Empty Wagon Repositioning; Heuristics for a Hub-and-Spoke Network

Master thesis

Author:
Ing. Tommy Blom
343918

Abstract This thesis considers an empty wagon repositioning problem with a perfect hub-and-spoke network. Within this problem, empty wagons are repositioned while the overall costs are minimized. Based on supply and demand, the wagons are repositioned under the assumption that they depart somewhere during the day and arrive before the end of the day. Small problems are solved with an integer programming (IP) problem which is based on a multi-commodity flow formulation. However, the introduced IP formulation is new, as it also takes reclassification costs explicitly into account. Wagons that travel a part of their route together, can be seen as one block during that part. This lowers the reclassification costs on that part of the route. For larger problems three heuristics are proposed, which are all based on the principle of splitting the complete problem into smaller sub problems which are then solved with the IP. Computational tests indicate that two heuristics can be used for practical purposes.

Keywords: hub-and-spoke; empty repositioning; reclassification costs; heuristics; network flow

Erasmus University Rotterdam
Econometric Institute

Supervisor:
Dr. Twan Dollevoet
Coreader:
Dr. Dennis Huisman

Ab Ovo International
Business Unit Advanced Planning and Scheduling

Supervisors:
Drs. Peter Koot
Drs. Dieter Veldhuis

October 9, 2014

Contents

1 Introduction

Next to maximizing profit, more and more companies try to reduce their CO_2 emissions. For transport companies, both goals can be improved by reducing the number of empty trips. When cargo is delivered, cars, trucks, and trains leave the receiving party most of the time empty. This would not be a problem if the vehicle goes to the neighbour to pick up the next cargo. Unfortunately this is rarely the case. If a vehicle is not repositioned, it needs to travel the complete way back to the depot, empty. This results in extra costs and emissions, without any revenues. Although empty repositioning has been studied even before the first world war by Cavanagh (1907), it became a more practical and popular research topic after the introduction of the container (Florez, 1986). Even though most research has been done on empty repositioning within the container business, still 20.5% of the containers handled by ports are empty (Rodrigue, 2013). This might seem a large percentage, but this is still better than the 53% which is the average percentage of "empty movements" of a large private rail cargo company (Maltsev, 2014). In this thesis we will develop a new approach to empty repositioning that takes reclassification costs into account. Our approach models the reclassifications using a new integer programming problem and multiple heuristics, which all reduce the number of empty movements for a train cargo company in order to increase profit while decreasing the CO_2 emissions.

In a rail network empty moves are partly caused by the limited parking space at the facility of end customers. For example, a lumber camp in a forest can most of the time only be reached using a single track. This means that once a train is full, it needs to leave. Because no empty train could park at the forest, production might stall until the next train arrives. At the other end of the process there is limited parking space as well. At the paper factory, empty wood wagons need to leave immediately, since it is required to offer room for new arriving wagons. If the lumber camp is already provided with empty wagons, the wagons at the paper factory are sent to the depot. From there, after a certain time, they will be sent back to the forest again. This results in an empty wagons storage at a depot. In this example, if production time is optimized, three trains are needed.

This is 1.5 per company, from which 0.5 car is in the depot without generating any revenue. *"If the time that a car is unutilized can be reduced, then the total number of freight cars can be decreased without reducing the amount of goods which can be transported, and it follows that the capital costs will decrease"* (Holmberg et al., 1998). When more wood processing companies are involved, this can be achieved by effectively reassigning and repositioning the empty wagons.

Compared to loaded transports, the planning of empty transports gives more freedom to the railway company on how to implement the empty movements. A transportation order of loaded cars has a fixed origin, a fixed destination, and is often scheduled for a specific day, which means that there are very few alternatives for implementing the order and the movement. When repositioning empty cars, there are no origin-destination specific demands that have to be fulfilled, and no specific time schedule for each empty transport (Joborn et al., 2004). Since most demands of customers are known beforehand and loaded wagons will lead to empty wagons, one can schedule empty wagons directly with the loaded wagons. The wagon scheduling process will then become more complex, but should lead to more profit. For the empty repositioning problem, specific times are not required. Knowing which wagons need to go to which place is enough to solve the repositioning problem.

The process starts with customers. Since these customers ask for train wagons, they will be located next to a train station. For companies, this station might just be a private track on their facility. These small stations are called service locations. Next to service locations, there are base locations. These locations are larger and are connected to multiple service locations. Base locations are bigger, centralized, and contain many railway tracks next to each other. At base locations, which are owned by the railway company, reclassification actions can be performed and wagons can be stored.

The meaning of reclassification is explained in the article of Bektas et al. (2008) in section 2.2. *"There exist two types of rail terminals in rail networks, flat and hump yards, depending on how the classification process is carried out. A detailed description of yard operations can be found in Petersen (1977). In Fig. 1, we depict* *the general structure of a hump yard. A flat yard exhibits a similar structure, except there is no hump. A train arriving at a yard first enters a receiving area, where the engines are taken away for inspection and maintenance, blocks are separated and cars are inspected. The classification operation begins from this point on, and it can be performed in two ways, depending on the type of the rail yard. In flat yards, a switching engine is used to push a group of cars out of the receiving tracks and shove them onto one of the classification tracks. In hump yards, classification is performed by using an artificially built hill, called the hump, where an engine pushes a group of cars out of the receiving tracks and up the ramp until it reaches the top of the hill. Then, with the help of the gravitational force, the cars roll down the incline, usually one car at a time, and are directed onto one of the classification tracks. Following this operation, each classification track thus becomes occupied by a group of cars that form the block. Each block then waits until the departure time of its outbound train. When the time comes, the blocks are pulled out of the classification tracks onto the departure tracks and are attached to the train. Following one last inspection of the whole train, the train and blocks leave the yard. Classification is not the only operation that is performed at a rail yard. Other types of operations include inspection, crew change, refuelling the trains, and dropping and picking up blocks of cars. Among these, however, classification is known to be the major time-consuming operation."* (Bektas et al., 2008)

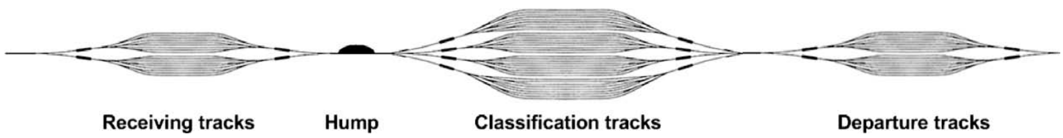

Figure 1: Classification yard example (Bektas et al., 2008)

In this paper we come up with a solution for empty wagon repositioning with the aim of reducing the total costs and emissions of railway companies. The solution is tested on a large real-size problem instance. In order to solve the empty repositioning problem, the exact problem is described in Section 2. In Section 3 the literature dealing with similar problems is reviewed. In Subsection 3.4 the conclusion of the literature research is given. In Section 4 the mathematical formulation is explained. The integer programming problem can be found in Subsection 4.2. Next to the IP, three heuristics are proposed in Subsection 4.3. The performances of the heuristics and the IP are analysed in Section 5. The main conclusions of this thesis can be found in Section 6. Finally, the main items of this paper are reflected and discussed in Section 7 where also some recommendations are given for practical implementation as well as for further research.

2 Problem description

We will solve the empty repositioning problem for a train cargo company in Europe. Their scheduling is split in an operational, a tactical, and a strategic phase. In the strategic phase, the so called 'model week' is planned. In the model week all trains are scheduled. Since production in- and output of most customers remains (almost) constant for a longer period, a model week can be repeated several times. Repeating such a week reduces the number of scheduling operations which is a complex, time consuming task. The model week can be planned months ahead. Based on experience of Ab Ovo, around 30% of the schedule will be adjusted, cancelled or added on a shorter notice to cope with changes in the demand. Such adjustments are made in the tactical phase. In the operational phase, day to day problems are solved. In this thesis we will focus on the optimization of the empty wagon repositioning for the model week.

The network is considered to be a hub-and-spoke network. Clustered customers are served from service locations. These are the smallest nodes in the network. They are linked to a parent node, the so called base locations. These base locations are then linked to their parent, an Empty Wagon Parking Place (EWPP). Each node has a limited parking capacity. The EWPPs are locations with a great quantity of parking capacity, used to store empty wagons. A schematic overview can be found in Figure 2. In practice there are also physical rail connections between base locations. For the sake of simplicity we assume a perfect hub-and-spoke network, like in the schematic overview. Although there might be some rail connections between two service locations in practice, we assume for each node, that they can only be reached via their parent node.

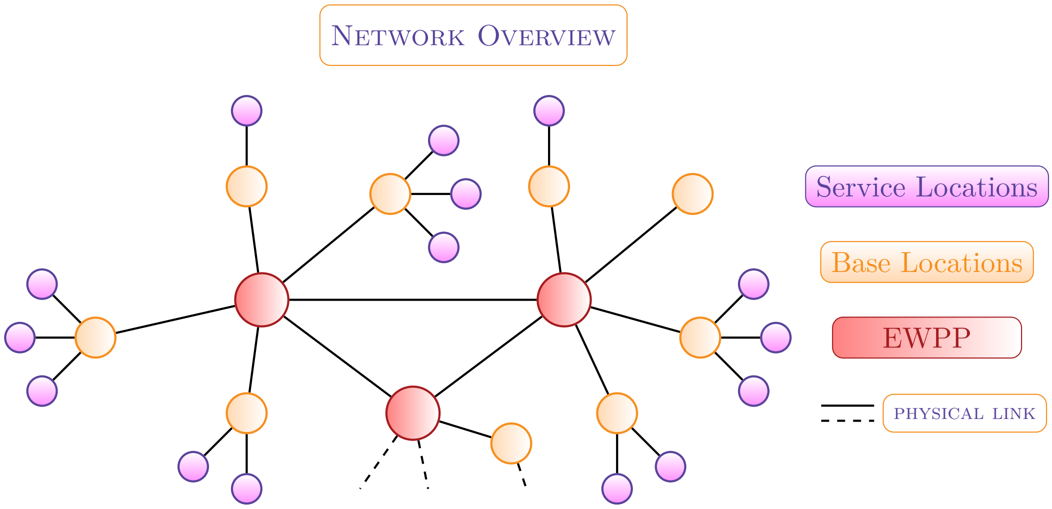

Figure 2: Hub-and-spoke rail network overview

In the model week, there are seven days with demand for empty wagons. All these demands need to be fulfilled. Every cargo delivery will result within a couple of days in new empty wagons. These wagons can in our model be viewed as supply which needs to be reassigned. For each demand, a demand-specific wagon type is required. Each day, there will be a surplus or deficit on some service locations, which differ per wagon type. We will restore balance by reassigning wagons on a daily basis. We do not schedule specific trains, but only send wagons at the beginning of the day to a new destination, where they will be for sure at the end of the day. Reassigning locations to each wagon will result in costs that need to be minimized. Besides the costs, demands, and parking capacity, we take limited capacity within a train into account. Since the railway network is mainly single-track, the total length of a train is limited by the length of the shortest side parking place, where a train can wait and be passed. Although we do not schedule specific trains, we do need to calculate the amount of trains we need, since an extra train will result in extra costs. This can be done by dividing all wagons on one track by the maximum capacity of one train.

2.1 Repositioning example

In this subsection, an example will show the complexity that planners face when they try to solve the empty repositioning. In the example, a small network is used with a limited number of wagon types. Next to the detailed description, the example is also visualized in Figure 3.

Consider a base location of the network with three connected service locations, an EWPP, and three different types of wagons. These wagon types are specified by a colour. At day one, service locations S1 and S2 hold three yellow wagons each. Service location S3 has two blue wagons and two green ones. The base location is empty and the EWPP contains ten wagons of each type. On the second day, there is a demand of two yellow wagons at S3. This demand can be met from three different locations, S1, S2, and the EWPP. On top of demand that needs to be met, each location has limited parking space. In this example, S1 and S2 can store three wagons and S3 can store four wagons. B1 has place for five and the EWPP for 40. The limitation at S3 forces the decision maker to reassign at least two wagons from S3. Whether these wagons are sent to B1, the EWPP or another location, is up to the planner, who has the goal to meet all demands while minimizing the overall costs. Although neither the blue nor the green wagons are needed on day two, this is the case for day three. S2 requires two wagons of type green and B1 two blue ones.

