a similar problem. We present the problem definition by Gaur and Fisher (2004) and their solution to the problem.

AH has to update their vehicle routing and delivery scheduling once every three to six months. This problem has the following main features (Gaur & Fisher, 2004):

1. The stores are delivered from one of the four regional distribution centres(DC)
2. All of the stores require replenishments several times per week. The time between two successive deliveries may not exceed a limit determined per store, due to the capacity restrictions of the stores and the nature of the products.

3. The time of the deliveries to a certain store are the same every week. However, they can vary from day to day, but have to be the same every week.
4. The demand per store is random and is variable with time. The hourly demand rate varies strongly over the days, but the pattern is rather constant over the weeks.
5. The delivery schedule is based on forecasts of demand but shipments are based on the actual orders by the stores. If the order from a store exceeds the truck capacity an additional shipment is scheduled to that store on the same day.
6. The routes determined are assigned to a heterogeneous fleet of trucks. These trucks have different fixed and variable costs. Some trucks are owned by Albert Heijn but the majority of the trucks is leased from external parties.
7. In the DCs the departure times of the trucks is as evenly spaced as possible to create a smooth workload throughout the day.
8. The delivery schedule problem must incorporate variable transportation costs, random demand, fixed and variable truck rental costs, workload capacity at the DCs and fleet-size constraints.

Based on these characteristics Gaur and Fisher (2004) created a model that aims to find the delivery times for each store, the clustering of stores together and the route of each individual truck to minimize transportation costs. In this model they assume they have a single homogeneous product with deterministic demand. The stores of Albert Heijn have a lot of different products, but because they are replenished from a single DC it is can be seen as a single-product problem (Gaur & Fisher, 2004).

Gaur and Fisher (2004) argue that their model cannot be categorized as one of the main IRPs, because normally IRPs are based on vendor managed inventory, while in this case the vendor is also the owner (in most cases) of the supermarkets. The model they create is called a periodic inventory routing problem, because the replenish cycle has to be the same every week. The model is based on a DC serving *n* stores, $i = 1, \dots, n$. The planning horizon is divided into T hourly periods. They define λ_{it} to be the demand of store i at time t. T_0 is the maximum time allowed between two successive deliveries to a store, $T_0 < T$. The index v defines the different types of trucks in the fleet and Q_v defines the capacity of each truck type v. k is defined as the index to enumerate the possible clusters of stores, with S_k denoting the set of all stores in cluster k. R_i is the set of clusters containing store i. The costs of transportation to store i from the DC and back using truck type v is defined as c_{iv}. The costs to deliver to all of the stores in cluster k is defined as $\bar{c}_{kv}$. $\bar{c}_{kv}$ is determined by solving a TSP for cluster k. For each cluster k the following decision variables are (Gaur & Fisher, 2004):

- u_{iktv} = number of direct shipments to store i in cluster k at time t using truck type v, $i \in S_k$
- v_{ktv} = number of shared shipments to cluster k at time t using truck type v
- q^d_{iktv}, q^s_{iktv}= quantity delivered by direct shipments and shared shipments, respectively, to store i in cluster k at time t using truck type v and 0 otherwise, $i \in S_k$

We show the formulation of the problem that has to be solved for each cluster k. This problem is formulated as a set of shortest-path problems. The formulation is as follows (Gaur & Fisher, 2004):

$$c_e(t_1, t_2) = min \sum_v \left[\bar{c}_{kv} v_{kt_1v} + \sum_{i \in S_k} c_{iv} u_{ikt_1v} \right] \quad (7)$$

such that

$$\sum_v \left(q^d_{ikt_1v} + q^s_{ikt_1v} \right) = \sum_{\tau=t_1}^{t_2-1} \lambda_{i\tau}, \quad (8)$$

$$q^d_{ikt_1v} \leq u_{ikt_1v} Q_v \; \forall \; i \; \in S_k, \quad (9)$$

$$\sum_{i \in S_k} q^s_{ikt_1v} \leq v_{kt_1v} Q_v \; \forall \; v, \quad (10)$$

$$u_{ikt_1v}, v_{kt_1v} \; integer \; \forall \; i \; \in S_k, v. \quad (11)$$

The objective function is the sum of the transportation costs of all direct and shared shipments at time t_1 to cluster k. Constraint (8) makes sure the amount of goods delivered to store i at time t_1 is equal to the demand of store i between time t_1 and t_2. Constraints (9) and (10) state that the amount of direct and indirect shipments are enough to deliver the required amount of goods to cluster k at time t_1. Constraint (11) makes sure the variables are integer.

The schedule of Albert Heijn has to be periodic, so the clusters and deliveries should fit a weekly schedule. To do this, Gaur and Fisher(2004) defined $P(t)$ to be the shortest path to get from t to $T + t$ and $C_p(t)$ the sum of the costs of the edges on this path. They continue by stating if they define $f(S_k)$ as the optimal delivery schedule for cluster k, they get: $f(S_k) = min\{C_p(t): 0 \leq t \leq T_0\}$. The formulation of the period IRP is defined as the following set-partitioning problem denoted P. In this formulation x_k equals 1 if cluster k is part of the solution and 0 otherwise (Gaur & Fisher, 2004):

$$min \sum_k f(S_k) x_k \quad (12)$$

such that

$$\sum_{k \in R_i} x_k = 1 \; \forall \; i = 1, \dots, n \quad (13)$$

$$x_k = 0 \text{ or } 1 \; \forall \; k. \quad (14)$$

Constraint (13) makes sure all stores are included in exactly one cluster.

In order to solve the problem presented in Section 4.2 Gaur and Fisher (2004) distinct between the case where clusters can have at most two stores and cases where more than two stores can be in a cluster. If there are just two stores allowed in a cluster Gaur and Fisher (2004) use a modified minimum weight-matching-based algorithm. This means they start with all direct shipments and will add different stores to this cluster and evaluate the costs function again. The optimal solution of this weight-matching-problem than gives the optimal solution for P. The total time complexity is $O(n^2T_0(T+T_0)^2+n^3)$ (Gaur & Fisher, 2004).

If clusters of more than 2 stores are allowed there is no algorithm available that can solve this in polynomial time. Gaur and Fisher (2004) use the randomized sequential matching algorithm (RSMA) to solve the problem. They start by having all direct shipments and the algorithm starts making clusters. In addition, the clusters will be split randomly after an iteration of the algorithm. They received good results, because their algorithm achieved the best feasible solution known in three out of eight cases.

3.4 Simulated annealing

In general CO problems can be solved in two ways: exact or approximate. For NP-hard problems we have learned that for most problems it is not yet know how to solve them exactly in polynomial. This means that only small instances of these problems can be solved within reasonable time. For example, a travelling salesman problem with five cities is solvable in a reasonable amount of time. For large problem instances it is not possible to solve this problem exactly. In the worst case, if we evaluate all the possibilities, we would have to evaluate $(n-1)!$ options. Better approaches exist; however, it has not been shown yet that there is an algorithm that can solve the general TSP within less than 2^n computation steps (Woeginger, 2003).

For this reason we have to use heuristics to come to an acceptable solution. In general we have two kinds of heuristics: heuristics that create a feasible solution and heuristics that improve a feasible solution. Table 2 shows an overview of different kind of solution finding techniques and gives a few examples of those techniques, based on Schutten (2012):

	Technique	
Exact	**Constructive heuristics**	**Improvement heuristic**
Branch-and-bound*	Greedy approach	Simulated annealing
Complete enumeration	Probabilistic approach	r-Opt (and Or-opt)
Dynamic programming	Adaptive search	Steepest Descent
		Tabu Search

Table 2: Different solution techniques *There are a lot of branch-and-bound techniques: for example Little's algorithm, Dakin's algorithm and Balas' algorithm.

We do not get into detail on all of these techniques and solution methods. The remainder of this section explains simulated annealing, because it is used by the WDScan model. Simulated annealing is an improvement heuristic and therefore needs a feasible solution to improve upon, Simulated Annealing was first introduced by Kirkpatrick et al. (1983).

The WDScan model uses simulated annealing to get a solution close to the optimum in reasonable time. Simulated annealing is based on the heating of a crystalline solid and then cool it down slowly and free of defects in the crystals (Henderson et al., 2003). In combinatorial optimization this analogy is used to allow worse solutions to be accepted to get out of a local optimum. A local optimum is a solution better than all of its neighbour solutions, but a better solution is available outside of its neighbours. Simulated annealing is able to find high-quality solutions from a random starting solution. This means the outcome is less dependent upon the chosen initial solution than other local search algorithms (Aarts & Korst, 1989).

We present pseudo code for simulated annealing as given by Henderson et al. (2003):

Initialisation:

- Create a random initial solution
- Choose: a value cooling parameter $c = c_0$, a decreasing factor α, Markov chain length k and stop criterion c_{stop}

Algorithm:

- Run this algorithm as long as $c > c_{stop}$ Execute with the algorithm k times:
    - Generate a neighbour solution ω
    - Execute transition with transition probability
- Decrease the cooling parameter c to c times α

In the beginning the algorithm accepts almost all of the neighbour solutions that are generated. However, if the temperature is near c_{stop} only solutions that are as good as or better than the objective value are accepted. The acceptance or rejection of a generated solution, for a minimization problem, can be shown in pseudo-code (adapted from Henderson et al. 2003):

Generate a solution $' \in N(\omega)$, where $N(\omega)$ is the collection of all of the neighbour solutions of ω
Calculate $\Delta_{\omega,\omega'} = f(\omega') - f(\omega)$
If $\Delta_{\omega,\omega'} \leq 0$, then $\omega \leftarrow \omega'$
If $\Delta_{\omega,\omega'} > 0$, then $\omega \leftarrow \omega'$ with probability $e^{\frac{-\Delta_{\omega,\omega'}}{c_k}}$

With c_k being the current temperature of the cooling schedule. This shows that the algorithm can accept solutions that increase the target function based on the current temperature. This is called a *hill climbing move*, because it does not yield a better objective value (Aarts & Korst, 1989)

3.5 Penalty functions

Penalty functions are used in combinatorial optimization, to penalize violation of restrictions. In general there are two basic penalty functions: interior and exterior penalty functions (Smith & Coit, 1997). Interior penalty functions penalize feasible solutions to force constraints to be tight, because optimal solutions require constraints to be active (tight). Exterior penalty functions penalize infeasible solutions. There are several ways to penalize infeasible solutions (Smith & Coit, 1997):

- Barrier methods, where no infeasible solutions are allowed
- Partial penalty functions, where a penalty is given to infeasible solutions near the feasible solutions

- Global penalty functions, where a penalty is given to all infeasible solutions

In this method the problem's difficult constraints are relaxed and the target function is modified to stay close to the feasible region. Smith and Coit (1997) give the following transformation of a target function to illustrate the penalty functions:

$$\begin{array}{lll} \min & f(x) & (15) \\ s.t. & x \in A & \\ & x \in B & \end{array}$$

In (15) x is a vector of the decision variables. $x \in A$ are constraints that are relatively easy to satisfy, while $x \in B$ are constraints that are relatively hard to satisfy. Smith and Coit (1997) reformulate the problem as:

$$\begin{array}{lll} \min & f(x) + p\big(d(x,B)\big) & (16) \\ s.t. & x \in A & \end{array}$$

In (16), $d(x,B)$ is a metric function that gives the distance between the solution vector x and region B. In (16) $p(\cdot)$ is a monotonically non-decreasing penalty function, where $p(0) = 0$. This means that when there is no distance from a feasible solution, the penalty is 0. In addition, Smith and Coit (1997) claim that if $p(\cdot)$ grows quickly enough outside of B, an optimal solution of (15) is also an optimal solution for (16). Furthermore, they claim an optimal solution for (16) gives an upper bound for (15), which is tighter than optimizing $f(x)$ over A.

Smith and Coit (1997) describe different kinds of penalty functions: static, dynamic and adaptive. Static penalty functions have a set penalty for violating a restriction, or a certain set penalty per distance from the feasible area. Dynamic penalty functions have a penalty set for a distance to the feasible solution, however this penalty increased gradually with the number of solutions evaluated. Adaptive penalty function take the length of the search and the distance to a feasible solution into account, and the success of the solution.

