

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization: Transport Engineering and Logistics

Report number: 2017.TEL.8200

Title: **Optimizing train parking and
shunting at NS service yards**

Author: E. Beerthuisen

Title (in Dutch) Optimalisering van het parkeren en behandelen van treinen op NS service locaties

Assignment: Masters thesis

Confidential: no

Initiator (university): ir. M.B. Duinkerken

Initiator (company): ir. B. Huisman (NS Techniek, Utrecht)

Supervisor: prof. dr. R.R. Negenborn

Date: Februari 14, 2018

This report consists of 86 pages and 2 appendices. It may only be reproduced literally and as a whole. For commercial purposes only with written authorization of Delft University of Technology. Requests for consult are only taken into consideration under the condition that the applicant denies all legal rights on liabilities concerning the contents of the advice.

Preface

With two generations in the rail business ahead of me, trains have always been a present factor in my life. I was only a few months old when I experienced my first international railway journey, and many would follow in the years to come. The regular visits to the national railway museum with my dad were a joyful experience, and it must be there that I aroused some of my interest for technics and science, because trains and steam locomotives in particular are some unique pieces of engineering. Eventually this interest led me to a bachelor grade in Mechanical Engineering, and a master in Transport Engineering.

In the Netherlands there is only obviously one major player in public rail transport, so in quest for an interesting graduation project, I made sure that NS crossed my path. I am really grateful they offered me a great logistical problem, which I really enjoyed working on. After playing my railroad rush hour game in spare time, now it was time to dig into the real deal; shuffling with real size (though imaginary) trains.

With help of my supervisor at NS we created an assignment that was both useful for NS, and created enough scientific possibilities for the TU Delft to approve. It took little over a year to take the research from a proposal to a fully substantiated solution and along the way I have been extensively assisted by a lot of people.

First I would like to thank Mark Duinkerken for the extensive reading and suggestions on my writing, and professor Rudy Negenborn for his supervising during my graduation project. This thesis however would not be the same without the support of NS, my supervisor Bob Huisman who was always available at the right moment, and the many colleagues at NS that helped me to master the problem. I am grateful for all the insightful discussions we have had to give me better understanding of the shunting problem and many other issues. Besides, although a lot of work is being done, days at NS were never dull.

Last but not least I would like to thank my brother for his review on my work, Myriël for adding some social pressure at difficult stages and my parents for their moral and financial support during the project.

I hope you will enjoy reading my thesis.

Erik Beerthuizen

Februari 2018

Summary

In the Netherlands train services are offered 24 hours a day, but some of the rolling stock is only used during peak hours. As a consequence a significant amount of trains is stabled during a major part of the day. For this purpose, shunting yards are available at key locations, where trains can be set aside. Some of these yards are also used to perform service activities on trains such as cleaning and safety checks, which means that trains have to be moved to a specific location during their stay in the yard.

Management of temporary superfluous resources and maintenance is a recurring topic in many fields; in the rail world it could be referred to as the Train Unit Shunting Problem (TUSP), or extended TUSP if maintenance activities are included. At NS these activities are currently scheduled by hand, which is difficult and time consuming, since multiple types of trains are not compatible with each other. This master thesis focusses on improving the planning of parking and service activities, resulting in the research question:

To what degree can the occupation rate of a rail yard be maximized by using heuristics, while maintaining or improving the ability to execute the desired service activities in the yard.

To answer this question an assessment is made on existing methods to solve the TUSP, and several container stacking and reshuffling methods. Based on this assessment we propose shunting policies for the extended TUSP that are derived from container stacking.

All proposed solution methods can generally be divided in two types and both use characteristics of trains to allocate trains to tracks. The first solution type strives to compose tracks each containing just one type of train; this type is referred to as Type Based Strategy (TBS). The second solution type allocates trains to tracks based on departure time, to create a chronological departure sequence of trains at each track; this type is referred to as In Residence Time Strategy (IRTS).

A discrete process simulation is constructed for two real life service yards 'Kleine Binckhorst' and 'Cartesiusweg' to assess the shunting policies. The policies are not capable of optimizing towards a perfect solution, but most proposed policies are able to solve instances of a reasonable size in a few seconds. The methods incorporate logistical knowledge obtained in practice to optimise flow through the yard, resulting in a policy that is easy to implement in reality. The quality of the policies can be scored based on the following criteria:

- Feasible solutions : Number of instances with a solution.
- Calculation Time : Time needed to calculate solutions.