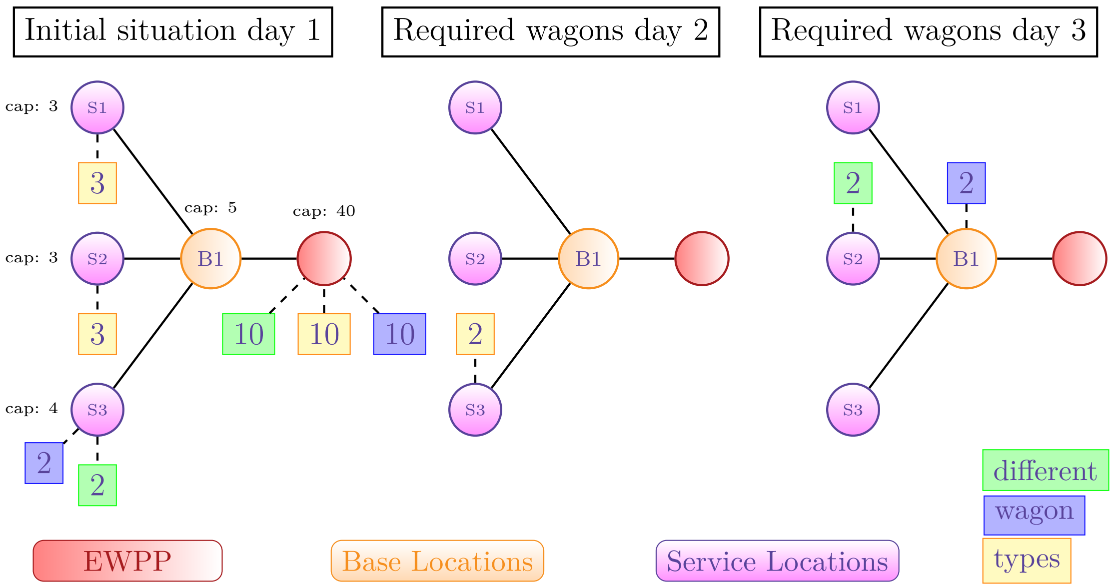

Figure 3: Repositioning example

To make room for the required two yellow cars at S3 for day two, the blue ones can be moved to B1. Since the blue ones should be there at day three, this makes sense. Another logical option is to move the green wagons. They need to be at S2 on day three, so they can be moved to S2. Unfortunately, there is no parking space left at S2. This can be solved by reassigning the yellow wagons, or simply by reassigning the green wagons to B1. B1 is on the route to S2 and B1 has still parking space left, so the wagons can stay there during day two. Independent of the planning decision of the green wagons, reassigning two yellow wagons to S3 is necessary anyway. Since parking space at S2 should be cleared for day three, this is an efficient way of reassigning the wagons. If the planner does not use a two day planning horizon, it might be the case that yellow wagons are assigned from the EWPP or from S1. Such possible solutions for day two are visualized in Figure 4.

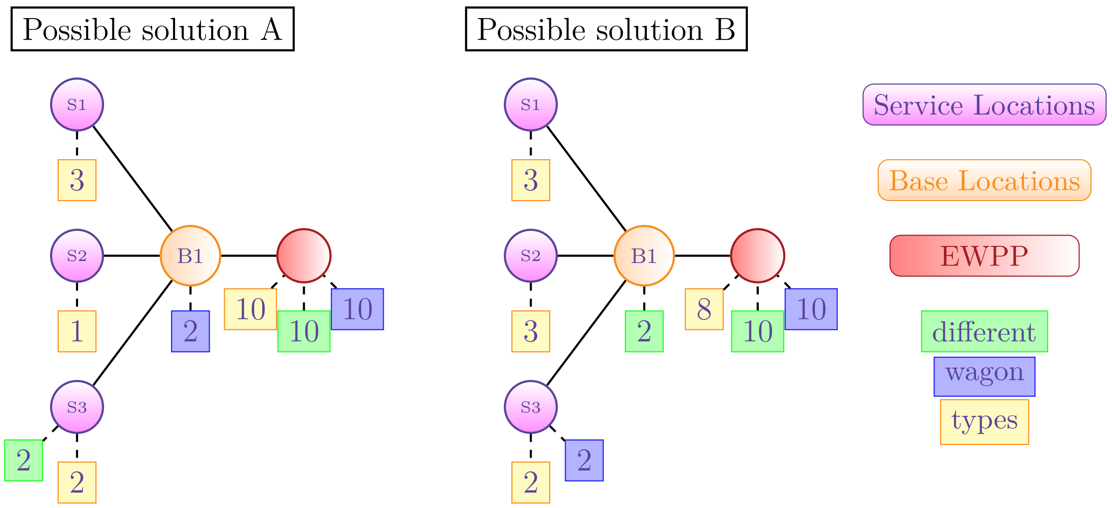

Figure 4: Repositioning example - possible solutions to satisfy the demand at S3

In this example, it is optimal to reassign the yellow wagons from S2. Since then already empty wagons leave S2, we might want to move the third one to B1 as well. The cost will be negligible compared to a new train, which is needed if the one wagon left needs to be picked up a few days later. The green and/or blue wagons can be moved to B1. The green wagons can also be sent directly to S2. With only the green and blue wagons from S3, already multiple different options are possible. Combine this with the option of leaving one yellow wagon at S2 and the number of different scenarios doubles. And these scenarios exclude options where yellow cars of S1 and the EWPP are reassigned. From all these options, a planner should pick the best option which is defined as the one with the lowest costs, while still meeting all demands.

Even with this small example, repositioning is already a challenging task. This thesis aims to find a method to solve this problem for the planners of railway companies. Such a method should minimize the overall costs. The different costs will be discussed in the next section.

2.2 Costs

In this problem setting there are three different cost types: train setup costs, travelling costs, and reclassification costs; these will be explained in the following subsections. We will need a function to calculate the overall costs in such a way, that we are able to compare different strategies. Therefore it is not required to calculate the exact total costs. Costs that will occur anyway and cannot be reduced by any scheduling decision, can be left out in order to reduce the number of calculations.

2.2.1 Train setup costs

Each train requires a new locomotive with a crew. The crew scheduling costs, insurance costs, and salary of the crew are used as setup costs.

2.2.2 Travelling costs

Travelling costs are computed per kilometre and per wagon. The costs consist of wear, fuel, and discounted maintenance costs.

2.2.3 Reclassification costs

Reclassification is an expensive, time consuming activity. Therefore it can have a major impact on the scheduling decisions. However, to compute such costs during the decision process, is a challenging and complex task. In this subsection, our approach is explained.

When a wagon needs to be placed behind a locomotive, it needs to be shunted. When a group of wagons are in a row and needed to be placed behind the same locomotive, this action requires only one shunting move. The process of shunting wagons from arriving trains to departing trains is called reclassification. It is inevitable that wagons are shunted at their departure. This is independent of the destination. Therefore we can exclude these first times from our cost function, since it will not influence the result. Shunting costs are independent of the location where it takes place. Each node in the network can be an origin or destination for some wagons. The fact that wagons can be clustered for a part of their route while they have different origins and destinations, forces us to calculate the number of reclassification moves for every part of a route instead of per wagon travelling from origin to destination. We will illustrate our solution with the example in Figure 5. This example is constructed to explain the determination of the reclassifications costs. It is not a realistic example, because the flows are very straightforward and there is only one solution. In general determining the flows is already difficult.

In the example there is only one EWPP, which has three base locations. Each base location has at least one service location. The service locations on the left, have a surplus in wagons, while the locations on the right have a deficit. The deficits are indicated with negative numbers. The rectangles have different colours for different wagon types and contain a number corresponding to the number of wagons. Base location 2 has also a surplus of two green wagons and the EWPP can supply the missing yellow wagon. For the example, we set the train capacity to five wagons.

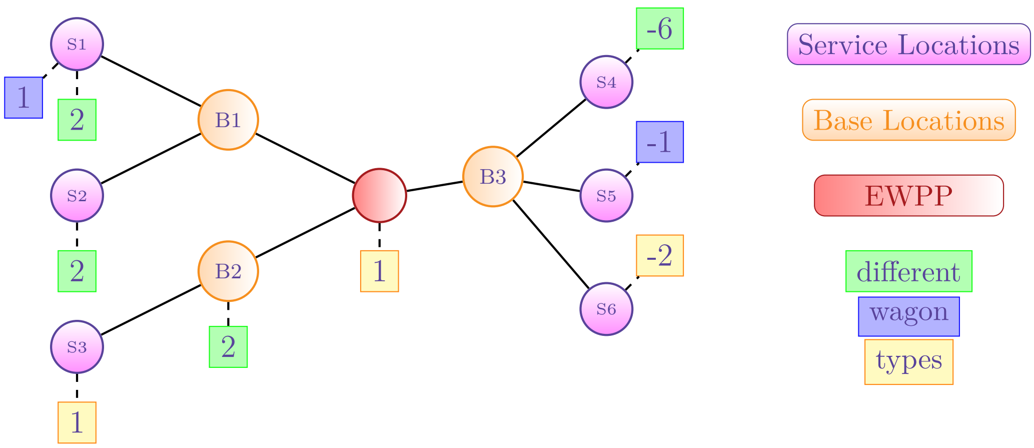

Figure 5: Shunting example

We will consider each node where reclassification takes place separately. We first look at service location S1. There is one blue wagon and two green ones. They all need to be shunted onto the train to base location B1. Since we do not count the start shunting moves, we count zero shunt actions at this location. This is the same for service locations S2 and S3. The first shunting moves that will be counted take place at B1, where the trains from S1 and S2 arrive. This can be seen in Figure 6. Although not all wagons have the same destination, the next station is for all wagons the same. Since the total amount of wagons does not exceed the capacity of one train, both arriving trains will be shunted to the same train, bringing the total number of reclassification moves to two.

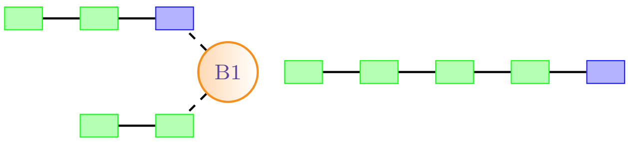

Figure 6: Shunting at B1

The next node is base location B2. There is only one train with one wagon arriving. This wagon needs to be shunted to the next departing train. On B2 there are also two green wagons waiting. These will be shunted onto the same train, but this is their first shunting, which we will ignore because it will not have influence on the decision process. Therefore only one shunting move will be counted at B2 bringing the total number of reclassifications to three. This can be seen in Figure 7.

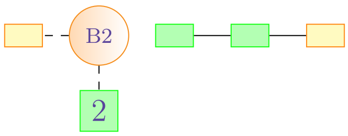

Figure 7: Shunting at B2

Both trains from B1 and B2 will arrive at the EWPP. Since both trains also need to go to base location B3, they only need to be shunted to the track to B3. This requires one shunting move per train. Also a yellow wagon is added to the train originating from B2. This is displayed in Figure 8. Because this is the first shunting of the wagon, it can be ignored. This brings the total amount of shunting moves to $3 + (1+1) = 5$.

In the current example, we assume that there are exactly enough wagons in the network available to cover the demand. Another option is to assume that the

EWPP has enough wagons of every type to cover all demand. In that case, the arriving trains will stay there while other wagons of the same type will be sent to the destination. In practice this might be even better, since all wagons can depart at the same time. In that case no shunt actions will be counted at the EWPP, because all shunting moves are start or end moves and need to be done anyway. Such decision is up to the planners of railway companies. For now, we continue assuming that the EWPP only has one yellow wagon and we need to shunt all incoming trains.