WDScan uses static penalty functions, because a set value for the distance from the feasible area is penalized. In WDScan penalizes for every constraint that has a penalty function. The penalty functions in WDScan are constructed similar to the formulation given by Smith and Coit (1997):

$$f_p = f(x) + \sum_{i=1}^{m} C_i d_i^{\kappa}$$

where d_i^{κ} is the distance metric of constraint i applied to solution x. When a distance matrix is used to calculate penalty costs, the choice of C_i is difficult. Richardson et al. (1989) suggests using the expected costs to repair the solution to a feasible solution. However, this might be very difficult or not possible for some problems. In WDScan we can do this for some of the constraints, where we know what the costs are of violating the constraints. For most constraints however, we are not able to give a good estimate. Smith and Coit (1997) suggest to estimate these C_i values based on the relative scaling of distance metrics of multiple constrains, the difficulty of satisfying the constraint or the seriousness of the violation. In addition, they suggest to determine the value of C_i experimentally.

3.6 Conclusion

This chapter answers the research question: *'What are similar problems to the WDScan problem in literature, and what are possible solutions for this problem in literature?*'. This chapter starts with explaining what kind of problem WDScan addresses. To explain what kind of problem WDScan is, we first discuss the general combinatorial optimization problem, followed by the vehicle routing problem. This chapter continues with explaining the inventory routing problem, a generalisation of the vehicle routing problem. Gaur and Fisher (2004) conducted a study at Albert Heijn and classified a problem similar to our problem as an inventory routing problem. The main difference between the problems is that our problem considers a costs function that evaluates the store, distribution centre, transportation and customers bothered, while Gaur and Fisher (2004) merely consider transportation optimization based on the amount to deliver to a store. The inventory model proposed by Gaur and Fisher is actually the model used by Albert Heijn currently. One of our improvements in Chapter 4 actually incorporates the inventory model into WDScan.

In addition to the positioning of our problem in literature we describe several techniques to help us solve our problem. Based on the items selected for improvement in Chapter 2, this chapter describes the simulated annealing technique and penalty functions. The next chapter uses the knowledge on penalty functions to propose improvements to the current penalty function values in WDScan.

4. The improved WDScan model

This chapter answers the research question '*How can we improve the WDScan model and what is the performance of the improved model?'.* Based on the improvements selected in Chapter 2, this chapter explains our actual improvements. The improvements discussed are: the delivery size estimation, the order picking model and the penalty costs. Each improvement section first describes the way the proposed improvement is currently modelled in WDScan. Second, the sections describe what the desired situation looks like. The desired situation is a combination of the practical situation and the literature on the topic. Third, the improvement sections elaborate on the actual improvement. Each improvement section finishes with the validation of the proposed improvement.

Section 4.1 presents the first improvement, the delivery size estimation. Section 4.2 elaborates on the second improvement, the penalty costs model of WDScan. Section 4.3 explains the third improvement to the WDScan model, the improved order picking model. Section 4.4 gives the conclusion of this chapter.

4.1 Improved delivery size estimation

This section elaborates on the delivery sizes per hour. Section 4.1.1 explains how the proposed improvement is modelled in WDScan. Section 4.1.2 states how the delivery sizes are determined in practice. Section 4.1.3 presents our hypothesis and statistical approach to find a better estimation of the delivery size. Finally, Section 4.1.4 tests the hypothesis.

4.1.1 The static delivery size in WDScan

As explained in Chapter 2 the delivery size estimation is an area for improvement of the current WDScan model. In WDScan one of the input parameters is the amount of goods (delivery size) that has to be sent to a store if a delivery is made at this hour. WDScan creates deliveries on heartbeat moments; the same delivery time window for every day a store is delivered for this flow. The delivery time window for Monday to Sunday is the same, if a delivery is made on this day. The delivery size is the same amount of goods that had to be delivered in the original DTWP. Table 3 shows a typical delivery time window plan of a store. WDScan assumes that indifferent of the new heartbeat moment, the store displayed in Table 3 gets 1031 goods delivered on Monday. Therefore, if WDScan changes the heartbeat moment of the store in Table 3 to 17:00-18:00, the amount of goods sent are the same as displayed in Table 3.

Day	Flow	Delivery time window		Total Goods	Total RCs	Goods transito	RCs transito
Monday	Fresh	10:00:00	11:00:00	1031	23	432	4
Tuesday	Fresh	10:00:00	11:00:00	940	22	375	3
Wednesday	Fresh	10:00:00	11:00:00	908	19	381	3
Thursday	Fresh	10:00:00	11:00:00	1042	23	449	4
Friday	Fresh	10:00:00	11:00:00	1464	31	653	5
Saturday	Fresh	10:00:00	11:00:00	996	21	446	4
Sunday	Fresh	10:00:00	11:00:00	996	21	446	4

Table 3: DTWP for the fresh goods of a single store

Table 3 shows the fresh DTWP for store 1172[6], which has a delivery time window on Monday from 10 am to 11 am, in between these times a delivery is made. In general, deliveries are made at the heartbeat moment. The heartbeat moment is the same Delivery Time Window (DTW) for each day, for 1172 this heartbeat moment is from 10:00 to 11:00. Table 3 shows the total amount of goods that has to be delivered to 1172 on Monday is 1031 packages; the total amount of load carriers (RCs) is 23. These are both important, because the store uses the amount of packages to know the time required for restocking, and the amount of RCs is important, because this has to fit in a certain truck. The Colli transito and RCs transito are added to the deliveries and not order picked at the distribution centres (DCs). These are required to calculate amount of goods that has to be order picked at the DCs. These are all vital to the calculation in WDScan, changing these would influence the calculation results.

The amount of goods that has to be considered when selecting a certain heartbeat moment is given per hour per store for each day in the input of WDScan. Currently, the amount of goods that has to be sent at a given day is constant. This means the ability to define the amount of goods per hour is not used. For Monday, WDScan assumes that at every given time on Monday a delivery of 23 RCs, containing 1031 packages of goods, is expected to be delivered to 1172. These assumptions result in the same amount of goods sent on a day, independent of the chosen heartbeat.

4.1.2 Delivery size in practice: the replenishment model

In practice, the size of a delivery is based on what is expected to be sold in the store until the next delivery is made. For every single item in every store it is determined how many items are expected to be sold in the period until the next delivery arrives at the store. The amount of goods is determined when the order is created, about 18 hours before the delivery to a store. For each single item, it is determined how many customers would buy this item per 1000 customers. Multiplied by the expected amount of customers, a demand for this time period is calculated. Based on the demand the choice is made for each item whether or not to send a package of goods. The choice to send a package of goods is based on the following principles. A package of goods of a certain product is sent if (Albert Heijn, 2014):

1. It fits on the shelf; the stock at the restocking moment plus amount of goods in a package has to be less than shelf capacity, or
2. It is demanded; the amount to be sent plus the stock at the restocking moment has to be more than the demand, or
3. It is required; in order to achieve certain availability, i.e. not to have an empty shelf.

If one of these conditions is met, the system orders a number of packages. The number of packages is calculated for all products of the flow for a delivery and the sum of the amount of packages makes a delivery. These deliveries are triggered by what is expected to be sold, therefore the amount of customers seems to be a good indicator of the size of a delivery. In addition, the SCCP department suggested using the amount of customers to estimate the amount of goods to deliver when choosing a

[6] A fictive store number is used, to illustrate the case.

certain heartbeat moment. The SCCP department uses the same rules when estimating what the volume change will be when changing a heartbeat moment for a certain store.

4.1.3 Customer patterns to estimate delivery size

In order to estimate the delivery size of a delivery when changing the heartbeat moment, we suggest determining the delivery sizes based on customers per delivery. This approach seems to be a good approximation of the replenishment model, explained in Section 4.1.2. We are not able to simulate the replenishment model for all of the individual products, because of a lack of data and the huge computation time. A better estimation of the delivery size is required, because the amount of customers per day is different over the days. In addition, the peaks in amount of customers also vary over the days. Because the amount of customers and customer patterns change over days, using the delivery size of the original heartbeat (as explained in Section 4.1.1) does not suffice.

The data available is the average amount of customers per hour and the average amount of total packages delivered at time *t*. Based on the data available, we determine for each delivery the amount of customers that are 'supplied' by this delivery. We distinguish between different flows of deliveries. Figure 10 illustrates the amount of customers that are considered for a certain delivery. The customers that are considered for a delivery are the customers that are in the store during the period of three hours from the delivery to three hours from the next delivery, because three hours from the delivery the goods are available at the shelves. The decision to send a package is based on the customers as explained in Section 4.1.2. Figure 10 shows an overview of the customers considered by our model to calculate what customers are served by this delivery. Our assumption is that amount of goods per customer can be used to estimate the decisions of the replenishment model, explained in Section 4.1.2.

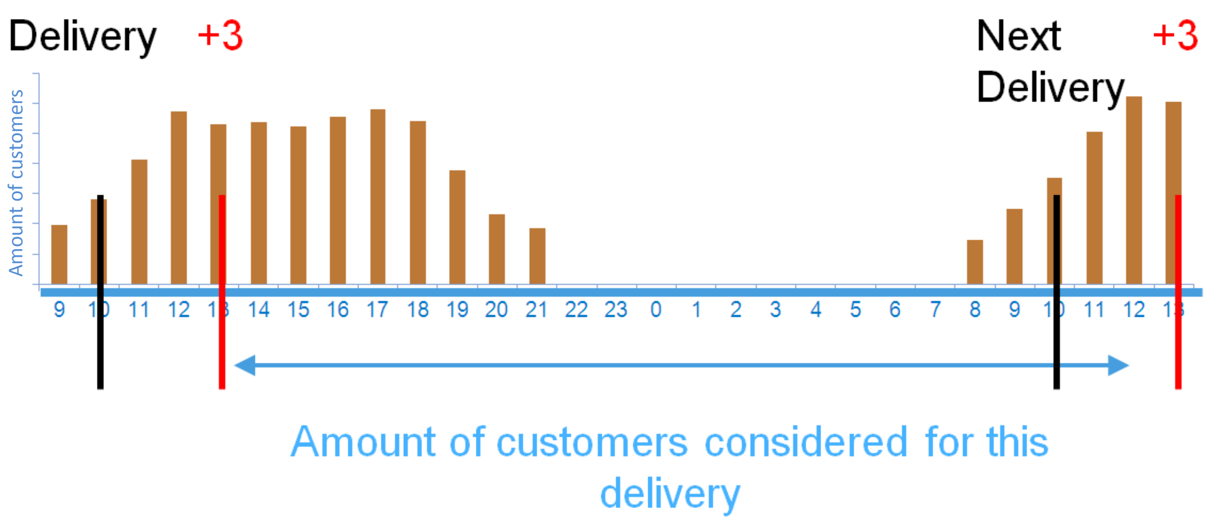

Figure 10: What customers are considered for a delivery in our model?