Furthermore the quality of the individual solutions can be graded by the number of occurrences of:

- Violation of preferred order : number of trains that were not parked in the preferred order.
- Combine at departure : number of trains units of which the train is not complete while parked.
- Moves : number of train moves needed for solution.
- Splits : number of trains that were split

TBS proves to be inefficient with the use of track length, since trains of the same length rarely add up to the full length of a track. As a result TBS is able to solve instances with up to 20 train units, in contrast to the 26 train units achieved by IRTS. Of the various variations on IRTS, the basic version proves to be best with 76% success out of theoretical possible instances. All attempts to improve the solution, lead to deterioration of the overall performance of the method.

The best performing method, IRTS, is compared to various other methods developed under supervision of NS. IRTS is able to find a solution in the same range as Tabu Search (Row Generation), developed by F. Wolfhagen [31]. TS-RG is able to successfully solve 6 instances more than IRTS, mainly in datasets containing less trainsets. Although the TS-RG is much more sophisticated and takes a significant amount of time to compute, up to 1800 seconds per solution compared to 6 seconds with IRTS, it is not performing significantly better.

OPG, one of the earlier developed tools by NS, is able to successfully solve 7 instances less than IRTS. OPG has a higher uncertainty of solving successfully instances with less trainsets, but it is able to solve some of the instances with more trainsets than IRTS.

IRTS performs not as well as any Simulated Annealing method, developed by R. van den Broek [6]. However when capacity at tracks is returned at departure of a train, IRTS nears the result of SA-with-Service-Tasks.

IRTS is also applied to a bigger yard with other layout and dead-end tracks. The best method achieves a 90% success rate out of the theoretical possible instances.

A major upside of the solution method, is that it illustrates a scenario that is easy to implement and execute in real world. Overall can be concluded that IRTS is a fairly simple policy that performs remarkably well when compared to various mathematical optimisation methods. These results suggest that IRTS offers a pragmatic, swift and flexible opportunity to improve train yard logistics. Further research is needed to validate the feasibility and effectiveness of IRTS in circumstances beyond those in this assessment.

Summary in Dutch (Samenvatting)

In Nederland gaat de treindienst 24 uur per dag door. Een deel van het materieel wordt echter alleen in de spitsuren gebruikt, waardoor een significant deel van de treinen voor een groot deel van de dag geparkeerd moeten worden. Voor dit doel zijn speciale opstelterreinen beschikbaar op tactische plaatsen waar treinen weggezet kunnen worden. Een aantal van deze opstelterreinen wordt ook gebruikt voor het uitvoeren van serviceactiviteiten, zoals schoonmaken en veiligheidscontroles. Hiervoor moeten treinen gedurende hun verblijf verplaatst worden naar specifieke locaties op het terrein.

Het beheren van tijdelijk overvloedige middelen en onderhoud is een terugkerend onderwerp in veel vakgebieden. In de spoorwereld kan ernaar gerefereerd worden als het treinstel rangeer probleem, of het uitgebreide treinstel rangeer probleem wanneer onderhoudstaken meegenomen worden. Bij NS worden deze activiteiten momenteel met de hand gepland, wat ingewikkeld en tijdrovend is aangezien er een hoop verschillende treintypes zijn die niet compatibel zijn met elkaar. Deze master thesis focust op verbetering van de planning van parkeren en service activiteiten, wat resulteert in de volgende onderzoeksvraag:

In welke mate kan de bezettingsgraad van een opstelterrein worden gemaximaliseerd door gebruik van heuristieken, terwijl de mogelijkheid om servicetaken uit te voeren gelijk blijft of verbetert.

Om deze vraag te beantwoorden wordt gekeken naar bestaande methodes om het rangeerprobleem op te lossen, en naar verschillende container stapel en -herschikkingsproblemen. Op basis van deze analyse introduceren we een nieuw beleid om het uitgebreide rangeerprobleem op te lossen.

Alle voorgestelde methodes kunnen grofweg worden onderverdeeld in twee categorieën; beide gebruiken eigenschappen van treinen om ze toe te wijzen aan sporen. Het eerste oplossingstype probeert sporen samen te stellen met één type trein per spoor; dit type wordt aangeduid als type gebaseerde strategie (TBS). Het tweede oplossingstype wijst treinen toe aan sporen op basis van vertrektijd, om zo een chronologische volgorde te creëren. Dit type wordt aangeduid als verblijfstijdstrategie (IRTS).