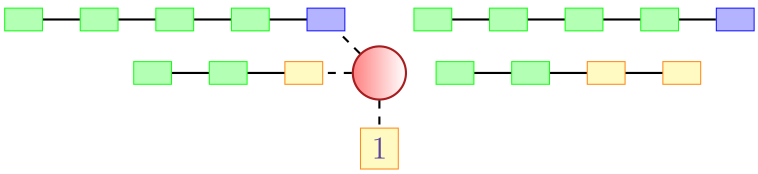

Figure 8: Shunting at the EWPP

At B3 both trains need to be split. Train 1, with four green wagons and a blue one, needs to be split into two trains. One train goes to service location S4 and one to S5. This requires two reclassification moves. The other train contains two green and two yellow wagons. The yellow ones will be shunted to a train to service location S6. The total number of reclassifications is now eight. The other two green ones, need to go to S4. Since there is already a train going to S4, it would be beneficial to shunt the two wagons onto that train. Unfortunately, this is not possible, because there are already four wagons on the train and the capacity is limited to five. When this train is filled to its maximum, we need to shunt one wagon to the existing train and one to a new train, which results in two shunting moves. Since we already need the new train, we can as well put both wagons on that new train which requires only one shunt move. In both cases two trains are needed, but costs can be minimized by not splitting the two wagons. This is visualized in Figure 9, which is the last location of the example.

Our procedure to estimate the reclassification costs considers each incoming train and counts the number of destinations inside the train. Not the final destinations, but the next stations. We simply sum this number over all trains arriving on a location. This is calculated for each location in the network. Based on the proposed estimation method, nine reclassification moves were needed for the entire network.

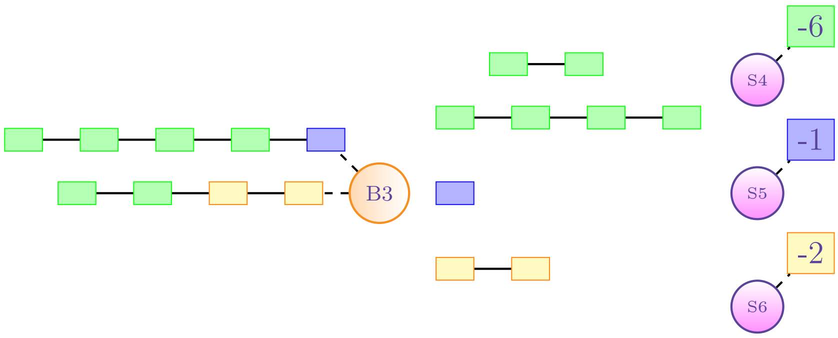

Figure 9: Shunting at B3

In the case of Figure 9, our solution approach estimates the number of reclassification moves at B3 (which is four) correctly. Unfortunately, we cannot assume that the estimation is always correct. A counterexample can be seen in Figure 10.

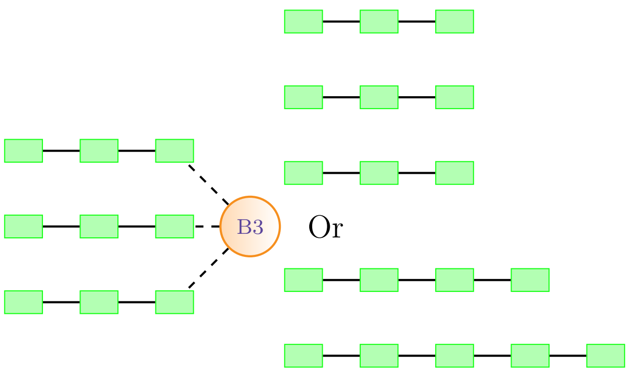

Figure 10: Shunting dilemma

We have three arriving trains with three wagons each. All wagons need to go to the same location. Again reclassification costs can be minimized by not splitting the wagon groups, but in this case this will lead to three trains and three reclassifications. Depending on the difference in costs between one extra train and one extra reclassification move, a planner might want to split one incoming train and dividing the two parts over the remaining two trains, resulting in two trains and four reclassifications. However, in the last case the total number of reclassification moves cannot be estimated easily. Finding the optimal reclassification solution for one location, is equal to solving the bin packing problem. Wagons within the same train, with the same next station, can be seen as one block. All blocks with the same next station should be divided over the outgoing trains, while minimizing the number of trains. This is the bin packing problem. In our case however, blocks might be cut into smaller ones, but only if this will lead to less trains. Within our bin packing problem, cuts are allowed, but this results in extra costs. Since the bin packing problem is proven to be NP hard (Martello and Vigo, 1998), we would rather estimate the number of reclassifications to simplify the problem. In the shunting dilemma of Figure 10, our estimation will lead to three reclassifications, where four might be needed. This depends on whether an extra train is cheaper than an extra reclassification move. Since the cost function is only used to compare different options and choose the best of them, we believe that this simplification is acceptable. Solving the bin packing problem is only possible when the numbers of incoming trains and wagons are known. Such information is only available when train schedule is complete. This is not the case in this thesis, since we want to take the reclassification costs into account during the decision process and not afterwards. Therefore implementing the bin packing problem is not possible.

2.3 Objective

The objective of this thesis is to find a method which solves the repositioning problem in reasonable time, while minimizing the total costs. In the railway network, companies supply or demand empty freight wagons. Repositioning these wagons to other companies instead of to the depot saves time and effort. A solution for this problem contains information about which wagons need to be repositioned, to which location and on which day. Planners should be able to run the optimization overnight. For realsize problem instances containing seven days to plan, within reasonable time means within at most twelve hours.

3 Literature study

Within the transport business, the repositioning problem has been tackled in various ways. In this chapter, different approaches are reviewed. First we will look at articles that contain a more general approach to the problem. This means that they contribute to the empty repositioning research, but not solve empty train wagons repositioning problems, or empty container repositioning problems. After that, we will look at the problem within shipping and within empty container repositioning problems. Next to that, empty repositioning in the rail environment is reviewed. Finally a conclusion is given which summarizes how the reviewed literature can be used within this thesis.

3.1 General approaches

An extensive overview of optimization models for freight transportation is done by Crainic (2002). He mentions empty movement problems but it is not the main topic of the paper. He shows a solution method which solves the empty repositioning on an operational level with a dynamic network representation for a certain horizon. The used multi-commodity flow approach schedules multiple network problems which in some cases contain empty wagons. Although we have a static network and we do not schedule normal demands, the use of a multi-commodity flow approach can also be applied to our problem.

A capacitated network flow model is used by Olivo et al. (2005). They focus on empty container repositioning instead. This model has a rolling horizon and also considers road and water transport options. They solved it for small and medium sized problem instances, because large problems took too much computation time. Shintani et al. (2005) use a similar network flow model, but solve it with a genetic algorithm heuristic. Their method converges within a minute for a medium problem size. Their algorithm was not tested on large problem instances, but the authors stated that computation time will increase consequently. From both articles it can be learned that for large problems, a heuristic is required to solve the network flow problem.

Godfrey and Powell (2002) use an adaptive dynamic programming algorithm that uses non-linear functional approximations for dynamic fleet management. Their functional approximations are piecewise-linear and provide integer solutions. They assume that random scenarios arise over time and solve it with a stochastic version of a dynamic resource allocation method. Because we plan the model week which does not contain any uncertainty, this model does not apply for us.

Zhang and Facanha (2014) consider a model which incorporates the possibility to transport empty containers from one port to another using rail transport. They used a network flow problem, which they split into smaller regions. They use regional, inter-regional, and national levels. This reduces computation times and does not lead to infeasibility. This approach of splitting the overall country into smaller regions can also be used in our problem.

3.2 Shipping approaches

A substantial amount of repositioning research on empty repositioning is done within the container liner shipping service. Because such networks have less locations and commodities compared to rail networks, the models cannot be used directly. But since the models are similar, the computation time and performance of the heuristics are interesting.

Dong et al. (2013) reviewed multiple articles about the mathematical modelling in maritime empty container repositioning solutions. They state that there are two groups of repositioning policies; state-feedback control policies and origin-destination matrix solutions. Because they only compare shipping networks subject to uncertainty with just three shipping service routes, the only contribution to this research is that computation times for optimal models are huge for realistic problem sizes. Again, this implies that a heuristic approach is required.

Dong and Song (2009) present a simulation based optimization tool to seek the optimal number of containers and the optimal threshold parameters which are used for empty repositioning. Combined with the simulation, they developed an evolutionary optimization algorithm based on Genetic Algorithms and Evolutionary

Strategy. They achieve cost reductions of more than 15% in computation times less than half an hour.

In the article of Erera et al. (2009), empty repositioning is optimized for a homogeneous fleet. They used a time-space network flow problem to solve the dynamic empty repositioning. A set of uncertainty is included in the model to forecast future supplies and demands. The main focus is to deal with uncertainty in an optimal way. This is also done by Francesco et al. (2009). They use stochastic programming for a similar problem. A sample average approximation method is applied by Long et al. (2012). They formulate a two-stage stochastic program model with random demand to incorporate uncertainties. In all three mentioned articles the focus is on dealing with the uncertainties within an operational week, while optimizing the empty repositioning for containers.

Song and Carter (2009) counter the repositioning problem by proposing four different strategies and solving each strategy with a mathematical program. They focus on possible cooperation between corporations within the liner shipping industry. According to them, sharing containers can reduce costs with 12 to 18%. But to solve the problem within the different liner companies, more research is required. Brouer et al. (2011) approach the empty repositioning problem for a liner shipping company with an arc-flow formulation which is decomposed using the Dantzig-Wolfe principle to a path-flow formulation. A linear relaxation is used with a delayed column generation algorithm. A feasible integer solution is found by rounding the fractional solution and adjusting flow balance constraints with leased containers. The proposed algorithm maximizes profit and is able to solve instances with 234 ports, 16,278 demands over 9 time periods in 34 minutes. The integer solutions found by rounding down are computed in less than 5 seconds and the gap is within 0.01% from the upper bound of the linear relaxation.

Meng and Wang (2011) consider a conventional hub-and-spoke network design in which the transfer costs at hub nodes are not accounted for. Transfer costs could be similar to reclassification cost, but in this paper such cost are neglected. They solved the empty repositioning problem with a mixed-integer linear programming model, without any heuristic. They solved the problem for only one shipping line and state that some meta-heuristic can be applied if the problem size increases. This optimal model for a smaller problem size is quite the same approach as the standard multi-commodity network approach of Chao and Yu (2012). Both give optimal results, but are not usable for larger problem sizes.

Song and Dong (2012) consider joint cargo routing and empty container repositioning at the operational level for a shipping network with multiple service routes. They propose two solution methods. The first is a two-stage shortest-path based integer programming method, which combines a cargo routing algorithm with an integer programming model of the dynamic system. The second is a two-stage heuristic rules-based integer programming method, which combines an integer programming model of the static system with a heuristic implementation algorithm in a dynamic system. They conclude that for a small case the heuristic method is 3.3% worse than the optimal method. For a realistic large case, the optimal method was not able to produce the solution due to computational complexity.

Choong et al. (2002) present an integer programming problem for empty container repositioning. They use an underlying space-time network to optimize the container flow between nodes. They do not redirect containers, but minimize the transports by planning the empty containers on passing barges which have some capacity left. Although this model cannot be applied in our case, it is interesting to see that doubling the horizon from 15 to 30 days, does not lead to extensive solution times.

3.3 Rail approaches

Although empty wagon repositioning is also studied within the railway environment, our specific situation has not been studied before. Reclassification costs are most of the time not included and a hub-and-spoke network has never been used for empty repositioning in the rail industry. Nevertheless, the used approaches can be useful for this thesis.

Evdokia et al. (1993) use game theory to approach the problem. They state that like an oligopolistic equilibrium system, the multi-commodity multi-railroad problem could fit within the frameworks of the Nash-Cournot non-cooperative game; a leader-follower Stackelberg game. They model the n-player game with a mathematical formulation. Although empirical tests have shown that the algorithm converges within reasonable computational time, a theoretical study has shown that the conditions for existence and uniqueness of an equilibrium solution are very strong and cannot be satisfied in a general class of problem instances. It is interesting to see that a problem similar to ours, is solved using Game Theory. However, the fact that theoretically the equilibrium solution might not exist, makes this approach unusable for practical implementation. Railway companies can only use a model which will always return a solution.