Our hypothesis is that based on the amount of packages per customers for a given hour a good estimation of the delivery size can be calculated for a change in delivery time. To calculate the packages per hour the total packages of this delivery and the total amount of customers are used. Figure 11 schematically shows the calculation of the packages per hour. We assume the amount of packages per customer to be fixed per delivery. When changing the heartbeat moment, the amount of packages per customer is changed. The change occurs, because partly the packages per customer of the previous or latter delivery are used, for the hours that have shifted from the current delivery. Figure 11 shows what a change of heartbeat moment does to the packages considered for a delivery.

$$\#\, packages for\ this\ hour = \frac{total\ packages\ for\ this\ delivery}{total\ customers\ for\ this\ delivery} \times customers\ for\ this\ hour$$

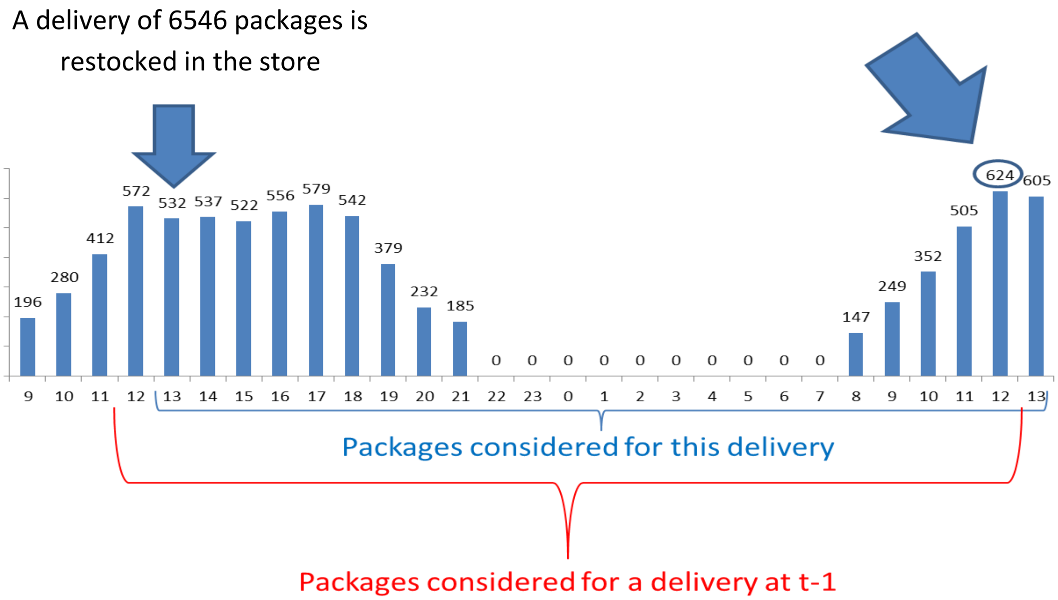

Figure 11: Calculation of packages per hour

4.1.4 Validation of the WDScan replenishment model

In order to assess the validity of using packages per customer for a delivery when shifting a heartbeat moment delivery, we use the following approach. If we find similar stores that have different heartbeat moments but similar customer patterns, we expect them to have the same normalized delivery size (percentage of weekly amount of packages) when changing all their heartbeat moments to a single moment. For example, if two stores now have a heartbeat moment of 5:00 and 17:00 we calculate the delivery sizes when shifting both heartbeat moments to 13:00. If these stores have similar customer patterns we expect that they receive the same normalized amount of packages. We test this by first selecting stores that have similar customer patterns. For these stores we calculate the normalized amount of packages for every hour of the week based on their original normalized delivery sizes. We test the hypothesis whether the normalized delivery sizes look similar. Because one heartbeat has seven deliveries (one each day) we assess the correlation between the 7 data points of a store and the 7 data points of another store. If these have a strong and significant correlation, they have a strong relationship. Because we use normalized data points, the sum of the weekly volume is the same for all the stores by definition. If a strong relationship is observed, the normalized delivery sizes are very similar and we can conclude our estimation is a good one.

We select ten stores to test our hypothesis. Appendix B shows the cluster analysis to find stores that have similar customer patterns. Appendix C shows the selection of the ten stores of this cluster based on similar results of normalized delivery size estimation using the method explained in Section 4.1.3. We select only the fresh goods flow for comparison, because ambient goods are less customer driven. For ambient goods the replenishment model sends packages in advance, while they are not yet required for sales. Sending packages in advance is not possible for fresh goods, because they perish over time.

The ten stores selected all have a different heartbeat moment, as shown in Table 4. In this table, the store numbers and their heartbeat moment are displayed. Because the heartbeat moment is an interval of an hour, we have displayed the start and end of each window. We have chosen to set the heartbeat moment of all of the selected stores to the heartbeat moment of store A: 16:00-17:00. The correlation shown in Table 4 is the correlation between the calculated expected delivery sizes when setting the heartbeat moment to the heartbeat moment of store A. This gives us 7 data points per store to calculate the correlation. Table 4 shows that for all but one store, the Pearson's correlation coefficient is significant. A Pearson's coefficient higher than 0.7 indicates a very strong (positive) correlation. For example the delivery sizes of store B, when setting the heartbeat of store B to the heartbeat of store A, have a 97% correlation.

	Store	A	B	C	D	E	F	G	H	I	J
OLD heartbeat	**Begin**	16	4	5	21	3	13	18	10	6	7
	End	17	5	6	22	4	14	19	11	7	8
NEW heartbeat	**Begin**	16	16	16	16	16	16	16	16	16	16
	End	17	17	17	17	17	17	17	17	17	17
Correlation stores' delivery sizes		1.00*	0.97*	0.92*	0.83**	0.92*	0.96*	0.88*	0.88*	0.74	0.89*

Table 4: Correlation between delivery sizes of stores, when selecting the same heartbeat moment *=significant at α=0.01, **=significant at α=0.05

This shows that, independent of the original heartbeat moment, the normalized delivery sizes are similar. In order to give a good estimation of the volume change when changing a DTW, this should hold for stores that have a similar customer pattern. Because nine out of ten have a very strong correlation and are statistically significant at $\alpha = 0.01$, we conclude that using the amount of customers to estimate the delivery size change for the fresh goods is a good estimation. We conclude based on this data that the delivery size estimation is a good estimation for the delivery sizes in normal weeks. This method can be used for the busy weeks, because in general the busy weeks only have larger volumes. However, using the current way to determine the DTWP for the busy weeks, as explained in Section 2.1, a store can have different delivery time windows over the days. WDScan only allows one heartbeat, therefore this method works when determining the DTWP using WDScan. The variation in the correlation can be explained, because of the (slight) differences in customer patterns. Variation in the correlation could also be caused by other store aspects, for example the shelf space or transactions per m^2.

4.2 Improved penalty costs model

The second proposed improvement concerns the penalty costs. This section describes the way the penalty costs are currently modelled in WDScan. Based on literature, we state how to assess the penalty functions. The last part of this section describes the validation of the new penalty functions.

4.2.1 WDScan penalty costs of the current DTWP

Penalty costs are a major part of the target function, as explained in Section 2.2.3. This section shows what these penalty costs are, and how they are constructed. Penalty costs are the costs of a certain violation, for example not enough time between two deliveries, times the penalty for this violation. In the case of the time between two deliveries, the amount of minutes this constraint is violated is multiplied by the costs for this penalty. These costs are divided into the following penalty costs in the WDScan Model (ORTEC, 2013):

- Infeasible DC Closed Loading, a penalty of € 3,000 for each trip that leaves the Distribution Centre (DC) outside of the opening hours of the Distribution Centre loading time of the DC
- Infeasible DC Closed Production, a penalty of € 3,000 for each trip that has its order picking done outside of the order picking opening hours of the DC
- Infeasible Store Interval Time, a penalty if two deliveries to the same store are too close to one another. For two deliveries of the same flow, four hours are required between two deliveries. For two deliveries that are not of the same flow, two hours are required. Violating this restriction gives a penalty of € 40 per minute violation.
- Infeasible Truck Volume, a penalty for using more than the capacity of a truck per RC over the capacity a penalty of € 1,000 is inflicted.
- Infeasible Store Closed, a penalty for if the start restocking is after the store is closed. Per store a parameter for the time after closing restocking is allowed. The penalty to start restocking after the allowed time is € 50 per minute.
- Infeasible DC Trip Departure, a penalty of € 1,000 when less than a given threshold value or more than the capacity of the DC amount of trucks depart from the DC for all the hours a trip is made.

The above mentioned penalties are the hard penalty functions in WDScan. In addition, some penalty functions are used to flatten the amount of order picking, trucks and departures at a DC. These penalty functions are called soft penalties. The soft penalty functions of WDScan are:

- Infeasible DC Prod Spread, a penalty of € 10,000 per roll cage (RC) outside the given bandwidth for a set amount of hours (parameter). The average over these hours is calculated and all the RCs outside of the given bandwidth result in a penalty of € 10,000 per RC. These penalties are calculated for all of the flows, and the sum of both of the flows is calculated.
- Infeasible Truck Type Spread, a penalty of € 10,000 per truck outside the given bandwidth for a set amount of hours (parameter). The average over these hours is calculated and all the trucks

outside of the given bandwidth result in a penalty of € 10,000 per truck. These penalties are calculated for all of the truck types, and the sum over all of the trucks is calculated.

In the current DTWP evaluated by WDScan, the penalty costs constitute about 2/3 of the objective function value. Table 5 gives an overview of the penalty costs and their values for the current DTWP. Notice that especially the soft restrictions have a very high penalty value in the current DTWP. In general, we do not want to violate hard penalties. However, in our current DTWP some hard restrictions are violated. In addition to the classification of the penalty functions, Table 5 sums the costs of the penalties in euro and amount of violations. The number of violations are the amount of stores that have one or more violations. Because the soft restrictions are not based on a violation per store, we do not have data on the amount of store violations.

Penalty	Soft/ Hard	Store/ Overall	Penalty (€)	Unit	# of violations
DC Closed Loading	Hard	Store	3,000	Per trip	7 stores
DC Closed Production	Hard	Store	3,000	Per trip	29 stores
Store interval time	Hard	Store	40	Per minute	48 stores
Truck Volume	Hard	Store	1,000	Per RC	29 stores
Store Closed	Hard	Store	50	Per minute	7 stores
DC Trip Departure	Hard	Overall	1,000	Per trip	n/a
DC Prod Spread	Soft	Overall	10,000	Per RC	n/a
Truck Type Spread	Soft	Overall	10,000	Per truck	n/a

Table 5: Penalty overview for current DTWP

The calculation of the current situation of Albert Heijn by WDScan results in the penalty costs as shown in Table 5. The majority of the penalty costs are based on the soft restrictions. This suggests that the current schedule is within the limits of the hard restrictions, or that the penalties of the hard restrictions are not that high in comparison to the soft restrictions. The soft restrictions' penalties are a large part of the total penalties in the current model. The goal of WDScan is to optimize the current DTWP to get a better overall plan. The reason to have penalty costs is to temporarily relax the problem's most difficult constraints (Smith & Coit, 1997). ORTEC introduced these penalties, because the current DTWP did not yield a feasible solution (Spoelstra, 2014). This means the current DTWP did not satisfy the constraints of the WDScan problem formulation. Section 4.2.3 elaborates on how to improve these penalty functions, or the restrictions they are based on.

4.2.2 Penalty costs when optimizing in WDScan

This section explains what the effect of the chosen penalty functions is on the optimization process of WDScan. Section 4.2.1 showed that in the current DTWP the penalty costs account for about 2/3 of the total objective function. Based on the current DTWP, WDScan makes changes to this plan and evaluates the target function. These changes are neighbour solutions of the current DTWP. A neighbour solution is a change in the heartbeat moment (the delivery time throughout the week) of a single store. For example, store 1172 as described in Section 4.1.1 has its heartbeat moment between 10 and 11 in the morning. A change neighbour solution could be a heartbeat moment between 11 and 12 in the morning. After selecting a neighbour solution, the target function is re-evaluated and the change is accepted or not (based on the simulated annealing settings and stage).

Apparently, changes to the plan that reduce (the most expensive) penalty costs are accepted, because they reduce the value of the target function most. Figure 12 shows the target value parts after a run of WDScan. The value of the target function is reduced by more than 25% by a run of WDScan. The penalty costs have been reduced most, however the real costs have grown. For example, the store costs increase by over 8% in the improved DTWP.