Een *discrete proces simulation* is opgezet voor twee echte opstelterreinen, de Kleine Binckhorst en Cartesiusweg, om het rangeerbeleid te testen. Beide methodes zijn niet in staat om een resultaat te optimaliseren, maar de meeste varianten zijn in staat om instanties van een redelijk formaat op te lossen in enkele secondes. De methodes maken gebruik van logistieke kennis uit de praktijk om de doorstroming over het opstelterrein te bevorderen, wat resulteert in een beleid dat makkelijk in te voeren is in werkelijkheid. De kwaliteit van de methodes kan beoordeeld worden op basis van de volgende criteria:

- Uitvoerbare oplossingen: Het aantal instanties met een haalbare oplossing.
- Rekentijd: De tijd die nodig was om tot een oplossing te komen

Verder kan de kwaliteit van de individuele oplossingen gewaardeerd worden door het aantal:

- Schending van voorgeschreven volgorde : aantal treinen dat niet geparkeerd wordt in de voorgeschreven volgorde.
- Combineren bij vertrek : aantal trainstellen waarvan de trein niet compleet is tijdens parkeren.
- Verplaatsingen : het aantal treinverplaatsingen benodigd voor een oplossing.
- Splitsingen : Het aantal treinen dat gesplitst is tijdens een oplossing

TBS bewijst inefficiënt om te gaan met de spoorlengte, veroorzaakt doordat treinen van dezelfde lengte samen zelden even lang zijn als een spoor. Als gevolg kan TBS instanties oplossen tot 20 treinen, in tegenstelling tot 26 bij IRTS. Van de variaties op IRTS is de basisversie de best presterende met 76% succes van de theoretisch mogelijke instanties. Pogingen om de individuele oplossingen te verbeteren leiden altijd tot een afname van de prestatie van de methode.

De best presterende methode, IRTS, is vergeleken met verschillende andere methodes die bij NS zijn ontwikkeld. IRTS is in staat om oplossingen te vinden in hetzelfde bereik als TS-RG van F. Wolfhagen. TS-RG kan 6 instanties meer oplossen dan IRTS, hoofdzakelijk in datasets met minder treinstellen. Hoewel TS-RG een stuk geavanceerder is dan IRTS en een stuk meer rekentijd nodig heeft met 1800 seconden per simulatie, ten opzichte van 6 seconde met IRTS, presteert het niet significant beter.

OPG, een van de eerdere methodes van NS, is in staat om 7 instanties meer op te lossen dan IRTS. OPG heeft een hogere onzekerheid bij kleine instanties, maar is in staat om instanties met meer treinstellen op te lossen dan IRTS.

IRTS presteert niet zo goed als SA, ontwikkeld door R. van den Broek. Wanneer echter de capaciteit van sporen wordt vrijgegeven bij vertrek, benadert IRTS het oplossingsbereik van SA-met-servicetaken.

IRTS is ook getest op een groter opstelterrein met een andere lay-out en doodlopende sporen. De beste methode behaalt een slagingspercentage van 90% van de simulaties.

Een groot voordeel van het rangeerbeleid is dat het eenvoudig toe te passen is in werkelijkheid. Globaal kan worden geconcludeerd dat IRTS een eenvoudig beleid is dat opmerkelijk goed presteert in vergelijking met verschillende mathematische oplossingsmethodes. Deze resultaten suggereren dat IRTS een pragmatische, snelle en flexibele oplossing biedt om de logistiek van opstelterreinen te verbeteren.

List of abbreviations

Abbreviation	Description
Bkh.kl	Kleine Binckhorst (Rail yard 1, near The Hague)
Ctw	Cartesiusweg (Rail yard 2, near Utrecht)
IRTS	In-Residence-Time-Strategy
NS	Nederlandse Spoorwegen (Dutch national railway operator)
TBS	Type-Based-Strategy
TUSP	Train Unit Shunting Problem

Content

Preface	i
Summary	ii
Summary in Dutch (Samenvatting)	iv
List of abbreviations	vi
Content	vii
1. Introduction.....	1
1.1 Company introduction	1
1.2 Assignment introduction	1
1.3 Scope.....	3
1.4 Aim	3
1.5 Main question	3
1.6 Research questions	3
1.7 Methodology.....	4
2. Analysis.....	5
2.1 Train	5
2.2 Layout of the yard	7
2.3 Setup of a Shunting Yard.....	8
2.4 Flow through a shunting yard	11
2.5 Number of trains entering the yard	13
2.6 Arrival times	16
2.7 KPI's	17
2.8 Conclusion.....	18
3. Literature	19
3.1 General shunting problems	19
3.2 Passenger train shunting and maintenance problems	20
3.3 Container assignment and relocation problems	24
3.4 Conclusion.....	31