Sherali and Tuncbilek (1997) compare static and dynamic models for calibration of strategic planning models for the multi-level rail-car fleet management. The dynamic model uses time-space networks, which might be interesting for this thesis, because they state that their decomposition heuristic recovers optimal solutions with a reasonable effort. But Sherali and Tuncbilek (1997) use the models to determine the rail-car fleet size, not the allocations. The fleet sizing problem is contained within our problem, so the success of a decomposition heuristic can be questioned, since their computation time was already around 700 to 1000 seconds.

The model of Holmberg et al. (1998) can be characterized as a time-expanded multi-commodity network flow problem with integer requirements, in which each commodity corresponds to one type of freight car. It also includes common restrictions on the train capacities for cars of different types; one of the key properties of this model. Next to that, the railway system is divided into 13 car distribution areas. They solve the multi-commodity flow problem with time and capacity constraints with a Lagrangian heuristic. They take exact time tables explicitly into account. This model shows that it is solvable in reasonable time for problems of realistic size. The model generates operationally implementable plans. They suggest the following possible extensions: group substitutable wagons, split capacity in weight and length, inventory costs in objective function and movements of loaded freight cars are optimized simultaneously as the empty freight car. Since we would like to group substitutable wagons and we are not interested in operational implementable plans, this model cannot be used directly. Splitting the railway system into different areas might be a useful solution.

Ireland et al. (2004) have developed a five step iterative process. They start with a traffic forecast, after which they design a blocking plan. The blocking plan is the foundation for the operating plan, determining the car routings, yard workloads, and contribution to customer service. This blocking plan creation is also an iterative process. After the blocking plan design, the train schedules are built. Again, an iterative procedure is used. Step four of the overall iterative process is a day-of-week simulation. Based on the results of this simulation, the blocking plan and/or the train schedule is adjusted. Finally, they finish the process with a crew and locomotive plan. This model does not focus on empty repositioning, but the iterative approach using simulation is an interesting way of tackling large problems.

Joborn et al. (2004) encounter the same cost structure as we do. They identify three different costs. Costs that do not depend on the transportation pattern design are excluded. They declare train costs as the combination of engine costs, driver costs, and energy costs. The paper aims at an operational planning and can therefore ignore the fixed train costs. For the transportation costs, a certain amount per mile is used. It is interesting to see that they cannot find an exact solution for the reclassification costs, which they call yard costs. A set covering method is used where paths are selected. A path is a possible route for a group of wagons with the same origin and destination. At each station in between, yard costs are counted. This approach neglects the fact that some paths share parts of their route. On the shared parts of their routes, all wagons can be handled as one block. Neglecting this fact makes the computation easier but less accurate. Although their approach is less accurate, the article states that this simplification is verified in Joborn (2001). The problem is solved using a tabu search heuristic on a capacitated network flow model. For the reclassification, a similar assumption is made by Narisetty et al. (2008), they state that fixing reclassification costs based on the number of stations between an origin and destination will not compromise the outcome of the model. In our hub-and-spoke network, lots of traffic can be expected between EWPPs. Neglecting the fact that most of the wagons can be handled as one block, does not seem to be logic in our case.

A Lagrangian based heuristic is used by Holmberg et al. (2007). They study a multi-commodity network flow problem with fixed costs on paths, not on arcs. Their case study is an empty freight car distribution on a rail network. Although the heuristic generates fairly good primal feasible solutions and lower bounds on the optimal objective function value, the computation times are quite long already for small problem sizes. They test with 25 nodes and 10 commodities. When the number of commodities is increased to 15, while the number of nodes remains 25, the computation time increases from 200 to 300 seconds. Real size problems on rail networks are much larger facing at least 200 nodes and at least 25 commodities. Although the optimality gap of this heuristic is below 25%, the computation time for our problem would probably be too long.

Narisetty et al. (2008) use an LP relaxation and state that it will return optimal integer solutions. Their approach is used on the Union Pacific rail road, the largest rail road in the United States. The total computation time is only a few minutes. They fix all costs on beforehand based on the origin and destination of the wagons and then solve the classical transportation problem of satisfying the demand of m customers from n supply points while minimizing the associated transportation costs. They do this for a short planning horizon of 10 days of demand and 1 day of supply. The main difference with our problem is that they fix the costs per wagon on beforehand, while in reality the costs depend also on movements of other wagons on nearby stations. Since they state that neglecting the reduction in reclassification costs does not lead to other results, we should take that into consideration as well.

3.4 Conclusions

Empty repositioning has been studied in different fields with many different approaches. Most articles conclude that an exact formulation will not produce the optimal solution within reasonable time for realsize problem instances. Splitting the total problem into smaller regions was successful for Holmberg et al. (1998) and Zhang and Facanha (2014). Therefore it seems to be a good solution approach for a heuristic. For the smaller region based problems, different approaches can be tried.

Where some articles state that for smaller problems the network flow problem will return optimal results within reasonable time, Narisetty et al. (2008) and Brouer et al. (2011) state that the LP relaxation will also return integer solutions. This needs to be verified for problems which are split into different regions.

Other heuristics which are used are based on: Lagrange relaxation, tabu search, game theory, sample average approximation, genetic mutation, iterative procedures or simulation. In some cases some techniques were combined. These approaches might be an option if an exact formulation and splitting regions will not return the required results. In multiple articles, simulations are used to test the solution for uncertainties. This will not be necessary for this thesis since we will plan a model week, which does not contain uncertainties.

4 Solution methods

In this section the integer programming formulation and heuristics are explained. The section starts with the introduction of the sets and variables which are needed for the IP formulation. After the introduction an overview of all sets, parameters and variables is given. Then the IP is explained. Finally three different heuristics are presented.

The integer programming formulation of the problem is based on a multi-commodity flow model. We consider a directed graph $G = (V, A)$ where all vertices $v \in V$ are locations in the hub-and-spoke network. The locations are connected with bidirectional arcs $a \in A$. We define the set $W(i, j)$ as the set of locations that can be reached directly after location j, when starting in location i. This comes down to all neighbours of location j and j itself, but excluding location i. This set is needed in order to calculate the number of reclassification moves. In a normal multi-commodity flow model, the decision variable has one index for the commodity and two for the location resulting most of the time in x_{ij}^k. In our model this will not do, due to the reclassifications. This is shown with an example. In Figure 11 node 1 and 2 will supply node 3 and 4.

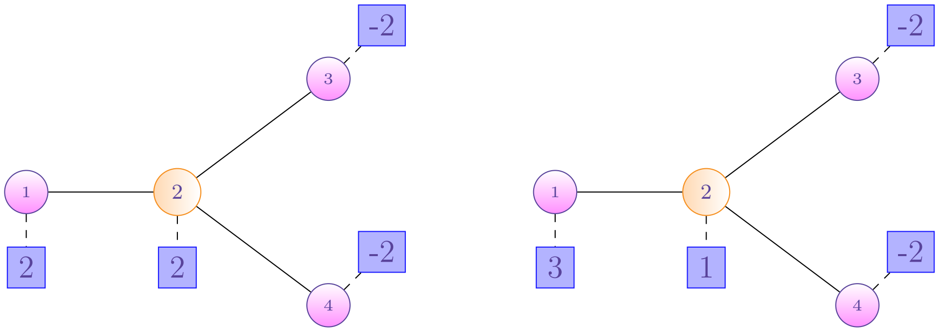

Figure 11: Explanation three indices

When using the normal notation, this will lead to $x_{12} = 2$, $x_{23} = 2$, and $x_{24} = 2$ for the left example and $x_{12} = 3$, $x_{23} = 2$, and $x_{24} = 2$ for the right one. In the left example there is one reclassification move, while the right one contains two. This cannot be derived from the standard decision variables. Therefore we introduce x_{ijl} which stands for the flow from i to l via j. In the left example this will lead to $x_{122} = 0$, $x_{123} = 2$, $x_{124} = 0$, $x_{233} = 2$, and $x_{244} = 2$. In the right example the values are $x_{122} = 0$, $x_{123} = 2$, $x_{124} = 1$, $x_{233} = 2$, and $x_{244} = 2$. $j = l$ implies that there are no reclassifications, since we do not count reclassifications at beginning and end nodes. The number of reclassifications can now be derived from the variables for which $j \neq l$ and are larger than zero. For the left one, this is 1 and for the right one this is 2. In this example only counting the non-zero variables for which $j \neq l$ is enough to determine the number of reclassification moves. When flows contain more than the capacity of one train, this need to be taken into account as well. This approach will be discussed in Subsection 4.2.

For each day in the model week it is decided which wagons will flow from one location to another. At the end of each day, most empty wagons will be used by customers and therefore disappear from the network. The unused wagons can be reassigned the next day. On the other hand, when a customer unloads cargo cars, the empty ones will (re)appear in the network. Where the standard multi-commodity flow has capacity on the arcs, our problem has no constraints on the arcs, but does have capacity constraints on each vertex. For each commodity, each day, there is a different minimum in the number of cars, which is equal to the demand. The maximum parking capacity is restricted by its length in metres.

4.1 Notation

First, the sets that are required to solve the model are described. Next the parameters are given, that should be known before solving the problem. Finally, the decision variables are listed and explained.

Sets	
V	All locations in the network
A	All arcs in the network
$W(i,j)$	All possible locations which can be reached directly after travelling from location i to j, including j (which implies staying at j)
K	All commodities/wagon types in the network
T	All periods $(1, 2, ..., \tau)$ in the planning horizon
Parameters	
$c_{s_{ij}}$	Setup costs per extra train starting in location i and travelling to location j with $(i,j) \in A$
$c_{t_{ij}}$	Costs of travelling between location i and j which occur per wagon, with $(i,j) \in A$
c_r	Costs per reclassification
$U^k_{dem_{(i,t)}}$	The minimum required number of wagons of type k at location i at the end of period t
$U_{park_{(i,t)}}$	The maximum parking space for empty wagons in metres at location i in period t. Loaded wagons can also use parking capacity. Therefore the maximum parking space can change per time period
$U_{cap_{(ij)}}$	The maximum capacity of one train on the track from location i to location j with $(i,j) \in A$
L_k	The length of the car type k in metres
$z^k_{i,t}$	The number of empty wagons of type k that will become available and (re)appear in the problem at the beginning of period t on location i
ϵ	A number which is smaller than the smallest cost parameter of the problem, but significantly larger than 0.
τ	The last period considered in the planning horizon
Decision variables, all in $\mathbb{N}$	
$x^k_{ijl,t}$	The number of cars of type k travelling from location i to l via j in period t, with $(i,j) \in A$, and $l \in W(i,j)$
$y^k_{i,t}$	The number of wagons of type k, at location $i \in V$ at the beginning of period t
$\gamma_{ij,t}$	An auxiliary variable used to determine the setup costs
$\delta_{ijl,t}$	An auxiliary variable used to determine the reclassification costs

4.2 Integer programming problem

The objective of the formulation is to minimize the overall costs of the problem. The overall costs are the sum of the setup, travelling, and reclassification costs. The setup costs occur when a new train is needed. This can be calculated by summing all departing wagons of one location in a specific direction at a specific day, divide the sum by the capacity of one train and rounding the answer up to the integer above. This integer is the number of leaving trains and is multiplied by the setup costs per train. This can be seen in Equation (1).

$$\min \sum_{t \in T} \sum_{(i,j) \in A} c_{s_{ij}} \left\lceil \frac{\sum_{k \in K} \sum_{l \in W(i,j)} x^k_{ijl,t}}{U_{cap_{(ij)}}} \right\rceil \tag{1}$$

s.t.

$$x^k_{ijl,t} \in \mathbb{N} \quad \forall \quad k \in K,\, (i,j) \in A,\, l \in W(i,j) \text{ and } t \in T$$

Unfortunately, this formulation is not linear. To linearise this part of the objective function, we use an additional decision variable $\gamma_{ij,t}$. This variable can be interpreted as the number of trains leaving from location i to j at time t. $\gamma_{ij,t}$ is minimized, while we add a restriction to the model that forces $\gamma_{ij,t} \times U_{cap_{(ij)}}$ to be at least as large as the sum of the leaving wagons. This results in Equation (2) and Equation (3).