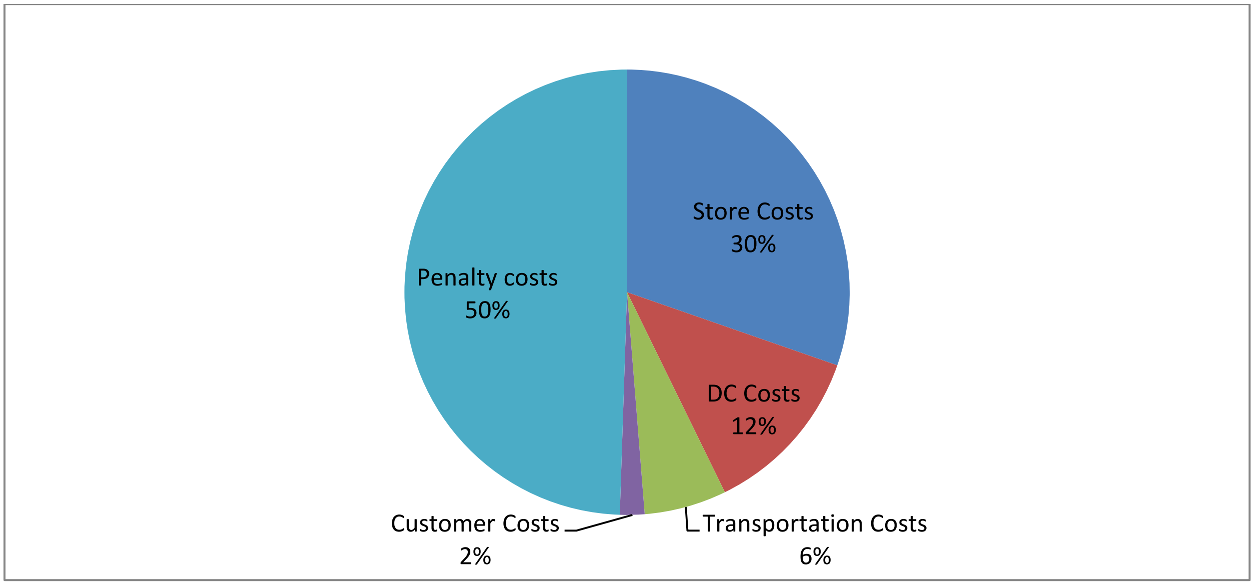

Figure 12: Target function value after a run of WDScan

The total objective value decreased after optimizing and the soft penalty costs have been reduced. However, investigating the penalties shows what has actually happened. The decrease in the penalty costs is mostly because of the reduction of the soft penalties, while the hard penalties are increased in costs and amount of stores affected. Table 6 gives an overview of the penalty costs of the current DTWP and the DTWP after optimization by WDScan. This table confirms that a shift from soft penalty costs to hard penalty costs. Some penalties are increased by as much as 300% in costs and stores affected. However, the soft penalty costs are decreased by up to 61%.

Penalty	Soft/ Hard	In- or de-crease	# of violations (DTWP)	# of violations (WDScan)	In- or de-crease
DC Closed Loading	Hard	+321%	7 stores	29 stores	+314%
DC Closed Production	Hard	+138%	29 stores	42 stores	+45%
Store interval time	Hard	+184%	48 stores	110 stores	+129%
Truck Volume	Hard	-100%*	29 stores	0* stores	*
Store Closed	Hard	+750%	7 stores	9 stores	+29%
DC Trip Departure	Hard	+100%	n/a	n/a	n/a
DC Prod Spread	Soft	-48%	n/a	n/a	n/a
Truck Type Spread	Soft	-61%	n/a	n/a	n/a
Total penalty costs		-46%	70 stores	114 stores	+62%

Table 6: Comparison current DTWP to WDScan optimization *Set as an actual hard restriction in WDScan, 0 by definition

The target function value decreases, however an infeasible solution is found. The solution is not feasible because a lot of hard constraints are violated. As Section 3.5 explains, penalty costs should help get to a feasible solution by relaxing the most difficult constraints (Smith & Coit, 1997). WDScan moves further from the feasible region, because it violates more hard constraints and less soft constraints. Section 4.2.3 explains how we improve the penalty functions to stay closer to the feasible region.

4.2.3 The improved hard penalty costs

There are two major issues with the current penalty costs: the hard penalty restrictions are violated a lot after there is a huge DC order picking spread penalty. In order to solve these issues we propose two improvements: the improved hard penalty costs and an improved order picking model. An improved order picking model makes sure the DC Closed production is feasible for the current DTWP. In addition, it removes the large DC Prod Spread penalty, because this becomes redundant. Reducing the DC Prod Spread penalty is not enough to create a reliable order picking model. Section 4.3 elaborates on the improvements to the order picking model. This section explains how to improve the penalty functions of the hard constraints that are difficult to satisfy.

The hard constraints that have a penalty function are the hard constraints that could not be satisfied in the current DTWP. As explained in Section 3.5, there are several ways to model penalty costs. Examples are static, dynamic and adaptive penalty functions (Smith & Coit, 1997). The WDScan penalty functions are dynamic and therefore based on the distance from the feasible solution. However, as shown in Section 4.2.2 the penalties for the hard constraints are too low, because they are favoured over other costs. Favouring violating soft restrictions over hard restrictions results in an infeasible solution. Smith & Coit (1997) argue that if a penalty function is too gentle, they can result in a final infeasible solution. However, if a penalty function is too tough this results in a non-optimal feasible solutions. We therefore propose a method to tune the penalty functions specifically for this problem.

The easiest way to get feasible solutions is by inflicting severe penalties for infeasible solutions. However, simulated annealing benefits from not too tough penalty functions, because it less likely to accept a solution when the evaluated neighbour solution's value is a lot worse. If a neighbour solution is worse than the current solution simulated annealing accepts the solution based on a probability function that includes the current temperature of the iteration and the difference in target value. We propose penalty functions gentle enough to select infeasible solutions at the start of simulated annealing. However, the penalty functions should be severe enough to not allow infeasible solutions to be selected near the stop temperature of simulated annealing.

The penalty functions in WDScan are dynamic penalty functions, where the amount of violation of a constraint is multiplied by a value. We find an upper bounds for these values in the current settings of WDScan. However, if the real costs of WDScan change, the penalty values have to be re-evaluated. The upper bounds have to be changed, because if real costs increase, a penalty function just severe enough, might be not severe enough anymore. The point where the penalty costs are just severe enough is determined by assessing when these hard penalty restrictions are violated in the solution WDScan provides. We advise using the value just above this point. Section 4.2.4 elaborates on the actual upper bounds found, and the impact on the target function of WDScan.

4.2.4 Upper bound determination for the hard penalty functions

This section describes the upper bounds of the penalty costs in WDScan. If hard restrictions are violated in the current situation, the restriction is incorrect or the current DTWP is infeasible. Using the current model we can impose high penalties so choosing these penalties will only occur when it is not possible to avoid them. The general idea is that Albert Heijn does not want violate to certain restrictions, except when there is no other choice. An example of such a restriction is the time between two deliveries, that has to be at least 2 hours. However, for some stores it is not possible to have 2 hours between deliveries. For these stores we can change the restriction to a smaller value according to their situation. This makes sure we do not incline extra costs by default. The penalties under investigation are the hard penalty costs as explained in Section 4.2.1. The DC order picking while the order picking area is closed we do not take into account, because this restriction is no longer required in the new order picking model. In addition, we do not take the Truck volume penalty into account, because WDScan sees this as a hard restriction which cannot be violated. Therefore the amount of violations is by definition zero. The remaining penalties under consideration are:

- Infeasible DC Closed Loading
- Infeasible Store Interval Time
- Infeasible Store Closed
- Infeasible DC trip departure

For the above mentioned penalties we provide an upper bound for the current model. As explained above, we do not consider the DC order picking penalties. These penalties will change when implementing a new order picking model, explained in Section 4.3. In order to get an improved upper bound we increase the penalty function until these penalties were not violated anymore. We increase each individual penalty function, until it was not selected over real costs. The technique we use for evaluation penalty functions is as follows. Set a high penalty value, to assess what stores violate the restriction. If the restriction is violated by some stores, assess whether the restriction has to be changed and if needed, change it. To determine this high value, we use a rough method where we increase the penalty value by a factor 10 until only the restrictions remain that have to be violated. The results of the calculations using an improved upper bound can be found in Table 7. For all of the hard restrictions that are not improved by the order picking model we calculated the upper bound. There is one exception, the DC Closed loading penalty we could not increase. This has been noted to ORTEC as a bug.

Penalty	Soft/ Hard	Penalty value WD Scan	Penalty value improved penalty	# of violations (WDScan)	# of violations new penalty	In- or de-crease
DC Closed Loading	Hard	4000	n/a*	29 stores	9 stores*	-69%
Store interval time	Hard	40	4000	110 stores	0 stores	-100%
Store Closed	Hard	50	500	9 stores	0 stores	-100%
DC Trip Departure	Hard	100,000	1,000,000	n/a	n/a	n/a
Total of selected penalty costs				114 stores	9 stores	-92%

Table 7: New penalty functions *= due to an error in WDScan, this could not be put as high as required

The evaluation of the Store Interval Time penalty is a good example of the above mentioned technique. We set the penalty value from 40 to 4000 per minute of violation. Only the restrictions where no other option is available are violated, as seen in Figure 13. For the Store Interval Time penalty the violations are stores that have a narrow available to delivery window. For example, store 1547 can only be delivered between 7:00 and 10:00, however 3 trucks have to deliver goods to this store. No solutions are possible where 2 hours between the deliveries is achieved. Therefore, we propose to change the restriction of the hours required between deliveries from 2 hours to $min\left\{2, \frac{available\ to\ deliver\ window}{number\ of\ deliveries - 1}\right\}$ hours. For 1547 the restriction becomes $min\left\{2, \frac{3\ hours}{3\ deliveries - 1}\right\} = 1.5\ hours$. The new restrictions make sure a solution is possible where no penalty costs incur for store 1547 and the largest time between deliveries is achieved. We use the same technique for the other hard penalty costs of WDScan. Please note that Table 7 does not use the same data as Figure 13, as Figure 13 merely illustrates the determining of the upper bound for the Store Interval Penalty.

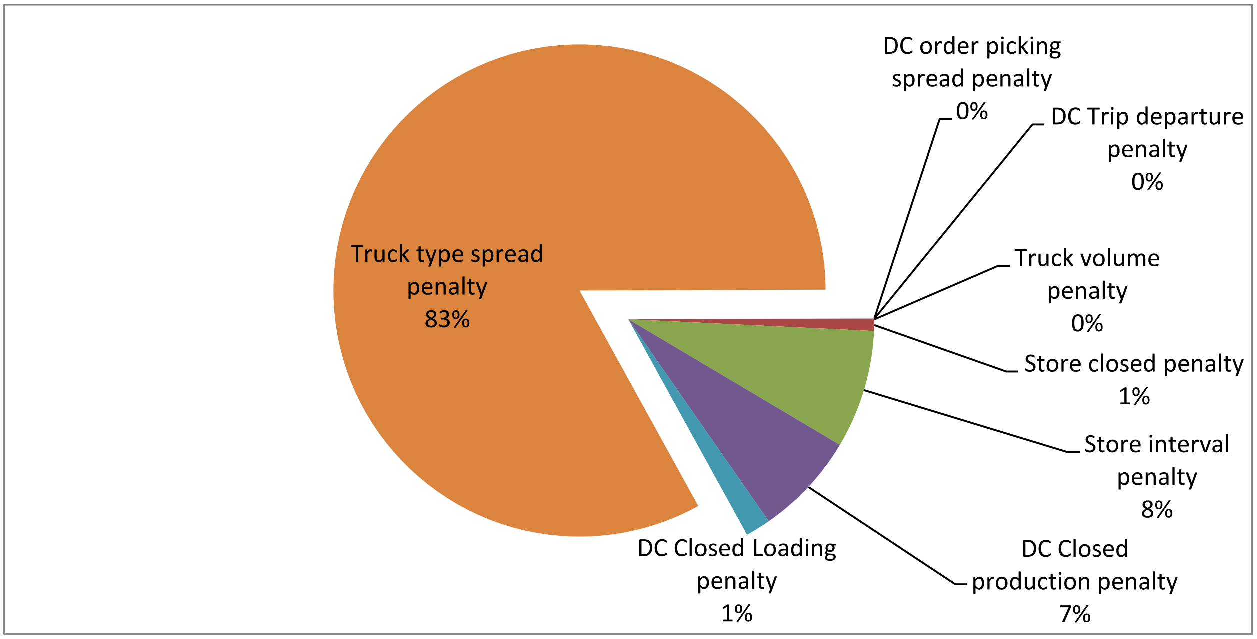

Figure 13: Determining the upper bound of the store interval time penalty

Figure 13 shows the penalty costs parts of WDScan using our new penalties, before changing the store interval restriction as described above. This figure shows the truck spread penalty is the biggest part of the penalty costs now. The total penalty costs have been decreased by 90% by removing the DC production spread penalty and adjusting the hard penalty costs. However, using high penalties for the hard restrictions as shown in Table 7 costs has an effect on the actual costs and customers bothered. With the new penalty costs, we force schedules to be feasible while not taking into account the order picking model. WDScan is less likely to find better solutions, because of these high penalty values. We observe that WDScan is less likely to find better solutions in the actual costs part of the target function. The actual part of the target function has grown by 7% using the new penalty functions. Note that some constraints are violated in the current DTWP, therefore the current DTWP could be less expensive. We propose to improve these penalty values by assessing at what point the restriction is not violated and therefore has a value of 0. Assessing the improved penalty values using the current order picking model would not yield results that are applicable after changing the order picking model. The value of the penalty functions should therefore be re-evaluated after improving the order picking model.