4.	Solution methods	32
4.1	Approach.....	32
4.2	Assumptions	33
4.3	Elaboration of solution methods.....	35
4.4	Service activities	38
4.5	Conclusion.....	38
5.	Application of the solution methods in a model	39
5.1	General setup	39
5.2	Train selection procedure	42
5.3	Track selection procedure.....	44
5.4	Special operations.....	47
5.5	KPI's	48
5.6	Conclusion.....	50
6.	Results	51
6.1	Location Kleine Binckhorst (Bkh)	51
6.2	Location Cartesiusweg (Ctw)	59
6.3	Conclusion.....	64
7.	Conclusion and recommendations	65
7.1	Future research	66
	References.....	67
	Appendix A	70
	Scientific paper	70
	Appendix B	77
	Treatment time of train types.....	77

1. Introduction

1.1 Company introduction

The Dutch rail network is one of the highest occupied rail networks in the world. Every day, over 600 train units execute services on more than 6000km of rail. In order to keep the system moving, the reliability of rolling stock is a big issue. NS Techniek, part of the Dutch national railway operator (NS), is entrusted with the task of planning and executing maintenance of the fleet of NS. Many types of operations are done to keep the fleet in perfect condition; from the overhaul of complete series of train units once in 15 to 20 years, to emptying the trash cans a few times a day.

The department of 'Fleet services', manages the maintenance of rolling stock on a strategical level. In this department the view is in general not at one specific train unit, but at generating methods to improve maintenance that applies to an entire series of train units, or even multiple series.

Within Fleet Services, the department 'Maintenance Development' works at an analytical level on prediction a of maintenance. The department uses numerical methods improve maintenance of the rolling stock of NS. For example, data retrieved from onboard computers, sensors along the track and maintenance reports are evaluated in order to detect typical behaviour prior to total failure of a component.

1.2 Assignment introduction

In the Netherlands train services are offered 24 hours a day. During peak hours however, trains contain more train units than outside peak hours. Some of the rolling stock is only used during peak hours, and as a consequence a significant amount of trains is stabled during a major part of the day. For this purpose, shunting yards are available at key locations, where trains can be set aside. Trains are parked on several adjacent tracks with multiple train units per track, possibly blocking each other's exit. Besides parking, some of these yards are also used to execute service activities, and this is where currently a current major problem arises.

At night time, most trains of NS are parked in shunting yards. At the end of their service, trains arrive in the yard in specific sequence. The order of trains entering is theoretically known in advance. During the night a lot specific trains will have to be moved to a certain location in the yard where some service activities are executed. These service activities differ per individual train unit.

In the morning trains have to leave in a specific sequence based on type of train. In most cases trains of the same type are interchangeable; it is only prescribed that a train of a specific type and specific length should start a specific service at a specific time. In some occasions a specific train unit should be assigned to a specific service, but this is rare.

A planner makes a shunting plan for the period between two peak hours based on the expected arrivals and departures. It takes him a few hours to produce an optimised and feasible plan. In reality however, disruptions in the rail network influence the flow of trains in the system. As a result, the order of trains arriving at the yard alters, requiring a new shunting plan for the yard. Sometimes, for instance in case of major disruptions in the network, the order in which trains enter the yard, only becomes clear only minutes before they arrive at the gate. This is too short notice to make any plan at all, causing an unstructured processing of trains in the yard, which might lead to a chaotic start-up of services with high risk of delays, or a lot of shunting during the night, which means there should be more tracks unavailable for storage in order to facilitate the extra movements.

Due to the service activities and resulting shunting movements, the shunting yard cannot be used to full capacity. When trains have to be rearranged at a track, one needs empty space to place a relocated train and free routes towards the service locations. The current utilization rate of a yard is approximately 70 percent of the static capacity. There should be better coordination in the shunting operations of trains, in order to achieve a higher utilization of the yard capacities. At NS various traditional methods are now being developed to optimise the problem. The problem has become more urgent, since NS has ordered new trains which will be delivered in the coming years, but it is not evident if the existing shunting yards are capable of storing and servicing this extra amount of equipment.