$$\min \sum_{t \in T} \sum_{(i,j) \in A} c_{s_{ij}} \gamma_{ij,t} \tag{2}$$

s.t.

$$\sum_{k \in K} \sum_{l \in W(i,j)} x^k_{ijl,t} \leq U_{cap_{(ij)}} \gamma_{ij,t} \qquad \forall t \in T, \forall (i,j) \in A \tag{3}$$

$$\begin{aligned} x^k_{ijl,t} &\in \mathbb{N} \quad && \forall \quad k \in K,\, (i,j) \in A,\, l \in W(i,j),\, t \in T \\ \gamma_{ij,t} &\in \mathbb{N} && \forall \quad (i,j) \in A,\, t \in T \end{aligned}$$

The travelling costs are more straightforward. This is the flow per arc multiplied by the costs for that arc, for every arc and every period in the planning horizon. This results in Equation (4):

$$\min \sum_{t \in T} \sum_{(i,j) \in A} c_{t_{ij}} \times \sum_{l \in W(i,j)} \sum_{k \in K} x^k_{ijl,t}. \tag{4}$$

The modelling of the reclassification costs is similar to the setup costs. For the setup costs, we were only interested in the flow from location i to j, no matter where the wagons were going after j. This is different for the reclassification costs. In this thesis the number of reclassifications is determined with a simplified approach. For each incoming train the number of different destinations for all wagons of the train is counted. To determine the number of reclassification moves at j, information about the next location is required. Compared to the setup costs, a similar formulation is used. Auxiliary variable $\delta_{ijl,t}$ is used, which is the number of reclassification moves that takes place at location j for wagons coming from location i and travelling to location l at period t. $\delta_{ijl,t} \times c_r$ is minimized in the objective function. The sum of all leaving wagons with the same destination and the same destination afterwards, should be smaller than the capacity of one train times the number of reclassification moves. If the number of wagons exceeds the capacity an extra reclassification move is necessary. The restriction of Equation (5) is added to the model for all $t \in T$, $(i,j) \in A$, and $l \in W(i,j) \backslash j$.

$$\sum_{k \in K} x^k_{ijl,t} \leq U_{cap_{(ij)}} \delta_{ijl,t} \tag{5}$$

When $l = j$, x_{ijj} implies staying at location j. Since we do not count start and end reclassifications, element j is removed from set $W(i,j)$. This removal could lead to a more accurate estimation of the number of reclassification moves, but this requires an additional cost parameter. This is explained with an example. Again we look at the left example of Figure 11. All relevant decision variables are x_{122}, x_{123}, x_{124}, x_{233}, and x_{244}. We assume that all setup costs and travelling costs are equal. When we solve the example, a possible solution is: $x_{122} = 0$, $x_{123} = 2$, $x_{124} = 0$, $x_{233} = 2$, and $x_{244} = 2$, which will lead to overall costs of $3 \times c_s + 6 \times c_t + 1 \times c_r$. However, x_{122} stands like x_{123} for the flow from location 1 to location 2. We would like to use x_{122} for flow staying at location 2, but this is not explicitly defined. The solutions: $x_{122} = 2$, $x_{123} = 0$, $x_{124} = 0$, $x_{233} = 2$, and $x_{244} = 2$, will lead to costs of $3 \times c_s + 6 \times c_t + 0 \times c_r$. Because we do not count end reclassifications, we do now count zero of them, which makes this solution cheaper than the correct one. In this example we can count end reclassifications as well to deal with this problem. However, this is not a valid solution in general. When the costs of regular reclassifications are exactly the same as end reclassifications, a solution might only contain the end reclassifications. In that case, the exact reclassification costs, cannot be estimated any more. This will result in a problem neglecting the fact that reclassifications take place. Our solution is to force $\sum_{k \in K} x^k_{ijj,t}$ to be only nonzero if there is no other option (i.e. if j is the final location) by assigning costs of $c_r + \epsilon$ to all $x^k_{ijj,t}$. The incorrect solution of $x_{122} = 2$, $x_{123} = 0$, $x_{124} = 0$, $x_{233} = 2$, and $x_{244} = 2$, will now lead to $2 \times \epsilon$ more costs than the correct solution. This approach is realized by adding $\sum_{t \in T} \sum_{(i,j) \in A} (c_r + \epsilon) \times \sum_{k \in K} x^k_{ijj,t}$ to the objective function. This brings the objective function with corresponding constraints to Equation (6).

$$\min \sum_{t \in T} \sum_{(i,j) \in A} \left(c_{s_{ij}} \gamma_{ij,t} + (c_r + \epsilon) \times x^k_{ijj,t} + c_{t_{ij}} \times \sum_{l \in W(i,j)} \sum_{k \in K} x^k_{ijl,t} + c_r \times \sum_{l \in W(i,j) \setminus j} \delta_{ijl,t}\right) \tag{6}$$

s.t.

$$\begin{aligned}
\textstyle\sum_{k \in K} \sum_{l \in W(i,j)} x^k_{ijl,t} \leq U_{cap_{(ij)}} \gamma_{ij,t} \quad & \forall \quad (i,j) \in A,\ t \in T \\
\textstyle\sum_{k \in K} x^k_{ijl,t} \leq U_{cap_{(ij)}} \delta_{ijl,t} \quad & \forall \quad (i,j) \in A,\ l \in W(i,j),\ t \in T \\
x^k_{ijl,t} \in \mathbb{N} \quad & \forall \quad k \in K,\ (i,j) \in A,\ l \in W(i,j),\ t \in T \\
\gamma_{ij,t} \in \mathbb{N} \quad & \forall \quad (i,j) \in A,\ t \in T \\
\delta_{ijl,t} \in \mathbb{N} \quad & \forall \quad (i,j) \in A,\ l \in W(i,j) \setminus j,\ t \in T
\end{aligned}$$

To ensure that all variables behave as they should, another constraint is needed. This is shown with the right example of Figure 11. A possible solution is to relocate two wagons of location 1 to location 3, one wagon from location 1 to location 4 and one from location 2 to location 4. This should lead to the following values: $x_{122} = 0$, $x_{123} = 2$, $x_{124} = 1$, $x_{233} = 2$, and $x_{244} = 2$, which will lead to overall costs of $3 \times c_s + 6 \times c_t + 2 \times c_r$. Unfortunately, setting the variables to $x_{122} = 0$, $x_{123} = 3$, $x_{124} = 0$, $x_{233} = 2$, and $x_{244} = 2$, will lead to overall costs of $3 \times c_s + 6 \times c_t + 1 \times c_r$. The variable $x_{123} = 3$ should mean that there are three wagons from location 1 are going to location 2, because they will go to location 3 after that. However, this is not strictly stated in the model. The flow from location i, via j to l, should be continued as flow from location j, via l to m, with $m \in W(j, l)$. Therefore the restriction of Equation (7) is added. This will force the flow x_{123} to be continued to x_{23m} where m can be location 3 or further locations.

$$x^k_{ijl,t} \leq \sum_{m \in W(j,l)} x^k_{jlm,t} \qquad \forall (i,j) \in A, k \in K, l \in W(i,j) \backslash j \quad (7)$$

To set the correct stock level ($y^k_{i,t}$) for each location, commodity and each time period, the current stock of each commodity is updated with new available wagons ($z^k_{i,t}$), the demanded wagons which will become unavailable as empty wagons ($U^k_{dem_{(i,t)}}$), the wagons entering the location and leaving the location. This results for every location $h \in V$ in Equation (8). Because wagons can be stored at each location, the constraints of Equation (8) can be seen as flow conservation constraints as well as inventory constraints.

$$y^k_{h,t} + z^k_{h,t} - U^k_{dem_{(h,t)}} + \sum_{(i,h) \in A} \sum_{l \in W(i,h)} x^k_{ihl,t} - \sum_{(h,j) \in A} \sum_{l \in W(h,j)} x^k_{hjl,t} = y^k_{h,t+1} \forall h \in V, k \in K, t \in T \backslash \tau \quad (8)$$

In the first period, all stocks are set to zero. If there are wagons available, they will be added using the $z^k_{h,t}$ parameter. This adds the following constraints to the problem.

$$y^k_{h,1} = 0 \qquad \forall h \in V, k \in K \quad (9)$$

At the end of each day, demand should be met. Again we add new available wagons, incoming wagons, and leaving ones to the stock of each location, but now it should be larger or equal to the demand. This gives Equation (10)

$$y^k_{h,t} + z^k_{h,t} + \sum_{(i,h)\in A} \sum_{l\in W(i,h)} x^k_{ihl,t} - \sum_{(h,j)\in A} \sum_{l\in W(h,j)} x^k_{hjl,t} \geq U^k_{dem_{(h,t)}} \quad \forall h \in V, k \in K, t \in T \tag{10}$$

Although Equation (10) is equal to $y^k_{h,t+1} \geq 0$, this is not defined for the last period. Therefore we will not use the shorter notation.

On each location the parking space is limited. The sum over all wagons multiplied by their length(L_k) should be smaller than or equal to the maximum parking space($U_{park_{(h,t)}}$). This adds Equation (11) to the model.

$$\sum_{k\in K} (y^k_{h,t} + z^k_{h,t} - U^k_{dem_{(h,t)}} + \sum_{(i,h)\in A} \sum_{l\in W(i,h)} x^k_{ihl,t} - \sum_{(h,j)\in A} \sum_{l\in W(h,j)} x^k_{hjl,t}) \times L_k \leq U_{park_{(h,t)}} \forall h \in V, t \in T \tag{11}$$

A complete overview of the model can be found on the next page.

Model overview

$$\min \sum_{t\in T}\sum_{(i,j)\in A}(c_{s_{ij}}\gamma_{ij,t}+(c_r+\epsilon)\times x^k_{ijj,t}+c_{t_{ij}}\times\sum_{l\in W(i,j)}\sum_{k\in K}x^k_{ijl,t}+\sum_{l\in W(i,j)\setminus j}c_r\times \delta_{ijl,t})$$

s.t.

$$\begin{array}{rcll}
\sum_{k\in K}\sum_{l\in W(i,j)}x^k_{ijl,t} & \leq & U_{cap_{(ij)}}\times\gamma_{ij,t} & \forall\quad (i,j)\in A,\, t\in T\\
\sum_{k\in K}x^k_{ijl,t} & \leq & U_{cap_{(ij)}}\times\delta_{ijl,t} & \forall\quad (i,j)\in A,\, l\in W(i,j),\, t\in T\\
y^k_{h,t}+z^k_{h,t}-U^k_{dem_{(h,t)}} & & & \\
+\sum_{(i,h)\in A}\sum_{l\in W(i,h)}x^k_{ihl,t} & = & y^k_{h,t+1} & \forall\quad h\in V, k\in K, t\in\{1,2,...,\tau-1\}\\
-\sum_{(h,j)\in A}\sum_{l\in W(h,j)}x^k_{hjl,t} & & & \\
y^k_{h,t}+z^k_{h,t}-U^k_{dem_{(h,t)}} & & & \\
+\sum_{(i,h)\in A}\sum_{l\in W(i,h)}x^k_{ihl,t} & \geq & 0 & \forall\quad h\in V, k\in K, t\in T\\
-\sum_{(h,j)\in A}\sum_{l\in W(h,j)}x^k_{hjl,t} & & & \\
\sum_{k\in K}(y^k_{h,t}+z^k_{h,t}-U^k_{dem_{(h,t)}} & & & \\
+\sum_{(i,h)\in A}\sum_{l\in W(i,h)}x^k_{ihl,t} & \leq & U_{park_{(h,t)}} & \forall\quad h\in V, t\in T\\
-\sum_{(h,j)\in A}\sum_{l\in W(h,j)}x^k_{hjl,t}) & & & \\
\times L_k & & & \\
x^k_{ijl,t} & \leq & \sum_{m\in W(j,l)}x^k_{jlm,t} & \forall\quad (i,j)\in A,\, k\in K,\\
 & & & \quad\;\; l\in W(i,j)\setminus j,\, t\in T\\
y^k_{h,1} & = & 0 & \forall\quad h\in V, k\in K\\
x^k_{ijl,t} & \in & \mathbb{N} & \forall\quad k\in K,\, (i,j)\in A,\\
 & & & \quad\;\; l\in W(i,j),\, t\in T\\
y^k_{i,t} & \in & \mathbb{N} & \forall\quad k\in K,\, i\in V,\, t\in T\\
\gamma_{ij,t} & \in & \mathbb{N} & \forall\quad (i,j)\in A,\, t\in T\\
\delta_{ijl,t} & \in & \mathbb{N} & \forall\quad (i,j)\in A,\, l\in W(i,j),\, t\in T
\end{array}$$

4.3 Heuristics

In this section, the three heuristics are explained. All three split the problem and solve it day by day. This approach will reduce the complexity of the overall problem, but will also lead to sub optimal solutions. When the planning horizon is only one day, the lack of information about other days makes the problem less complex and easier to solve. This will lead to faster computation times. However, the lack of information makes it impossible to combine trips of multiple days. This will lead to extra trips and to sub optimal solutions.