4.3 Improved order picking model

This section describes the improved order picking model. Section 4.3.1 concentrates on explaining the order picking model used in WDScan. Section 4.3.2 focusses on the actual order picking model used in the distribution centres of Albert Heijn and explains the difference with the WDScan order picking model. Section 4.3.3 explains the improved order picking model. Finally, Section 4.3.4 explains why Albert Heijn should implement the improved order picking model.

4.3.1 Order picking in WDScan

The order picking model in WDScan is based on just-in-time order picking. This means the order picking is an immediate predecessor of the checking of the shipment, being a predecessor of the loading of a truck at the distribution centre. In addition, WDScan mimics the capacity by restricting the amount of trucks that leave the DC every hour. The flow of an individual store order in the distribution centre is schematically shown in Figure 14.

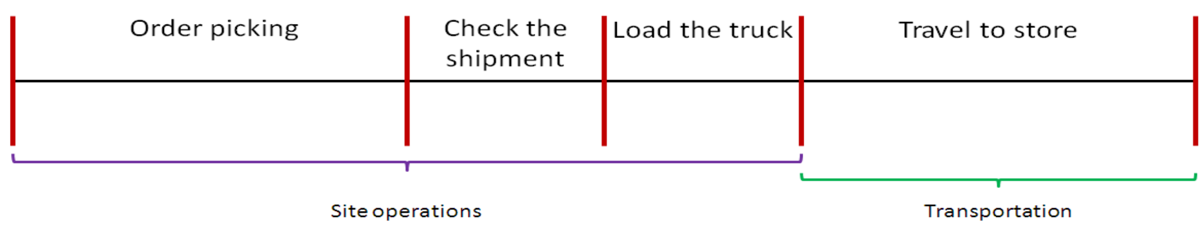

Figure 14: Distribution Centre process, adapted from ORTEC (2013)

The total amount of order picking in the distribution centre has to be within a certain bandwidth to avoid huge peaks in the order picking process. If the amount of order picking for a certain hour is outside this bandwidth a penalty is given. Figure 15 displays the order picking WDScan schedules, when evaluating the current (manually generated) DTWP. The order picking amounts per hour WDScan schedules, vary severely from hour to hour.

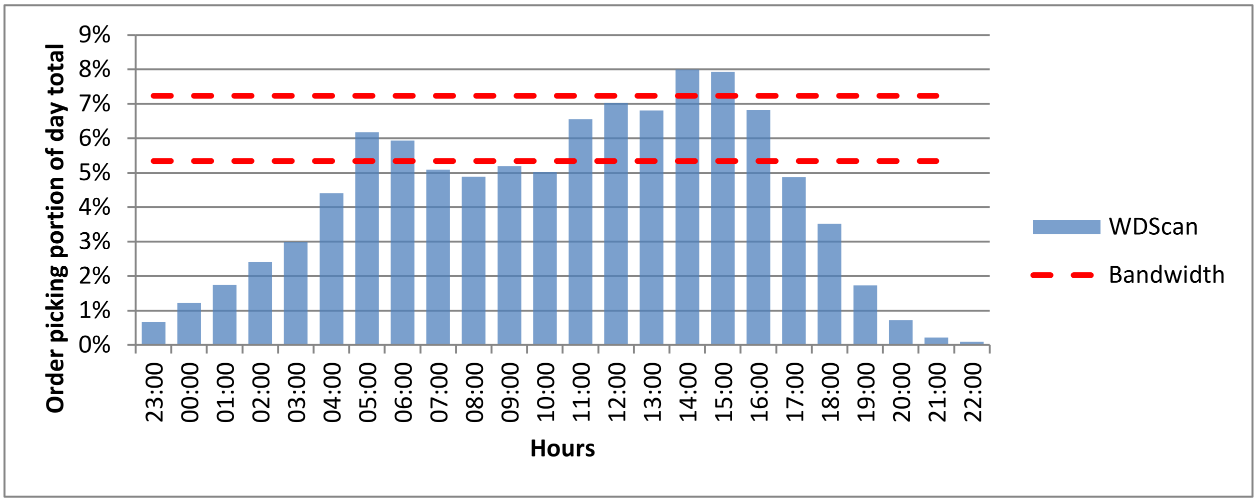

Figure 15: Order picking amounts in WDScan

The penalty for order picking more than a certain bandwidth outside of the average is modelled to make the model act like the current situation. Figure 15 shows that the current DTWP calculated by WDScan is not within this bandwidth. The amount of order picking outside of this bandwidth is calculated over the hours between 5:00 and 17:00, the default setting in WDScan.

WDScan uses the following method to mimic the current order picking model:

- The amount of load carriers(RCs) to be picked in an hour should be within a certain bandwidth of the average RCs to be picked over specified hours; if not, a penalty imposed
- A maximum amount of trucks can leave the DC per hour
- The order picking process is a predecessor of the loading process of a truck, with a fixed amount of checking of the shipment in between.

4.3.2 The order picking model in practice

The way WDScan models the order picking is just-in-time, based on the departure of the truck from the distribution centre. However, in the current situation at the distribution centres, order picking is done in advance, and buffer capacity is used to facilitate this. WDScan uses the amount of outgoing trucks per hour to model the capacity restrictions of the amount of trucks. However, the current models assesses if the proposed DTWP is executable (Albert Heijn, 2014). This model does not take the amount of trucks that have to leave the DC into account. We propose a new method to model the order picking process and the DC capacity problem accordingly.

The current situation in the distribution centres is that two shifts of order pickers, one shift during the night and one during the day, pick the orders. This means there is a certain fixed capacity based on the amount of employees in these shifts. Contrary to the order picking model in WDScan, a certain buffer capacity is available to store orders until the orders have to be loaded. This buffer capacity is actually the space at the docks where the outgoing goods are waiting to be loaded. This gives a certain flexibility to order pick in advance. In general, the first shift starts at 23:00 to pick orders for the following day. The first deliveries have to leave the distribution centre at about 2:00 in the morning. This means that during the first hours the space at the docks is filled with goods, which leave the DC at a later point in time. The order pickers' last shift ends at 16:00, however a lot of deliveries to stores leave the DC after 16:00. Departing after 16:00 is possible, because the deliveries are placed at the docks in advance.

The order picking process in WDScan does not work as described above. WDScan assumes the order picking process for a delivery is a direct predecessor of the loading of the truck. Figure 16 illustrates the order picking amounts difference per hour between the WDScan model and the current situation as a percentage of the total amount to be order picked.

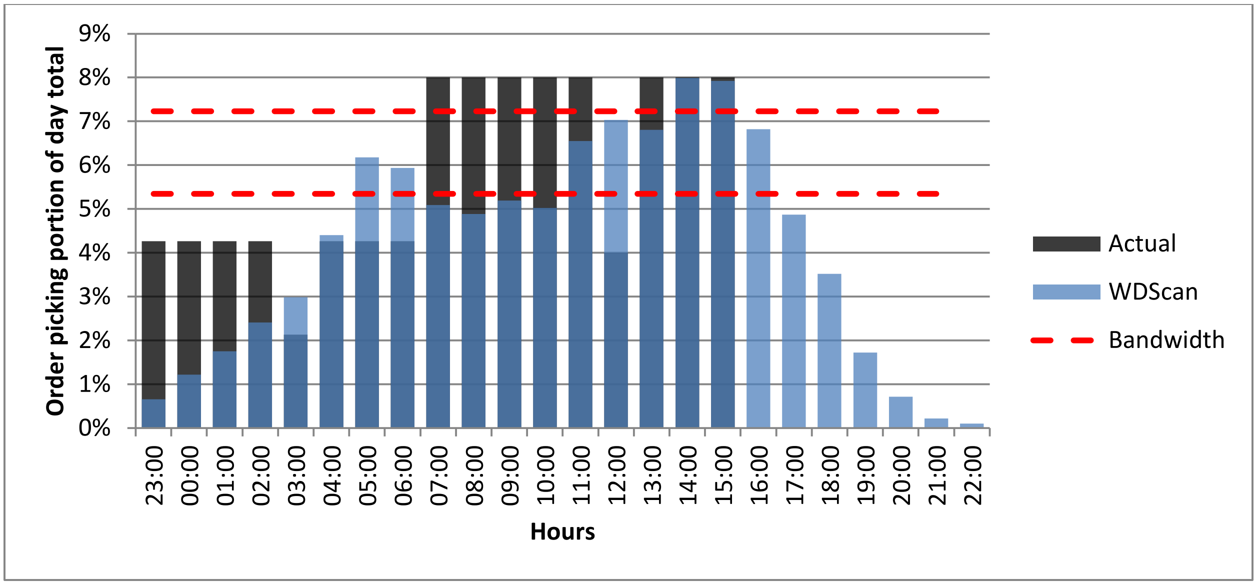

Figure 16: WDScan order picking amount v actual order picking amount

WDScan schedules a lot of order picking after 16:00, where in the current situation no order picking is done. WDScan does not facilitate order picking in advance, because the buffer capacity is not taken into account. In the current situation order picking in advance is common practice, therefore no order picking is scheduled after 16:00. Order picking in advance is the essential difference between the order picking in practice and the way WDScan models order picking. The lower values of the current situation at 3:00 and 12:00 are caused by the brakes in the shifts.

To summarize, the biggest difference between the current method and the WDScan order picking model is as follows. The WDScan model does not facilitate order picking in advance. In the current situation at the distribution centres order picking in advance is common practice. This allows a lot of flexibility, because buffer capacity is available to buffer orders before they have to be loaded into a truck. WDScan does not have this flexibility but has to keep the order picking amount between certain boundaries. This means a lot of feasible solutions will get high penalty costs when evaluated in WDScan, as shown in Section 4.2.

4.3.3 The improved order picking model

In order to facilitate the current order picking process in the distribution centres of Albert Heijn in WDScan we develop an improved order picking model. This model is based on the evaluation method used by Albert Heijn for the busy weeks (BWs) as explained in Section 2.1. This model assesses whether the proposed DTWP can be executed by the distribution centres. It assesses whether the amount of goods can be order picked before the truck has to be loaded. In addition, the model checks whether the buffer capacity constraint at the loading docks is not violated and whether the amount loaded is not above the maximum order picking capacity. However, this model is an excel model, where the supply chain specialist has to enter a certain order picking amount per shift. This order amount is a percentage of the maximum order picking capacity. We propose that this model replaces the current order picking model in WDScan. When using this model, the DC closed order picking penalty is no longer required. Instead, a penalty for using more buffer capacity than available is introduced.

The improved model uses the amount of goods that has to be loaded for each quarter. The trips that have to be loaded in this quarter are calculated by WDScan based on the DTWP. We assess for each day how many goods have to be loaded at each quarter. For each quarter we calculate the sum of the goods that has to be loaded until that point in time. The goal of the model is to see how many employees we need to make sure it is possible to order pick the goods before they have to be loaded into the trucks. To calculate the amount of employees needed, we calculate the amount of goods order picked in a certain quarter, by multiplying the amount of employees in this quarter times their productivity in amount of goods per quarter. The result is the total amount of goods that are order picked up to all quarters.