Since traditional methods are already under investigation, NS is interested in learning from similar problems in other industries. In complete different type of logistic chains, the quest for optimisation has started some time ago. In maritime shipping for instance, the demand for a better processing of containers has been a topic for research over the last decades. Due to globalisation the volume of containers going through ports keeps increasing and, although change goes slow in a conservative world like maritime shipping, the demand for implementation of new tactic are now emerging.

Modern seagoing vessels of ever increasing size, put the logistics of a terminal under more pressure than ever. Since a ship only earns money while on the move, and costs money while waiting at the quay, the shipping company desires to spent as few time at the quay as possible. Modern ships of 13000 TEU, will easily deliver up to 4000 TUE at a time in one port. Those containers should not only be unloaded, but also have to be stored until they are collected for further transport. At the same time an equivalent number of containers that have been delivered earlier should be retrieved from the storage to load the ship. Storage takes place at vast yards near the quay, where containers are stacked in long rows of several bays, with multiple containers on top of each other. Traditionally the intended loading position of each specific container in a ship is determined in advance, irrespectively their location of the container in the yard. Due to these constraints, the peak demand for in-terminal transport is currently so high, that it formed a bottleneck in the logistic chain. The stacking crane is responsible for retrieving a container from the yard. If the container is not on top of the pile, other containers should be removed first, adding extra tasks to an already overloaded process. In order to reduce the workload of the

stacking cranes, several methods have been developed to reduce the risk of a container not being on top.

The apparent similarities of the two problems aroused the interest of NS to perform research towards applying improved methods of the container stacking problem to the track assignment problem.

1.3 Scope

During this research only the area of the shunting yard is taken into consideration. Trains appear at the connection with the main line, at the entrance of the railyard, and are assigned to tasks along the railyard. The scope will be inside the yard and will include the time-dependant service activities that take place at different locations in the yard. The timeframe that is taken into account is the period from the first train arriving at the yard at the end of service until all trains are processed, or limited by the available time during the day until a new service.

1.4 Aim

The goal of this assignment is to develop a new approach to improve the efficiency of train parking, while keeping the possibility to execute all servicing activities. Efficient in this context refers to a feasible solution, with improved use of capacity, calculated in a time window of a few minutes. The assignment will focus on optimizing the consecution in which trains are processed, but will take service activities such as cleaning into account.

1.5 Main question

To what degree can the occupation rate of a rail yard be maximized by using heuristics, while maintaining or improving the ability to execute the desired service activities in the yard.

1.6 Research questions

1. What tasks take place in a passenger train railyard
2. What methods exist for solving related rail problems, embedded in the Train Unit Shunting Problem, and what is their performance
3. What methods exist for improving similar problems in other industries, and can they be adapted for the rail yard maintenance problem.
4. What alternative methods can be applied to the TUSP.
5. What would be a suitable model to assess the alternative methods for the TUSP.
6. What effect has the alternative method for train parking when applied to a traditional 'carousel' type railyard.
7. What effect has the alternative method for train parking when applied to a modern 'shuffle board' type railyard.

1.7 Methodology

The key to this research is to evaluate the logistic problem in a passenger train shunting yard, compare it to other logistic problems in the rail industry and other industries, and use this comparison to improve the solution of the train unit shunting problem. First the individual elements and processes in a shunting yard are explained in detail in chapter 2. In chapter 3 the scientific efforts to solve relevant problems are reviewed during a literature research. In the second part of the same chapter a translation between the various problems and the rail problem is made. In chapter 4 the solution methods will be introduced and in chapter 5 a discrete process simulation will be created to evaluate the results that can be obtained by the proposed heuristics and strategies. The simulation will be created for two locations, so the effect of different yard layouts can be taken into account. Finally, the results will be presented in chapter 6, in which the results are compared to the physical maximum capacity of the yard and methods that were developed earlier under supervision of NS.

2. Analysis

In this chapter an analysis is performed on the train unit shunting problem. The train unit shunting problem is evaluated by systematically describing the problem. In paragraph 2.1 and 2.2 all the individual aspects are described in detail. Executing this analysis will provide the answer to research question 1.