For each subproblem, the IP will find moves. A move contains information about the wagon type, the number of wagons, the start location and the next location. If 4 wagons of the same type are repositioned from A to D, via B and C, will this result in 3 moves (AB, BC, and CD).

4.3.1 Heuristic 1

Heuristic 1 solves the problem for each day separately, using the IP as described in Section 4.2. The heuristic starts with the available wagons of day 1 and the demanded wagons of day 2. Once the IP has solved the problem for day 1, the obtained moves are updated by setting the wagon inventories of all locations. The next iteration, day 2 has inventory as well as new available wagons. This combination is used to satisfy the demand of day 3. This iterative process continues until the demands of all days are satisfied.

4.3.2 Heuristic 2

Like heuristic 1, heuristic 2 splits the problem per day. On top of that, heuristic 2 uses a divide-and-conquer method and splits the network into regions. In the hub-and-spoke network, base locations can only be reached from their parent EWPP, or from their child service locations. Heuristic 2 uses these facts to speed up the calculations. Instead of solving the problem for the entire network, the problem is first solved using only EWPP-regions. These regions contain all demand, supply, and parking capacity of all underlying nodes of one EWPP. With the regions, a new network is created containing only a few nodes (the same amount as there are EWPPs). This is visualised in Figure 12.

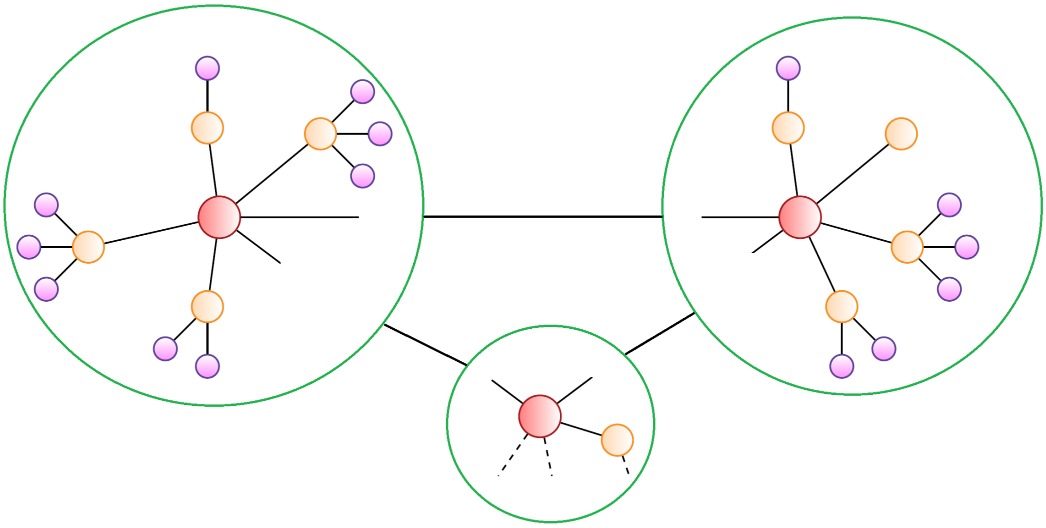

Figure 12: EWPP region overview

The new network is solved with the exact formulation. When this sub problem is solved, wagons are supplied and delivered at and from the EWPPs. After their stock is updated, the problem is solved locally. This will be explained with an example.

When in a region, say 1, wagons of type A are available on a service location, the region can supply these to another region, say 2. In region 2 there is a service location which needs wagons of type A. This can be seen in Figure 13.

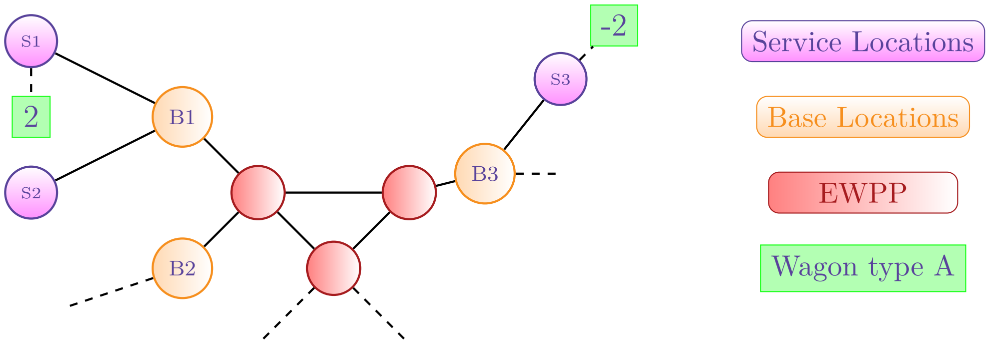

Figure 13: Example Heuristic 2 - overall problem

The first step is to create regions which supply and require all the wagons inside the region. This problem is visualised in Figure 14.

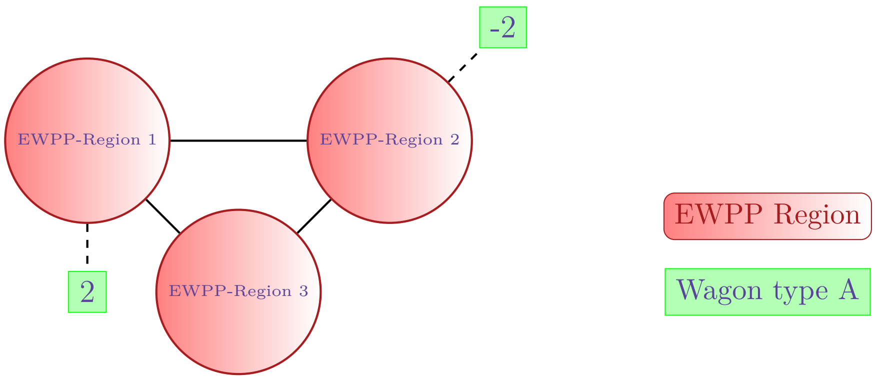

Figure 14: Example Heuristic 2 - region problem

After the region problem is solved, EWPP region 1 has supplied wagons of type A, while it is not available at the EWPP location itself. The next step of the heuristic, is to solve each region individually, again using the exact formulation. Now, the EWPP location of region 1, requires wagons of type A instead of supplying it. This is drawn in Figure 15.

Figure 15: Example Heuristic 2 - problem region 1

When region 1 is solved, wagon type A is moved from the service location, to the EWPP of region 1. Now the heuristic solves the second region individually. The EWPP of region 2 supplies now wagons of type A, which are moved to their destination. This problem is displayed in Figure 16.

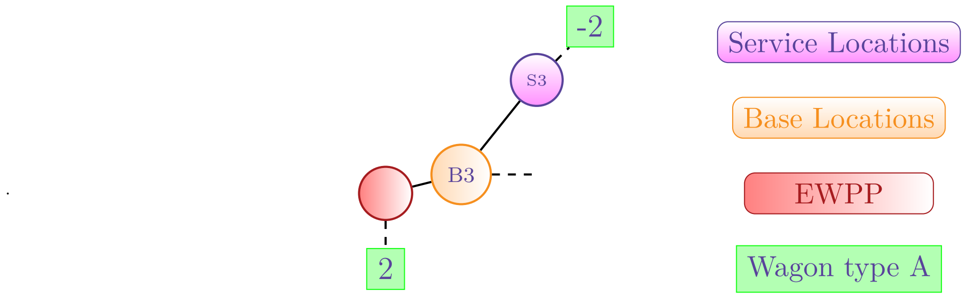

Figure 16: Example Heuristic 2 - problem region 2

4.3.3 Heuristic 3

Heuristic 3 also splits the problem per day and also uses the divide-and-conquer approach of heuristic 2. The difference is the way of solving the individual EWPP region problems. Where heuristic 2 uses the exact formulation to solve the individual EWPP regions, heuristic 3 uses again a divide-and-conquer method. When solving the same example as discussed with heuristic 2 in the previous section, heuristic 3 solves the problem of Figure 15 by creating new regions for each base location. These regions contain all demand, supply, and parking capacity of all connected service locations. The heuristic solves the EWPP region then as displayed in Figure 17 with the exact formulation.

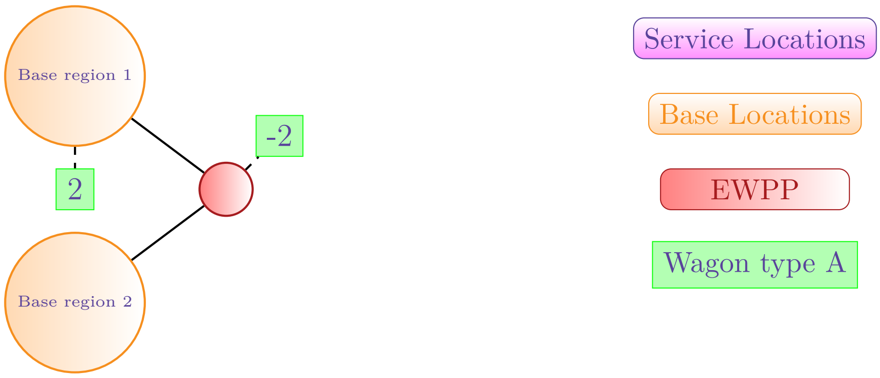

Figure 17: Example Heuristic 3 - problem EWPP region 1

After solving the EWPP region, all base regions are solved using the same principle. All the wagons that the region supplied or demanded, are placed on the base location. If they are not available there, it becomes demand for the new sub problem. If they are not needed at the base location, the wagons become supply. Such sub problem contains now only one base location and its corresponding service locations. This can be seen in Figure 18. Like all sub problems, this one is also solved using the exact formulation.

Figure 18: Example Heuristic 3 - problem Base region 1

5 Performance analysis

In this section the performance of the IP formulation and the heuristics are analysed. In Section 5.1 the data that is used to analyse the performance is discussed. The performance of the IP is analysed in Section 5.2. In Section 5.3 the different heuristics are tested and discussed. All tests are performed on a laptop with 8 GB RAM memory and an Intel(R) Core(TM) i7-4800MQ cpu @2.7GHz. The IP and heuristics are programmed in the Quintiq software package and solved with CPLEX 12.5.

5.1 Data

Ab Ovo has made customer data available for this thesis. The network can be interpreted as a perfect hub-and-spoke network. There are 3 EWPPs, 23 base locations, and 199 service locations. Similar wagon types are combined in a wagon type replacement group (WTRG). For the data only WTRGs should be used from which there are 29. In all the available demands and supplies of empty wagons, 12% of the locations requires wagons, 11% has empty wagons available, and around 4% of the locations has demand as well as supply. The parking spaces are for an EWPP 50.000 metre, for a base location 2500m, and for a service location 1000m. The length of the WTRGs lies between 11 and 26 metres. Other given parameters are the costs which are €1 per wagon per kilometre, €1000 per reclassification, and €500 per new locomotive used. This might seem counter intuitive, but it can be assumed that there are enough locomotives. Hiring a crew for a locomotive is much cheaper than for a reclassification yard, since shunting requires more people. Reclassification is also more time consuming than driving from one station to the next one. All the costs are equal for all datasets. The train capacity is set to 40 wagons, which is also used for all arcs in the network. ϵ is set to €5.