We calculate for each quarter whether or not the amount of goods picked is sufficient to supply the total amount of goods that has to be loaded and does not exceed the buffer capacity (loading docks at the DCs). The case at Albert Heijn has two shifts, one during the night and one during the day. We present a more general model, able to cope with more than two shifts. However, our solution method is only the optimal solution (under the assumptions mentioned in Section 4.3.3.1) when a certain condition is met. We first describe the model, continuing with our solution strategy.

4.3.3.1 Assumptions and model formulation

The improved order picking model has two general assumptions. The first assumption is that the productivity per order picker is constant over the shift (or at least a certain known value for each hour of the shift). The second assumption is that if it is possible to order pick a certain amount of goods, these picked goods are available to load into the trucks the next moment in time. The amount of goods that has to be loaded at certain times are input. However, these can vary because of changes in the amount of store orders at the actual day. The intention of this model is not to provide an order picking schedule, but rather assess at tactical level whether a certain DTWP is feasible. The hours of a shift are also considered input, this means the total amount of labour hours are set. The goal of the model is therefore to find the best weighted average of the costs. We argue that the hours of each of the shifts should be modifiable in WDScan, in order to assess what the impact of a change of the amount of shifts or shift hours would have.

To illustrate the model, assume there are two shifts, one during the night and one during the day. The night shift has a costs of 20 euro per employee per hour, while the day shift has a costs of 10 euro per employee per hour. In this case, order picking most in the day would be best, because of the lower costs per hour. However, all the orders have to be picked before they are loaded. This means the orders that have to be loaded during the night shift have to be order picked during the night shift. In addition, the orders that have to be loaded at the start of the day shift have to be picked during the night. The case where our distribution centre, for example, can facilitate at most 100 order pickers we would have to assess (at most) all of the possible combinations of using 0 to 100 order pickers in each of the shifts.

Complete enumeration, assessing all of the possible solutions, would give $(n+1)^m$ combinations to evaluate. Where *n* is the maximum amount of order pickers allowed and *m* is the number of shifts. Doing complete enumeration for two or three shifts is doable in reasonable computation time. However, when using complete enumeration for more shifts, the computation time grows exponentially. Each combination should be evaluated to meet the restrictions. These restrictions are:

- The amount order picked up to time *t* should be more than or equal to the amount of goods that has to be loaded up to time *t+1*
- The amount order picked at the end of a shift minus the amount of goods loaded up to the end of the shift should be more than or equal to the amount of goods loaded up to the first 1.5 hours of the following shift. This constraint mimics the lead time in the distribution centres. This next shift could start at the end of the previous shift or there could be multiple hours between these shifts.
- The amount of goods waiting over the dock capacity should be minimized for the given DTWP. Because certain DTWP cannot be facilitated by the DC, we do not want the amount of goods waiting over the dock capacity as a hard restriction. In addition, when using our solution method we start with an infeasible solution for the amount of goods at the docks.

With each combination these would have to be evaluated for each time t of the day of the shift. This means that for every evaluation at least $3t$ calculations are required. The output of the model are the total order picking costs and if the amount of dock space used is higher than the given capacity, the penalty for this DTWP. The model assesses if the given DTWP is feasible, what order picking costs would incur. If the assessed DTWP is not feasible, it provides the order picking costs and a penalty for violating the capacity at the docks. This penalty is the amount of goods that are over the capacity times the time it is over the capacity. The weight of this penalty function should be assessed the same way the other hard restrictions penalty weights are assessed, as described in Section 4.2. However, in order to assess this, the model should be implemented in WDScan first.

4.3.3.2 Solution method

As described in the former section, using complete enumeration would yield a lot of calculations when a lot of shifts are used. Currently Albert Heijn uses two shifts in the distribution centres. However, this could change over time. We propose a different solution method to avoid complete enumeration. Our solution method is based on starting with the maximum order picking amount and gradually decrease this by one order picker. We start with the shift that has the highest costs per hour. This solution method takes at most $n \times m$ evaluations of the restrictions, because there are at most n options per shift, for each shift.

However, for two shifts the maximum amount of evaluations of our solution method is equal to the complete enumeration option. If we take into account that we order pick exactly the amount required for a given day, the chosen value in shift 1 has a corresponding value in shift 2. Their values are linked, because the goods that are not order picked in shift 1 have to be picked in shift 2, and nothing more. For two shifts it seems reasonable to use the complete enumeration method. Although, there is another difference: in the proposed solution method we do not have to evaluate a target function. We do not have to compare different outcomes. We only have to assess what the feasible solution with the least order pickers in the most expensive shift is, for all shifts. In addition, our improved model starts from the maximum capacity and decreases the amount of order pickers. Our improved model is favourable over the old model, because the utilization of the shifts is about 75% of the maximum capacity on average. This means we only have to assess about 25% of the solution space on average.

We evaluate the restrictions after each decrease in order pickers. Figure 17 schematically shows the improved approach. This figure shows that we start with the most expensive shift and decrease the amount of order pickers (N in Figure 17) by one. For each decrease, we perform an evaluation of the solution as described in Section 4.3.3.1: determining for all time slots whether the restrictions are met. When one of the restrictions is violated, we use the nearest solution where the restrictions were not violated. After optimizing the most expensive shift, the amount of order pickers in the second expensive shift is calculated the same way. The process stops when all shifts are calculated.

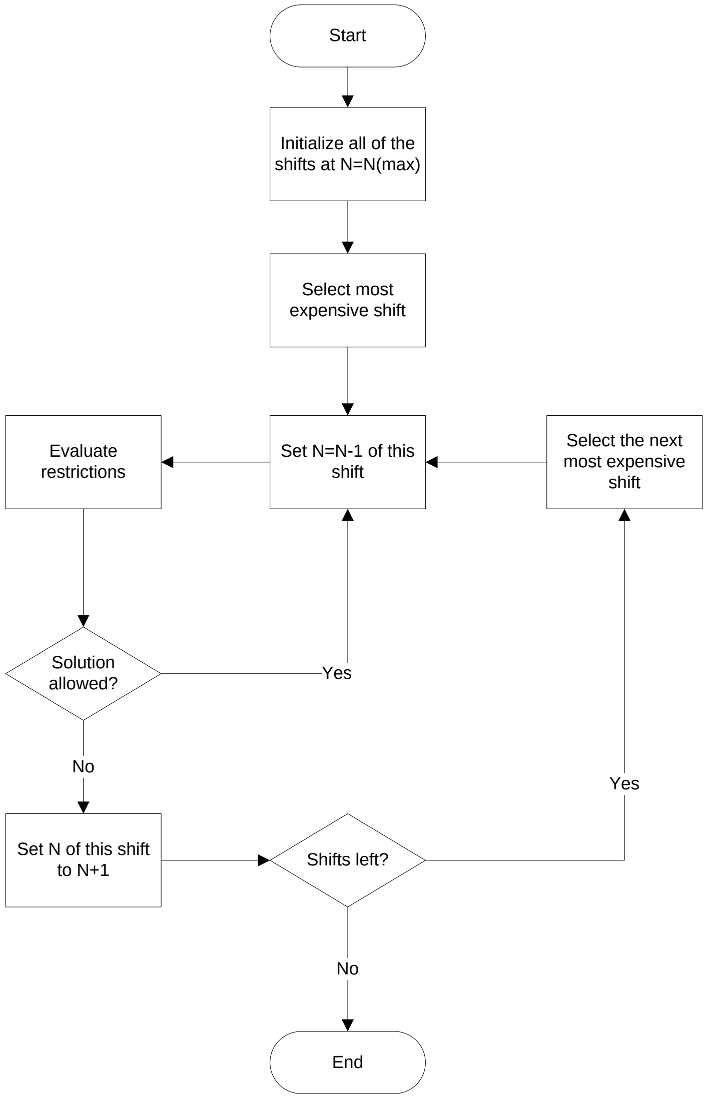

Figure 17: Improvement order picking model, N being the number of order pickers

We claim our solution method yields the lowest costs giving the following assumptions. (1) The shifts are in decreasing order of costs, (2) The total amount of goods to be order picked is fixed, (3) The productivity per employee is fixed. In addition, using this solution method will imply the least over usage of dock capacity, given the orders. Section 4.3.4 provides proof of this claim.

4.3.4 Motivation for the improved order picking model

This section explains the motivation for the improved order picking model. We propose to use this model to replace the current way of modelling the order picking. There are two hard constraints we want to meet. In addition we minimize the costs of the shifts and the excess usage of dock capacity. We claim our solution method generates an optimal solution to this problem for a given DTWP. The solution method proposed is that we start with the maximum order pickers and reduce the amount of order pickers in the most expensive shift by 1 until this violates a restriction. After finding the last feasible solution, we set the amount of order pickers of the last feasible solution as the amount of order pickers for the shift under consideration. We then continue to do the same for the next shift.

This yields an optimal value for the costs of the shift when the first shift is more expensive then the second shift, when using two shifts. The total amount of orders that have to be picked is fixed, as explained in Section 4.3.3.2. This means the total order picking hours are fixed, because we assume a fixed productivity. In general, for N shifts this holds when the costs c_{N+1} of shift $N+1$ are less than the costs c_N of shift N, because in this case it is always better to order pick something in a later shift, if possible. Order picking something in a later point of time means it stays the least amount of time at the docks.

The results of our solution method compared to the method used by the SCCP department (explained in Section 2.1) are similar. The biggest difference is due to the different transportation model used. The SCCP department uses Shortrec, while we used the WDScan transportation model. If we use the order picking amounts WDScan calculates and run both the models, they have comparable outcomes. The percentages of order pickers required in the shift is in the original model 64% and 90%, while in our model 63.71% and 90% are required. This difference is caused by the accuracy of the models; our model has a step size of 0.005%, while the SCCP model has a step size of 1%. We suggest to use a step size of one employee, which calculates the exact amount of employees required. The same hours were indicated where the buffer capacity was surpassed. However, the power of this model is not in the accuracy, but the integration with WDScan.

The manual process is based on changing delivery time windows, because the buffer capacity is over used, as explained in Section 2.1. The SCCP department indicates the hours where too much buffer capacity is used, and requests changes to the delivery time windows, e.g. changing some deliveries of stores to less busy slots. Our model can be used in WDScan, to evaluate a certain DTWP. The costs of order picking this DTWP is returned to WDScan and the total DTWP is evaluated. WDScan changes a delivery time window and re-evaluates the order picking costs. We argue that a better DTWP can be found, because instead of making a DTWP that fits, a DTWP with minimal costs is created.

4.4 Conclusion

This chapter answers the research question '*How can we improve the WDScan model and what is the performance of the improved model?'*. We propose three improvements to the WDScan model. The first is the delivery size estimation model, which determines the amount of goods sent to stores when changing the heartbeat moment. The second improvement is the penalty costs. The third improvement is the order picking model.

The new delivery size estimation improves the accuracy of how many goods a change in the heartbeat of a store would inflict. WDScan now assumes that no changes occur in the delivery size when changing the heartbeat moment. We have shown that in practice the delivery size fluctuates based on the amount of customers supplied by this delivery. The improved model takes the amount of customers supplied by this delivery into account and calculates how many goods are delivered for a certain hour of sales. Based on these goods per hour, when changing the heartbeat moment from morning to afternoon we see different amounts of goods delivered between the two heartbeat moments. We validate the estimates by using stores that have the same customer pattern but a different heartbeat moment. In addition, we illustrate practical case where a heartbeat moment of a store was changed. The improvement of the delivery size estimation is only viable for the fresh goods, because the ambient goods are less customer driven.

The improved penalty costs improve the ability of WDScan to generate better solutions. WDScan currently optimizes by reducing penalty costs and increasing real costs. However, because the current DTWP is a feasible solution but yields a lot of penalty costs we propose a new method. The new method is an upper bound for the hard restrictions that have to be evaluated after implementing the new order picking model. These new upper bounds already reduce the penalty costs greatly. However these new upper bounds make it more difficult to find optimal solutions, because these optimal solutions are usually near the infeasible area. The soft restrictions' penalties have to be assessed after the improved order picking model is implemented.

The improved order picking model provides a more accurate model of the order picking process in practice. WDScan assumes an order is picked just-in-time, however in practice orders are order picked in advance. This means WDScan assigns high penalty costs to solutions that are actually feasible. Our improved model is able to cope with order picking in advance. The solution method of our model finds an optimal solution, given the model assumptions. In addition, our model is able to cope with more than two shifts. The order picking in the DCs in currently done in two shifts. However, if this changes, our model is able to cope with more than two shifts.