2.1 Train

The word 'train' is a collective name for a composition of one or more rail wagons or rail cars. In this research we will consider:

1. Rail car or rail wagon; one individual unit
2. Train unit; a composition of one or multiple rail cars in a fixed order, mostly self-propelled
3. Train; a composition of one or multiple train units for executing a passenger service

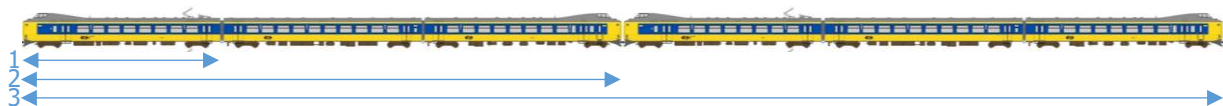
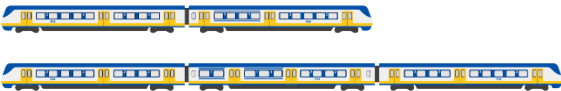


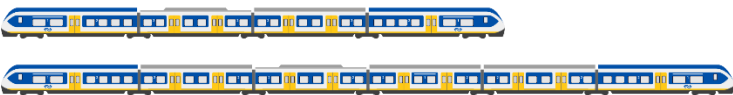




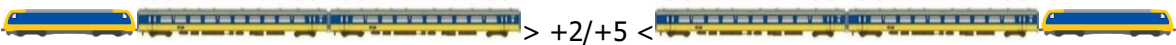


Figure 1-Example of a train containing multiple train units and rail wagons

Trains are assets with a high value. Due to time consuming developing and admittance procedures of new rolling stock, trains are designed and maintained to have a long-life expectancy. NS is no exception. Most trains of NS are planned to be in service for 30 to 40 years, but exceed this in reality. For financial and practical reasons not all trains are bought at once, but smaller series of trains are bought over the decades. As a result of this policy, a rail companies fleet typically consists of many types of trains that are often not compatible with each other. Train units of the same type of various lengths can be coupled to form a longer train. Theoretically a train can consist of up to 5 train units, depending on the type of train. In practice trains are however no more than 4 train units, or up to 12 rail wagons long. The train types that are currently in service at NS are depicted in the overview at the next page.

Local service

Name	Rail cars	First introduction	Comments
SGM	2 or 3	1975	
			
DDM1	4*	1985	+ locomotive
			
DDAR	3*	1992	+ locomotive
			
SLT	4 or 6*	2009	Different wagon length
			
Flirt	3 or 4*	2016	Different wagon length
			

Intercity

Name	Rail cars	First introduction	Comments
ICM	3 or 4	1977	
			
ICR	6 or 9*	1980	+ 2 locomotives
			
VIRM	4 or 6	1993	
			
DDZ	4 or 6	2010* (1992)	Build up from overhauled DDAR wagons
			

For this research not all train types are relevant. Some types only operate in a specific part of the network, and do not visit the yards in scope of this research. Only 3 types are used in this research: SLT, VIRM and DDZ.

2.2 Layout of the yard

A rail yard consists of several tracks with variable length. Each track is partitioned in slots with the length of a standard wagon. Trains occupy a number of slots, corresponding to the number of wagons in the train.

Rail yards exist in many different layouts, but have some characteristics in common. They all consist of several tracks and each track can contain several railcars. Generally, tracks are long enough to facilitate more than one train unit, which means trains can block each other from leaving in a specific direction. It varies if the tracks have dead ends, where trains enter and leave the track at the same side of the track (First In, Last Out), or if the tracks have open ends, where trains enter and leave at the both sides of the track. For logistical reasons, open end tracks could be initially treated as one-way tracks (First In, First Out), but it is obviously possible to enter and leave on the same side or use it in opposite direction.

The railyards in scope of this research are all equipped with at least one cleaning platform, which can be reached from each of the entry tracks. Trains have to be internally cleaned before each morning rush hour, so every train passes the cleaning platform during their stay in the yard.

Some trains in the yard demand special treatment.

- Once a week a train should be externally cleaned, then it should visit an extra location in the yard.
- Some train units should visit a specific maintenance location within a few days, so the train cannot be assigned to any train, but should service a specific route.

In this research two railyards are considered, a small 'carrousel' type yard near The Hague and a large 'shuffleboard' type yard near Utrecht: the 'Kleine Binckhorst' (Bkh) and 'Cartesiusweg' (Ctw). In a carrousel type yard, the trains drive around the yard along different service locations in the yard. In a shuffleboard type yard, the trains are shoved onto a dead-end track.