5.1.1 Data Case generation

To analyse the performance of the exact IP formulation and the different heuristics, data cases smaller than the real size case are needed. These cases are generated using the following guidelines:

- The EWPPs are all connected to each other
- The EWPPs all have the same number of base locations
- Each base location has the same number of service locations
- All distances between the EWPPs are equal
- All other distances are generated
- A pseudo random generator is used to make sure a dataset can easily be reproduced. For each data case a seed number is required, which can be used to create different datasets with the same parameters
- There are at least two days of which the first day only has supply and the last day only has demand.

5.2 Integer programming approach

Since the IP is used to solve sub problems of the heuristics, it is interesting to know the influence of the parameters on the runtime of the IP. A small network is used to research the influence on the computation time of the number of WTRGs, the number of days, the number of locations with demand, and the number of locations with supply. Next to the runtime, total costs and the number of moves are compared. A move means that a number of wagons of the same WTRG are transported from location A to location B. Travelling along multiple locations requires also multiple moves.

The standard network uses 3 EWPPs, 3 base locations per EWPP, and 3 service locations per base, bringing the total number of locations on 39 locations.

5.2.1 Effect of the number of days

The increase of the number of days, implies also an increase in the number of locations with demand and supply. If a specific location requires wagons every other day, the increase of the number of days will increase the number of demand locations in the network. A location which requires wagons as well today as tomorrow, is counted as two demand locations. In Table 1, it can be seen that the number of days can be increased to 4, while keeping the run time acceptable. The difference between 3 or 4 days, is not that different when one looks at the number of locations with supply/demand. This results in just a little bit more computation time. When the number of days becomes 5, again more locations with demand/supply are added to the network, which leads to another increase in computation time. Six days will lead to that many options, that the runtime exceeds one hour. Since this proves that the problem becomes to complex to solve within reasonable time, runs exceeding one hour were terminated. One can conclude that increasing the number of days leads to more complex problems because the number of locations with demand and/or supply will increase as well.

Nr. of days	Nr. of WTRG	Nr. of locations with demand	Nr. of locations with supply	Build time	Run time	Total costs(€)	Nr. of moves
3	3	12	11	0:00:00	0:00:48	157,406	16
4	3	13	12	0:00:01	0:01:09	234,579	22
5	3	21	22	0:00:01	0:19:17	438,724	37
6	3	25	26	0:00:01	>1:00:00	-	-

Table 1: Influence of number of days

5.2.2 Effect of the number of wagon types

Adding more WTRGs to a network only makes the problem more complex, if the number of locations with demand/supply increases. If the number of WTRGs increases, but the number of locations with demand/supply remains equal, the problem becomes easier. This can be seen in Table 2, when the number of WTRGs is increased from 3 to 4. This can be explained by comparing two scenarios. In scenario 1 there are three locations with ten wagons of type A and three locations which all need five wagons of type A. In scenario 2, there are also three locations, but only one with ten wagons of type A. The others are one with ten wagons of type B and one with ten wagons of type C. On the demand side, for each wagon type, there is only one location which requires five wagons of that type. Clearly, the second scenario is easier to solve, because there is only one possible solution instead of multiple ones. It can be concluded that increasing the number of WTRGs does not need to lead to more complex problems in theory. In practice however, more wagon types will attract new and more customers which will result in a larger and more complex problem.

Nr. of days	Nr. of WTRG	Nr. of locations with demand	Nr. of locations with supply	Build time	Run time	Total costs(€)	Nr. of moves
3	1	8	6	0:00:01	0:00:04	138,370	15
3	2	9	8	0:00:00	0:00:06	114,203	13
3	3	12	11	0:00:00	0:00:48	157,406	16
3	4	12	10	0:00:00	0:00:18	119,211	15
3	5	7	11	0:00:00	0:00:09	120,906	12
3	6	12	17	0:00:01	0:01:30	229,085	27
3	7	16	17	0:00:01	0:01:24	222,663	38
3	8	18	20	0:00:01	0:01:44	322,632	50

Table 2: Influence of the number of WTRGs

5.2.3 Effect of the number of supply locations

If more locations with supply are added, there will be more options to choose from. On the other hand, the probability that a few supply locations are much closer than the rest increases. When this occurs, the decision becomes easier, because it is cheaper to supply a demand location from a location in the neighbourhood. This effect has impact on the runtime. In Table 3 the run times are quite volatile. When a new dataset is generated, demand and supply locations are randomly distributed over the network. The different distances and possibilities for each WTRG determine whether a problem is hard or relatively easy to solve. This results in different solving times for quite similar problems. The conclusion that can be drawn from these results, is that the number of supply locations, does not lead to longer computation times, as long as the other parameters remain the same.

Nr. of days	Nr. of WTRG	Nr. of locations with demand	Nr. of locations with supply	Build time	Run time	Total costs(€)	Nr. of moves
3	3	11	12	0:00:00	0:02:28	207,283	21
3	3	11	13	0:00:00	0:00:14	191,503	20
3	3	11	16	0:00:01	0:06:31	255,569	36
3	3	11	17	0:00:01	0:08:23	295,828	32
3	3	12	18	0:00:00	0:02:59	292,749	28
3	3	12	21	0:00:01	0:03:56	276,162	27
3	3	12	23	0:00:00	0:06:09	274,777	28
3	3	12	24	0:00:00	0:03:06	191,376	17
3	3	12	25	0:00:00	0:02:30	284,713	22

Table 3: Influence of the number of locations with supply

5.2.4 Effect of the number of demand locations

All extra demand locations need to be met as well, unlike the supply locations, where extra options are not necessarily needed to be used. So by adding demand locations, the problem size does increase. This should lead to more complex problems and longer computation times. This conclusion is supported by the results in Table 4. When the number of locations with supply and the locations with demand both increase, the problem becomes more complex. The results can be found in Table 5.

Nr. of days	Nr. of WTRG	Nr. of locations with demand	Nr. of locations with supply	build time	run time	total costs(€)	Nr. of moves
3	3	14	11	0:00:00	0:00:36	233,147	34
3	3	15	11	0:00:00	0:00:23	241,813	35
3	3	16	11	0:00:00	0:00:35	247,251	37
3	3	17	11	0:00:00	0:00:41	255,504	38
3	3	18	11	0:00:00	0:01:04	252,768	41
3	3	19	11	0:00:00	0:02:08	256,531	44
3	3	20	11	0:00:00	0:04:27	259,175	45

Table 4: Influence of the number of locations with demand

Nr. of days	Nr. of WTRG	Nr. of locations with demand	Nr. of locations with supply	build time	run time	total costs(€)	Nr. of moves
3	3	13	12	0:00:00	0:00:15	164,857	29
3	3	12	14	0:00:00	0:01:05	257,543	37
3	3	15	16	0:00:01	0:03:14	354,565	41
3	3	16	18	0:00:01	0:06:11	409,176	46
3	3	17	17	0:00:00	0:23:13	415,843	49
3	3	23	20	0:00:01	1:01:42	490,794	56

Table 5: Influence of locations with demand and locations with supply

5.2.5 Effect of reclassification costs

In the Section 2.2.3 an approach to estimate the number of reclassifications is proposed. Given the data from a railway company, reclassification costs are € 1000 and the setup costs are € 500. To verify if the use of reclassification costs in the decision process is needed, the effect of the costs is researched. To do so, the reclassification costs are lowered in steps and the effect on the number of moves in analysed. It is not usefull to compare the number of reclassifications. Due to the cost structure, moving with a reclassification is always ϵ cheaper then moving to a final destination. The number of reclassification moves will therefore always be correlated with the number of moves.

It can be expected, that the total number of moves decreases when the reclassification costs decrease. Take for example a location which requires WTRG A and B on the same day. When WTRG A needs to be moved from the other side of the network and reclassification costs are high, it is cheaper to add WTRG B along the route. Even if this is also at the other side of the network. Since the reclassification costs already occur because of WTRG A, they are 'free' for WTRG B. Although WTRG B might also be available at a location closer to the destination, the reclassification costs makes it more expensive to use the closer wagons. The final and cheapest solution will be the one where both WTRGs are delivered from the other end of the network, which results in more moves than the solution where WTRG B is delivered from a closer location.

In Table 6 can be seen that the optimal solution changes when the ratio between the reclassification costs and the setup costs change. However, the total number of moves does not change. To make sure that the costs do have such an effect, a bigger dataset is needed. We use a medium dataset, of which the exact size will be further explained in Table 8 in Section 5.3. Unfortunately, the IP cannot solve the medium dataset within reasonable time, so heuristic 2 is used. As will be explained later, heuristic 2 solves the problem with the medium dataset the fastest. The results are displayed in Table 7 from which can be concluded that indeed the reclassification costs do affect the outcome of the solution and should therefore be used in the model.

Reclassification costs	Moves day 1	Moves day 2	Total moves	Total reclassification moves
€1000	0	12	12	8
€750	2	10	12	8
€600	2	10	12	8
€500	6	6	12	8
€400	6	6	12	8
€0	6	6	12	8

Table 6: Effect reclassification costs on small dataset with IP

Reclassification costs	Moves day 1	Moves day 2	Total moves
€1000	161	136	297
€750	160	137	297
€600	155	126	281
€500	155	103	258
€250	132	97	229
€0	97	83	180

Table 7: Effect reclassification costs on medium dataset with heuristic 2

5.2.6 Conclusion

Based on the results of this section, it turned out that the parameters do not influence the computation time, as long as the problem size remains equal. As soon as the problem size increases, computation time also increases. Next to that can be concluded that the reclassification cost do have an impact on the solution and should therefore be taken into account.

5.3 Heuristics

The three different heuristics will be compared based on runtime, total costs, and the number of moves they need to solve the problem. To compare the performance, multiple test cases are used. In Table 8 the sizes are given. All heuristics were also compared to the optimal IP solution, if that was available.

size of dataset	Nr. of EWPPs	Nr. of base per EWPP	Nr. of service per base	Nr. of locations with demand	Nr. of days	Nr. of WTRGs
small A	3	3	3	7	3	5
small B	3	3	3	8	3	5
small C	3	3	3	7	3	5
medium A	3	4	15	46	3	10
medium B	3	4	15	40	3	10
medium C	3	4	15	50	3	10
large A	3	4	20	5	141	20
large B	3	4	20	5	117	20
large C	3	4	20	5	117	20
realsize	3	n/a	n/a	402	7	29

Table 8: The sizes of the data cases

There are 4 types of datasets that differ in size: small, medium, large, and realsize. The small, medium, and large datasets were generated with an equal amount of locations per parent location. This network remains the same for the three versions (A, B, and C), where supply and demand is different. Since the number of locations with demand has the most influence on the complexness of the problem, this informations was also given in Table 8. Large B and C do have the same amount of demand locations, but they were differently distributed over the network. The realsize dataset was not symmetric generated like the other datasets. Its layout was based on customer data and has 23 base locations divided over 3 EWPPS and 199 service locations. This could be approximated with 7 base locations per EWPP and 11 service locations per base. In practice however, the number of service locations differ per base location. The minimum is one service location per base and the

maximum 20. The results of the optimal solution and the three heuristics can be found in Table 9.
For the small data cases the results were similar. The IP formulation and heuristic 1 were both much slower than heuristic 2 and 3. The big difference between the IP and heuristic 1 is the number of moves. The optimal solution over all three days, had the least moves. Heuristic 1 splits the problem per day, which leads to almost the double number of moves. This was different for heuristic 2 and 3. Their results were the same as the result of heuristic 1, but their computation time was much lower. But that was just for the small datasets.

On the Medium data case, the exact formulation caused memory problems on the laptop. Both methods were also tested on a server of Ab Obo, but even 25GB of RAM was not enough. In order to be able to still compare the solutions, the solver was cut after 50 minutes to prevent system failure. The fact that some runtimes were higher then 50 minutes for the IP solutions can be addressed to the buildtime of the model. For heuristics the 50 minutes could be exceeded because they used the solver multiple times.