The three proposed improvements provide big improvements to the WDScan model. The order picking and penalty costs model provide a great reduction in the penalty costs of the current DTWP. Reducing this penalty costs is necessary, because the current DTWP is a feasible schedule and should therefore not inflict this many penalty costs. Using the improved order picking model and penalty costs means WDScan is not just trying to inflict less penalty costs, and can actually improve the current DTWP. The delivery size estimation makes sure the expected amount of goods for a delivery is more accurate. More accurate delivery size estimation is required, because it affects the amount of orders to be picked in the distribution centre and the amount of load carriers in a truck used to calculate distribution centre and transportation costs. In addition, the improved delivery size estimation provides more insight to the store managers what the effect of a change in their heartbeat moment is.

5. Conclusions and future research

The goal of this chapter is to answer the research questions, assess whether the research goal is achieved, recommend additional steps to improve WDScan and propose interesting future research areas. Section 1.2 defined the main goal of the research as: '*Provide a plan to use WDScan to make better decisions on store, DC and transportation costs versus the amount of customers bothered for the busy weeks'.* Section 5.1 answers this research question and assesses whether the research goal is achieved. Section indicates interesting future research areas.

5.1 Conclusions

Based on the problem definition described in Chapter 1, our research goal is formulated. To achieve this research goal, we formulate four research questions. This section answers the research questions and determines whether we achieve the goal of the research. The problem indicated by Albert Heijn is that they would like to use WDScan to determine the Delivery Time Window Plan (DTWP) for the busy weeks, where WDScan is not yet able to determine a proper DTWP.

To answer the first research question, *'How are the time windows determined in the current situation?',* we investigate the current situation. The current process is a manual process that takes a lot of time. The manual process starts with determining the increased delivery sizes. Based on these increased delivery sizes, addition trucks are scheduled to deliver stores. The next step is determining whether the distribution centres are able to order pick these additional and increased orders. A feasible schedule is created by making changes to the proposed DTWP.

WDScan should be able to automate the DTWP process. We investigate WDScan by answering the second research question, *'What does the WDScan method look like, and what are its benefits and drawbacks?'*. WDScan uses an incremental approach to determine the DTWP. It assesses the distribution centre, store, transportation and customer costs of different DTWPs. WDScan modifies the DTWP randomly by using simulated annealing. The total costs are compared to the previous DTWP each iteration. The major benefit of WDScan is that it gives an incremental overview of the costs of a certain DTWP. However, because of this incremental approach, WDScan uses a lot of input data and several costs models. To investigate why WDScan is not yet able to produce a proper DTWP we investigated the input and the parts of WDScan in depth. We found that the order picking model, which determines the distribution centre costs and the validity of the DTWP for the distribution centres, is flawed. In addition to the real costs, WDScan uses penalty costs to model indirect costs and allow some infeasible schedules. We found these penalty costs do not represent these indirect costs properly and result in many infeasible schedules. The final drawback is based on the input used by WDScan. The input facilitates hour specific delivery sizes to stores, but using hour specific delivery sizes is not used by Albert Heijn yet. Using hour specific delivery sizes increases the accuracy of the model and the acceptance of the store managers. We therefore propose to improve this.

In order to place our problem in literature and find solutions, we investigate the third research question, *'What are similar problems to the WDScan problem in literature, and what are possible solutions for this* *problem in literature?*'. We found that our problem is similar to an inventory routing problem as described by Gaur and Fisher (2004). The IRP model that Gaur and Fisher (2004) describe was built for Albert Heijn, and concerns a similar problem. The differences are that this problem does not investigate the actual costs of a DTWP, but proposes delivery time windows mainly based on transportation and distribution centre costs. Our problem incorporates, in addition to the transportation and distribution centre costs, the store costs and customer costs. In addition, we explain simulated annealing and how to deal with penalty functions.

In order to improve WDScan, we answer the fourth research question, '*How can we improve the WDScan model and what is the performance of the improved model?*, These improvements are based on the indicated areas for improved introduced in Chapter 2. To improve the hour specific delivery sizes of the input, we assess the usage of amount of customers per delivery to determine the delivery size. We show that using the amount of customers per delivery to determine the delivery size is a good estimate by calculating the hour specific delivery sizes and comparing these for a number of stores. We use stores with similar customer patterns, therefore we expect similar delivery sizes (as a fraction of their total delivery size) when setting the same delivery time window. It turns out these delivery sizes have a statistically significant, high correlation. The penalty functions are improved, for the hard restrictions. We determined the upper bounds such that infeasible solutions are penalized severely. In addition, we introduce an improved order picking model. The basis of this improved order picking model is the method used in practice, as explained in Section 1.2. Using this improved order picking model gives a more realistic view compared to the just-in-time model currently used. Our improved model allows order picking in advance, similar to the situation at the distribution centres of Albert Heijn. The improved order picking model uses a solution method that determines the least amount of order pickers required to facilitate the DTWP under consideration. We claim this solution method finds the optimal solution, given the assumptions. In addition, we describe steps Albert Heijn has to take in addition to the hour specific delivery sizes, the order picking model and the penalty costs.

Finally, our research goal '*Provide a plan to use WDScan to make better decisions on store, DC and transportation costs versus the amount of customers bothered for the busy weeks'* follows from the research questions. To make WDScan make better decisions we advise Albert Heijn to use the improved order picking model, the improved penalty costs functions and the amount of customers to determine the hour specific delivery sizes. In addition, we recommend Albert Heijn to take additional steps to further improve WDScan. These additional steps are explained in Section 5.2

5.2 Recommendations

In order to use the improved WDScan model to determine the DTWP for the busy weeks additional steps have to be taken. Section 2.3 explained that some parts of WDScan are not selected for improvement, because they are out of the scope of the research. Our analysis of WDScan shows that in order to do these other improvements, more pressing matters should be addressed first. Figure 18 shows which improvements are within the scope of this research; those improvements are addressed in

Chapter 4. These improvements are the delivery size estimation per hour, the upper bounds for the penalty costs and the improved order picking model.

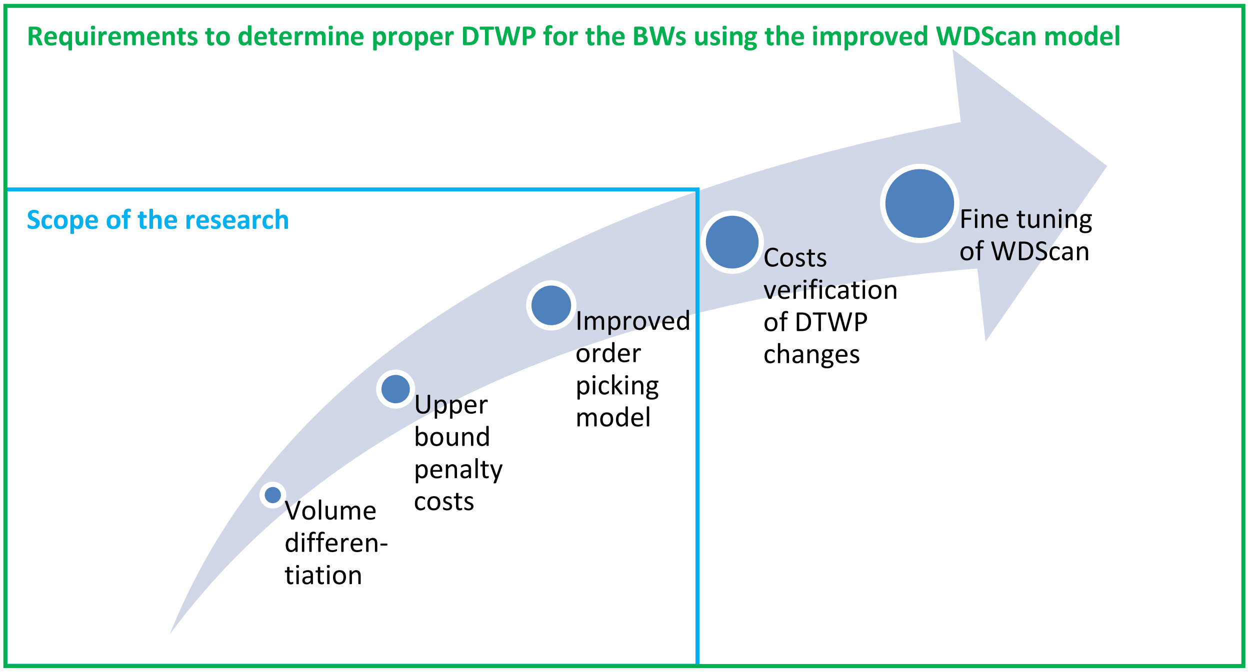

Figure 18: Model requirements WDScan for determining the DTWP for BWs

We recommend Albert Heijn to do additional improvements to WDScan. The improvements we propose are: (1) the costs verification of DTWP changes and (2) the fine tuning of WDScan. We describe the two recommendations in Sections 5.2.1 and 5.2.2.

5.2.1 Costs verification of DTWP changes

The costs verification of DTWP changes considers the validation of a new DTWP. The costs of transportation, distribution centre and store per hour are considered here. Especially the store costs are considered to be flawed, because there is just a small increase in restocking costs between restocking during cheap hours and expensive hours. The cheap hours are for example in the afternoon where a lot of students are available for restocking, while the expensive hours are during night. During the night, restocking has to be done by expensive shifts, and enquires more costs per load carrier. We propose a method to evaluate small changes to the DTWP. We advise Albert Heijn to first evaluate the current DTWP, and use the small changes done to an actual DTWP to assess the difference in costs. These costs should be comparable to the change WDScan calculates between these two DTWPs. In addition, reviewing whether changes for individual stores give representative changes. This means viewing the difference in store costs in the output of WDScan and comparing this to what changes in the actual store. If these changes are validated, larger changes should be evaluated. These can be assessed using the bigger DTWP changes that happen once a year at AH.

5.2.2 Fine tuning of WDScan

The second recommendation to Albert Heijn is fine-tune the model. The parts that have to be fine-tuned are the weights of the target function parts and the penalty costs. To illustrate the target function parts, we present the target function of WDScan:

$$Min\ z = \sum_i \{A(Customers\ bothered_i) + B(Transportation\ costs_i) + C(\ Store\ costs_i) + D(Distribution\ centre\ costs_i) + \ penaltycosts_i\} + global\ penalty\ costs$$

In the target function of WDScan A, B, C and D are the weights of the target function parts. These weights are used to give the parts relative importance. These should represent the preferences of Albert Heijn. An important question to assess the preference of Albert Heijn is: *how much is not bothering a customer worth to Albert Heijn?* In combination with the soft penalty costs, the weights in the target function move the solution in a certain direction. The soft penalty costs should not be a huge part of the target function if the soft penalties are not considered important. Therefore representative soft penalty costs and target function weights should help the model move to a preferred solution. In addition, these penalty costs and target function weights have to be matched with the hard penalty costs and real costs. In order to make sure the hard penalty costs are not violated in favour of the soft penalty costs, as in the current WDScan model.

Based on the upper bounds of the hard penalty costs, the hard penalty costs should be tuned. First, the upper bounds should be validated. This means it has to be assessed whether the upper bounds still hold. The upper bounds should be high enough to make sure no infeasible solutions are allowed. However, using these upper bounds would limit WDScan to feasible solutions. For the final solution using only feasible solutions is desirable. When optimizing using simulated annealing it is beneficial to be able to search through infeasible solutions. These upper bounds should be improved by determining the value high enough, that they do not allow infeasible solutions, but as low as possible. The problem here is that the penalty costs have to be re-evaluated if the total costs of a solution change significantly.