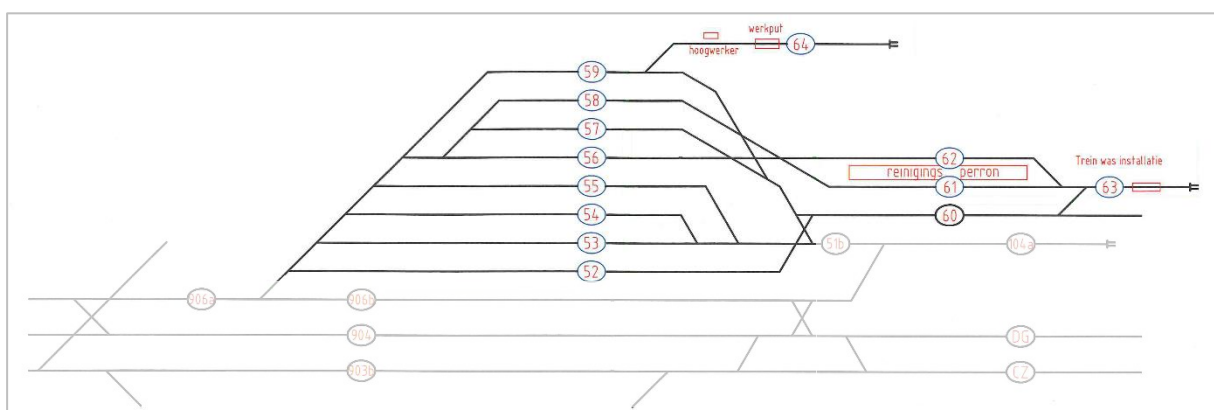


Figure 2-Layout of Kleine Binckhorst (Bkh)

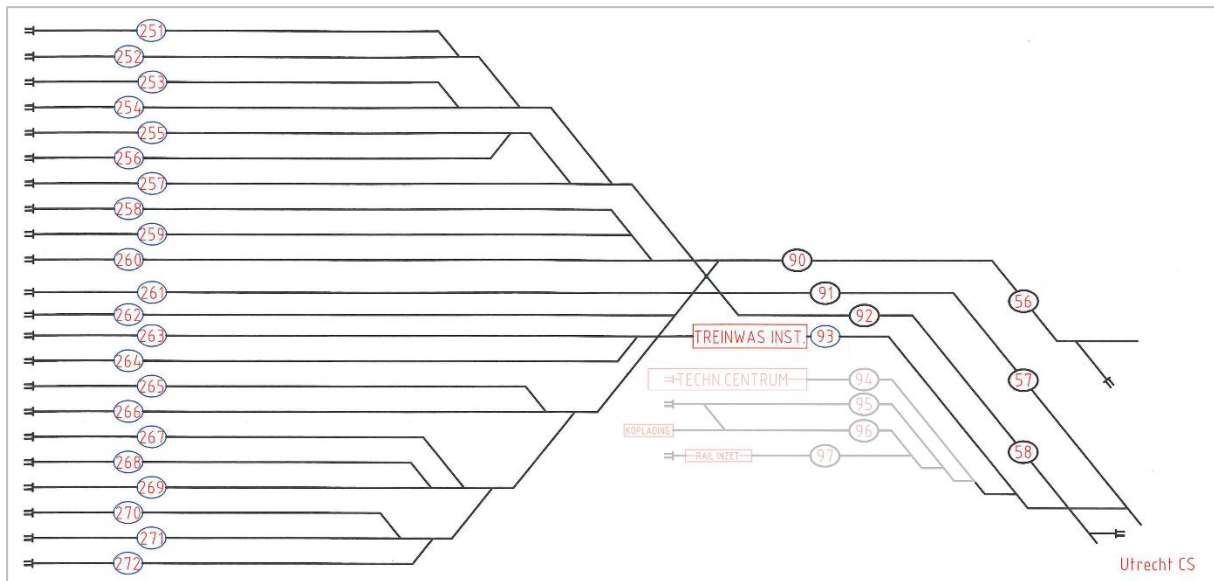


Figure 3-Layout of Cartesiusweg (Ctw)

2.3 Setup of a Shunting Yard

As mentioned in the introduction, a large percentage of trains is only used during peak hours. During the remaining hours they are stored in a railyard, often with multiple trains on the same track. After finishing a passenger service, train units roll out on their final destination. When a free path is available from the station towards a railyard, the train is moved. At entry of the yard, the train is preferably placed at specific tracks that are reserved for incoming trains.