For the realsize dataset, the solver was unable to find any solution within these minutes, not even for Heuristic 1, where the network was split per day. Therefore only heuristic 2 and 3 are suitable for practical use. From the medium and large dataset can be learned that solutions found by the heuristics are around 30% higher than the best found costs. In most datasets, heuristic 2 is takes more time, but finds a better solution. However, when solving the medium datasets, heuristic 2 is quicker. This can be addressed to the form of the dataset. Solving a few problems of a certain size can in some cases be quicker than solving much more smaller problems.

For practical purposes, heuristic 2 and 3 are the only options. Realsize problems cannot be solved with the IP or with heuristic 1. Since heuristic 2 finds the same solution or a better one compared to heuristic 3 and the sum of both heuristics is still a very reasonable runtime, it is recommended to use heuristic 2.

Dataset	Heuristic	Runtime	Optimality gap	Nr. of moves	Total costs(€)	% from IP solution
Small A	1	0:00:06	-	20	126,380	4.5%
Small A	2	0:00:01	-	21	127,380	5.4%
Small A	3	0:00:01	-	23	127,906	5.8%
Small A	IP	0:00:08	0.0%	12	120,906	0.0%
Small B	1	0:00:17	-	35	130,255	6.6%
Small B	2	0:00:01	-	29	131,647	7.8%
Small B	3	0:00:01	-	39	140,525	15.1%
Small B	IP	0:00:07	0.0%	21	122,139	0.0%
Small C	1	0:00:07	-	20	126,772	4.4%
Small C	2	0:00:01	-	21	127,772	5.2%
Small C	3	0:00:01	-	23	128,406	5.8%
Small C	IP	0:00:07	0.0%	12	121,406	0.0%
Medium A	1	1:56:28	-	187	1,638,615	2.8%
Medium A	2	0:02:33	-	324	1,919,603	20.4%
Medium A	3	0:06:11	-	412	2,124,521	33.3%
Medium A	IP	0:55:16	1.5%	124	1,593,697	0.0%
Medium B	1	1:42:55	-	200	1,648,630	0.6%
Medium B	2	0:02:11	-	388	1,919,930	17.2%
Medium B	3	0:00:23	-	479	2,038,507	24.4%
Medium B	IP	0:56:42	1.8%	167	1,638,630	0.0%
Medium C	1	1:48:35	-	230	2,073,280	1.1%
Medium C	2	0:01:58	-	438	2,345,080	14.4%
Medium C	3	1:40:29	-	580	2,787,373	36.0%
Medium C	IP	1:02:21	1.4%	195	2,050,280	0.0%
Large A	1	4:00:11	-	695	5,473,880	2.9%
Large A	2	0:29:29	-	1298	6,709,666	26.2%
Large A	3	0:02:58	-	1441	6,930,849	30.4%
Large A	IP	1:06:38	1.2%	473	5,317,035	0.0%
Large B	1	3:56:27	-	582	5,020,390	2.1%
Large B	2	0:38:03	-	1127	6,322,369	28.6%
Large B	3	0:02:29	-	1327	6,601,223	34.2%
Large B	IP	1:06:17	1.2%	440	4,917,824	0.0%
Large C	1	3:48:10	-	592	4,964,181	2.0%
Large C	2	0:19:02	-	1165	6,261,195	28.7%
Large C	3	0:02:32	-	1320	6,498,023	33.6%
Large C	IP	1:05:12	1.4%	420	4,865,020	0.0%
Realsize	1	-	-	-	-	-
Realsize	2	1:07:13	-	3475	21,590,848	-
Realsize	3	0:04:02	-	4192	22,458,708	-
Realsize	IP	-	-	-	-	-

Table 9: Results IP and heuristics

6 Conclusion

The main focus of this thesis was to find a method which solves the empty wagon repositioning problem in reasonable time, while minimizing the total costs, given that the network is a hub-and-spoke network.

The first challenge was to take the reclassification costs into account. In literature such costs are neglected or estimated. In this thesis the proposed approach is also an estimation. This estimation is most of the cases exactly correct. The extra information about reclassification leads to better solutions than without reclassification costs, because more moves can be combined.

To solve problem instances exactly, an integer programming formulation was introduced. The IP can only solve small problem instances. This makes the IP not directly usable to solve practical realsize problems. However, the impact of different parameters on the runtime was determined with the IP. Increasing the planning horizon automatically increases the number of demand and supply locations. This effect makes the problem more complex as the number of days increases.

The number of wagon types will not necessarily lead to more complex problems. This is similar with the number of supply locations, as long as all the other parameters remain the same. Increasing the number of demand locations will lead to more complex problems. When the number of locations with demand and the number of locations with supply both increase, the problem becomes quickly more complex.

To solve the larger complex problem three heuristics are tested. All three split the problem into smaller regions. The regions of heuristic 1 are still too big to solve. Heuristic 2 and 3 however, are able to solve a realsize problem within reasonable time. For the tested datasets, heuristic 2 found better solutions. Since the computation time of heuristic 2 is still reasonable, this solution method is recommended.

7 Discussion and recommendations

In this section the strengths and weaknesses of this research are discussed as well as the recommendations for practical implementation and further research.

In the literature different heuristics are applied on similar problems. Other papers use the actual rail networks, instead of a hub-and-spoke network. In this thesis only heuristics based on the divide-and-conquer technique were used and other known heuristics were neglected. These other types might be interesting for further research.

In this thesis the reclassification costs were implemented in the IP formulation. Some papers also included those costs, but their estimation approaches were not as sophisticated as ours. Other articles stated that neglecting reclassification costs does not lead to other results. Although implementing makes the decision problem more complex, it was proven in this thesis that those costs could not be neglected in our cases. Therefore the implementation leads to more realistic results.

The divide-and-conquer heuristics made an excellent use of the hub-and-spoke network. The network layout lent itself perfectly for the creation of sub problems. Results showed that splitting in regions is an appropriate way of tackling the problem, but splitting on days will have a negative effect on the solution. For further research and implementation, it is very interesting to see what the results will be if the region approach is applied while also multiple days are considered.

Because the available data was limited, test cases were generated. The generation has been done with great care and as realistically as possible, but still an enormous amount of possible scenarios are not generated and tested. It is recommended for further research to extend the current research with more test cases in order to make sure more scenarios can be tested.

For practical implementation there are a few recommendations as well. First of all, the grouping of wagon types might be unnecessary. An increase in wagon types does not lead to more complex problems as long as no other parameters increase. Secondly, it is optimal for a solver or heuristic to leave empty wagons at service locations if they are not needed elsewhere. When new empty wagons will become available at the same location, available parking place can become an issue. For an optimizer, it is cheaper and therefore better to send the wagons to the closest available space. In the case of a service location, this will be the base location. A planner probably wants the empty wagons back at the Empty Wagon Parking Place. Such imbalance needs to be detected, or planned manually.

To prevent infeasibility and imbalances, it is recommended to check for all demand to determine whether it is possible to solve the problem at all. Next to that, it can be an idea to neglect the inventories at the EWPPs at first, then check the infeasibility and only add the needed wagons to the problem. With this approach, one prevents that the solver supplies all demands from EWPPs and creates an imbalanced flow in the network. If wagons are repositioned via an EWPP which has similar wagons available, a planner can still split the route into two, one from the origin to the EWPP and one from the EWPP to the destination. This has the advantage that both trains can leave at the same time.

When the heuristics are used to create a model week, which is repeated several times, it is recommended to start one day before the start of the week. This day will be different next week, because a new model week implies other demand and/or supply than the week(s) before. Since the planning horizon with the heuristics of these thesis is at most one day, one day is enough to initialize the model week. After the initial-day, the found moves can be repeated each week, until the demand changes again. Implementing these recommendations will enable one to effectively reposition the empty wagons.

References

Bektas, Crainic, and Morency. Improving the performance of rail yards through dynamic reassignments of empty cars. *Transportation Research part C*, 2008.

Brouer, Pisinger, and Spoorendonk. Liner shipping cargo allocation with repositioning of empty containers. *INFOR*, 49(2):109–124, May 2011.

Cavanagh. The pooling of freight cars. *Sage*, 1907.

Chao and Yu. Repositioning empty containers in east and north china ports. *Maritime Economics & Logistics*, 14(4):435–454, 2012.

Choong, Cole, and Kutanoglu. Empty container management for intermodal transportation networks. *Transportation Research E*, 38:423–438, 2002.

Crainic. A survey of optimization model for lang-haul freight transportation. *Transportation Science*, 2002.

Dong and Song. Container fleet sizing and empty repositioning in liner shipping systems. *Transportation Research E*, 45:860–877, 2009.

Dong, Song, and Xu. Assessment of empty container repositioning policies in maritime transport. *The International Journal of Logistics Management*, 24(1): 49–72, 2013.

Erera, Morales, and Savelsbergh. Robust optimization for empty repositioning problems. *Operations Research*, 57(2):468–483, 2009.

Evdokia, Adamidou, and KornHauser. A game theoretic/network equilibrium solution approach for the railroad freight car management problem. *Transportation Research B*, 27B(3):237–252, 1993.

Florez. *Empty-Container Repositioning and Leasing: An Optimization Model*. PhD thesis, PolyTechnic Institute of New York, 1986.

Di Francesco, Crainic, and Zuddas. The effect of multi-scenario policies on empty container repositioning. *Transportation Research E*, 45:758–770, 2009.

Godfrey and Powell. An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. *Transportation Science*, 35(1): 21–39, 2002.

Holmberg, Joborn, and Lundgren. Improved empty freight car distribution. *Transportation Science*, 1998.

Holmberg, Joborn, and Melin. Lagrangian based heuristics for the multicommodity network flow problem with fixed costs on paths. *Operational Research*, 188: 101–108, 2007.

Ireland, Case, Fallis, Van Dyke, Kuehn, and Meketon. The canadian pacific railway transforms operations by using mmodel to develop its operating plans. *Interfaces*, 34(1):5–14, 2004.

Joborn. *Optimization of empty freight car distribution in scheduled railways*. PhD thesis, Linkoeping Univ. (Sweden). Dept. of Mathematics, 2001.

Joborn, Crainic, Gendreau, Holmberg, and Lundgren. Economies of scale in empty freight car distribution in scheduled railways. *Transportation Science*, 38(2): 121–134, 2004.

Yin Long, Loo hay lee, and Ek Peng Chew. The sample average approximation method for empty container repositioning with uncertainties. *European Journal of Operations Research*, 222:65–75, 2012.

Sergey Maltsev. Globaltrans annual report 2013. Technical report, Globaltrans, 2014. URL `http://www.globaltrans.com/fileadmin/content/2014/Annual_report_2013/Globaltrans_Annual_Report_2013_execution_copy.pdf`.

Martello and Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Science 44*, 44(3):388–399, 1998.

Meng and Wang. Liner shipping service network design with empty container repositioning. *Transportation Research Part E*, 47:695–708, 2011.

Narisetty, Richard, Ramcharan, and Minks an Fuller. An optimization model for empty freight car assignment at union pacific railroad. *Interfaces*, 38(2):89–102, 2008.

Olivo, Zuddas, Di Francesco, and Manca. An operational model for empty container management. *Maritime Economics & Logistics*, 7:199–222, 2005.

Petersen. Rail yard modeling: part i. prediction of put-through time. *Transportation Science*, 11(1):37–49, 1977.

Rodrigue. The repositioning of empty containers. The geography of Transport systems, 2013. URL `https://people.hofstra.edu/geotrans/eng/ch5en/appl5en/ch5a3en.html`.

Sherali and Tuncbilek. Static and dynamic time-space strategic mmodel and algorithms for multilevel rail-car fleet management. *Management Science*, 43(2): 235–250, 1997.

Shintani, Imai, Nishimura, and Papadimitriou. The container shipping network design problem with empty container repositioning. *Transportation Research E*, 43:39–59, 2005.

Song and Carter. Empty container repositioning in liner shipping. *Maritime Policy and Management*, 36(4):291–307, 2009.

Song and Dong. Cargo routing and empty container repositioning in multiple shipping service routes. *Transportation Research Part B*, 46:1556–1575, 2012.

Zhang and Facanha. Strategic planning of empty container repositioning in the transpacific market: a case study. *International Journal of Logistics Research and Applications: A Leading Journal of Supply Chain Management*, 2014.