5.3 Future research

In addition to the required recommendations described in Section 5.2 we illustrate some interesting areas for future research. The first area interesting for future research is how to cope with Sundays. In the WDScan model, Sunday is considered a normal day. However, the distribution centres have different opening hours on Sunday and some stores cannot be delivered. Currently, WDScan ignores the different Sunday opening hours. Because we use a heartbeat moment, where we deliver the stores at the same time every day, we cannot change this for a single day. A manual process is currently required to make the Sunday DTWP fit within the restrictions on Sunday. An interesting area for future research is how to make WDScan cope with the Sunday restrictions.

The customer bothering model used in WDScan is another interesting future research area. Section 2.2.3 explains that the amount of transactions is not the best way to assess how many customers are in a store at a certain moment. We suggest to investigate whether the spending per transaction could be used together with the number of transactions. Van Lunteren (2007) has shown this gives a better calculation of the timeslots store managers perceive to be busy. However, this research is done a small scale and it should be assessed how to incorporate this in WDScan.

Probably the most interesting future research is how a model such as WDScan is able to determine the best time windows for a general Inventory Routing Problem. WDScan has been customly designed for Albert Heijn. However, it is interesting to investigate whether this approach can be used to evaluate soft time windows for Inventory Routing Problems. WDScan is not primarily focused on transportation costs, but rather tries to give a total overview of the costs.

References

Aarts, E., & Korst, J. (1989). *Simulated Annealing and Boltzmann Machines: a stochastic approach to combinatorial optimization and neural computing.* New York, NY: John Wiley & Sons, Inc.

Albert Heijn. (2013). *Uitleg WDScan (in Dutch).* Zaandam: Albert Heijn BV.

Albert Heijn. (2014). *Basiskennis Replenishment (in Dutch).* Zaandam: Albert Heijn.

Albert Heijn. (2014, April 7). Simulatie tool uitval (in Dutch). Zaandam, The Netherlands.

Beerepoot, M., & Spoelstra, A. (2014, April 2). Flaws in inputdata. (R. Smit, Interviewer)

Beerepoot, M., & Spoelstra, A. (2014, February 19). Interview 1. (R. Smit, Interviewer)

Bertazzi, L., Savelsbergh, M., & Speranza, M. G. (2008). Inventory Routing. In B. Golden, S. Raghavan, & E. Wasil, *The vehicle routing problem: latest advances and new challenges* (pp. 49-72). New York, NY: Springer Science+Business Media, LLC.

Cordeau, J.-F., Laporte, G., Savelsbergh, M., & Vigo, D. (2005). *Vehicle Routing.* Montréal: Canadian Natural Sciences and Engineering Research Council.

Danzig, G., & Ramser, J. (1959). The truck dispatching problem. *Management science, 6(1)*, 80-91.

Garfinkel, R. (1985). Motivation and Modeling. In E. Lawler, J. Lenstra, A. Rinnooy, & D. Shwoys, *The traveling salesman problem* (pp. 307-360). Chichester: John Wiley & Sons.

Gaur, V., & Fisher, M. (2004). A Period Inventory Routing Problem at a Supermarket Chain. *Operations Research vol. 52, No. 6*, 813-822.

Henderson, D., Jacobson, S. H., & Johnson, A. W. (2003). The theory and practice of simulated annealing. In M. Gendreau, & J.-Y. Potvin, *Handbook of Metaheuristics* (pp. 287-318). New York, NY: Springer.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science, New Series, Vol. 229, No. 4598*, 671-680.

Laporte, G., Nobert, Y., & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research 33*, 1050-1073.

ORTEC. (2013). *Modellering ScenarioanalysetoolWinkedistributie Albert Heijn(in Dutch).* Gouda: ORTEC Consulting Group.

ORTEC. (2014, April 4). Mail conversation Noud Gademann. Zaandam, The Netherlands.

Papadimitriou, C., & Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity.* Englewood Cliffs, N.J.: Prentice-Hall.

Richardson, J. T., Palmer, M. R., Liepins, G., & Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. *Proceeding of the Third International Conference on Genetic Algorithms* (pp. 191-197). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Schutten, J. (2012, September 9). Lecture slides DOBP. Enschede, The Netherlands.

Shapiro, S., & Wilk, M. (1965). An Analysis of Variance Test for Normality (Complete Samples). *Biometrika, Vol. 52, No. 3/4 (Dec)*, 591-611.

Smith, A. E., & Coit, D. W. (1997). Constraint-Handling Techniques - Penalty Functions. In T. Baeck, D. Fogel, & Z. Michalewicz, *Handbook of Evolutionary computation.* Bristol, U.K.: Institute of Physics Publishing and Oxford University Press.

Spoelstra, A. (2014, March 7). WDScan difficulties. (R. Smit, Interviewer)

Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem.* Philadelphia, PA: SIAM Monographs on Discrete Mathematics and Applications.

Van Lunteren, R. (2007). *Analyse klantenpiekmomenten (in Dutch).* Zaandam: Albert Heijn.

van Lunteren, R. (2014, March 25). Clusteranalyse winkels (in Dutch). Zaandam, The Netherlands.

Vanderbei, R. (2008). Integer programming. In R. Vanderbei, *Linear programming: foundations and extensions* (pp. 385-405). New York, NY: Springer.

Woeginger, G. (2003). *Exact Algorithms for NP-Hard Problems: A survey.* Enschede: University of Twente.

Appendices

Appendix A: List of input parameters

Due to confidentiality reasons, this table is not displayed in the public version of this thesis.

Table 8: List of input parameters

Appendix B: Cluster Analysis

In order to find stores that have the same customer pattern, a cluster analysis is provided by Albert Heijn (van Lunteren, 2014). These clusters are based on an analysis of the stores of Albert Heijn that have a delivery of fresh goods every day of the week (Monday through Sunday), and are the same kind of store. This resulted in 126 stores that could be clustered. We have used the biggest cluster to assess the performance of the algorithm. This cluster contains 36 stores that have a similar customer pattern. From this pattern we have removed the extreme values; these have characteristics that we cannot explain based on the customer pattern, or are the extreme values of this customer cluster. The cluster that remains is displayed in Figure 19. In the figure, the amount of customers is displayed as a fraction of the total amount of customers over the week for this store, over the hours that count towards this total. Some hours are not used, because not all stores are opened at these times, and would not give a good clustering.

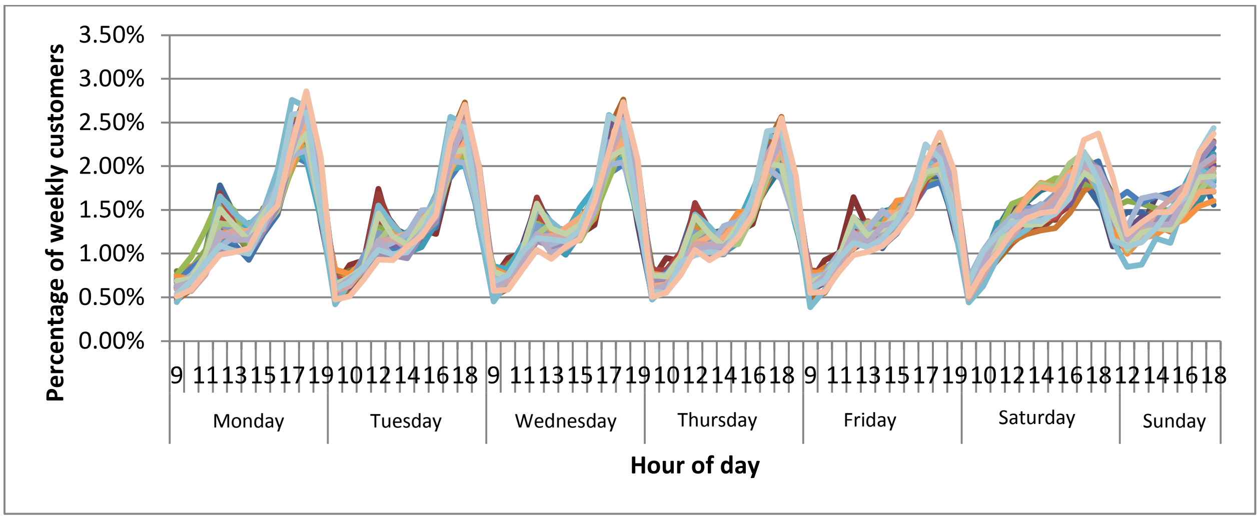

Figure 19: Customer pattern over the week for selected stores

Figure 19 shows the customer pattern based on multiple weeks of data for different stores. In this figure the lines represent the pattern of the different stores, the X-axis shows the days of the week and the hours of these days. It can be seen that these stores have a larger peak in the first few days, and during weekdays a peak during lunch and a large peak around 5 pm. The amount of customers on Saturday and Sunday is significantly lower than during weekdays. Because of the large peaks during lunch and around 5 pm we would expect these customer patterns in urbanized areas. Most of the stores in the cluster illustrated in Figure 19 are located in Utrecht or Amsterdam. The most deviation of customer patterns is seen on Monday, Saturday and Sunday.

Appendix C: Selection of 10 stores

Based on the clusters provided by Albert Heijn (Appendix B), we have run the improved delivery size estimation model to test whether the fraction of the fresh goods to deliver is the same over the week independent of the original heartbeat moment. We have calculated the fraction of the week volume for a delivery, if the delivery time window average of a certain time is used. For the cluster shown in Appendix B, Figure 20 shows the percentage of week volume for these days calculated by the model. This Figure shows that the algorithm expects that if a heartbeat moment is chosen at 1 am, for store 1078, 16 percent of the weekly volume will be delivered on Monday. This figure illustrates that the different stores, with different delivery time plan windows, have a similar percentage of goods that are expected to be delivered. On the x-axis the days and hours of this day are displayed.

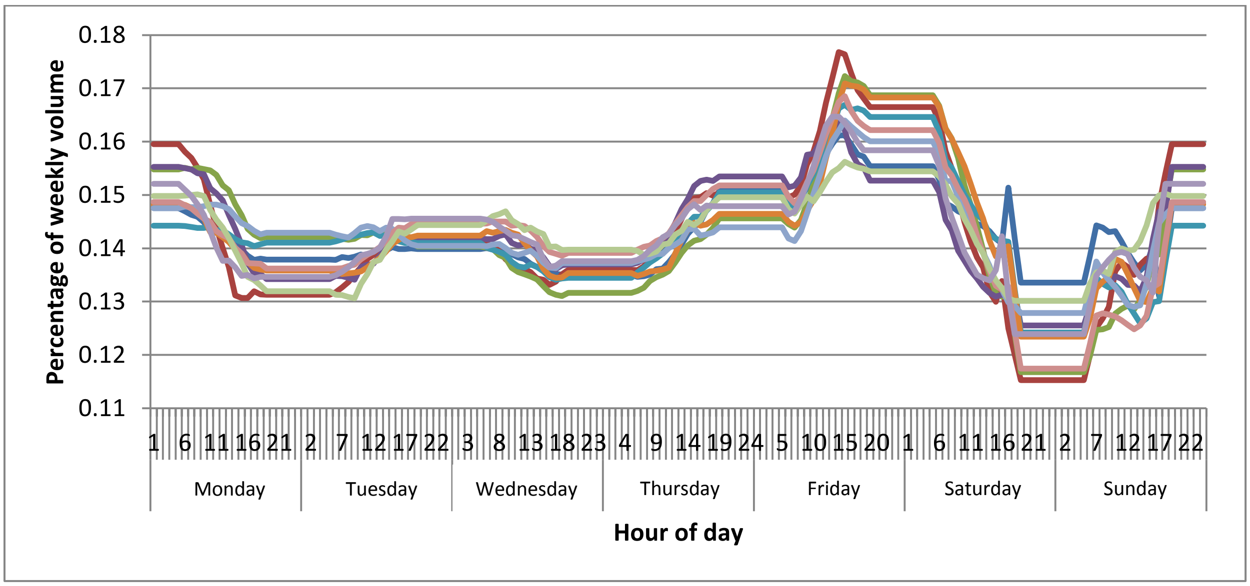

Figure 20: Delivery size fractions over the days for selected stores

Figure 20 shows fewer stores than Figure 19, because we removed extreme values and patterns that did not match. The goal is to find stores that are very similar, not just by customer pattern. The rules whether to send a package of goods or not are not just based on customer patterns. These rules also take the shelf capacity and availability into account.