The yard has been introduced previously as a location where trains are stored while not in service, but the idle time period is also used to execute several maintenance tasks. The most important tasks that take place at a yard are:

- Parking
- Reassembling of trains
- Internal cleaning
- External washing
- Technical checks
- Small repairs

Some of these maintenance tasks can only be executed on a specific track in the yard. As a result of this, trains will move through the rail yard during their stay to reach an appropriate location. In order to be able to move trains around the yard, some space has to be reserved. For a start there has to be enough space at the destination to receive a train, and there have to be some connecting tracks between the two locations that are not suitable for parking as well. These factors create a significant reduction in capacity, compared to the static capacity.

In reality, tracks are partitioned in slots. Each slot is the length of a standard passenger rail wagon. The slots are indicated with a marker along the track, so a train's driver knows where he should stop his train in order to optimally use the length of the track without blocking switches. In this research we will use the same partitioning of the yard.

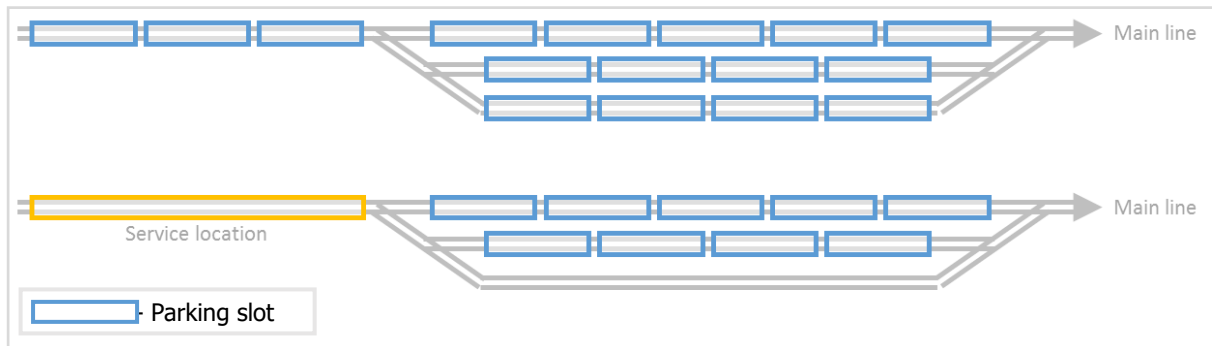


Figure 4-Storage yard versus service yard

We consider a very simple example to illustrate the variability in parking capacity of a yard. Assume a yard with a connection with the mainline at one side, 3 tracks for storage and one tail track as depicted in Figure 4. If the only function of all tracks is storage, and recovery of trains from the yard is no issue, 16 railcars can be stored in the yard. We refer to this as the static capacity.

However, when a service location is situated at the tail track, and this location should be accessible by trains arriving from the main line, the capacity reduces significantly. The service location cannot be used for storage, and one track should be empty to accommodate access to the service location. This empty track could also be used for re-arranging trains on the storage tracks. The dynamic capacity is 9 railcars; a reduction of 44% compared to the static capacity.

Furthermore, it is not evident that the capacity can always be used to the maximum. NS has train units of various lengths, ranging from 2 to 6 railcars. If badly managed, the yard from the example can be filled by two train units of two cars, and one train unit of three cars. Only 7 slots are occupied, but no train can be added without reshuffling of the yard. Both previous examples illustrate that capacity changes under influence of occupation of the yard, and assignment of dedicated tracks. Since the remaining capacity changes under influence of the filling of the yard, we will refer to this phenomenon as 'dynamic capacity'.



Figure 5-Dynamic capacity

Several tasks are executed in the yard, which are already mentioned in the beginning of this paragraph. In the following part each task is elaborated in detail.

Internal cleaning

Over one million people travel by train each day. All these people produce lots of dirtiness inside trains. In order to maintain pleasant surroundings, each train is cleaned once a day. All trains should be clean when entering the morning rush hour.

The number of locations suitable for internal cleaning depends on the layout and setup of the yard. Health and safety regulations prescribe that each door that can be opened by a cleaner, should provide a safe exit. In modern railyards this is solved by adding movable steps along the track that can be aligned with the doors of the trains. If these steps are mounted along all tracks of the yard, the place dependency of this task is eliminated. Most railyards however have a more traditional setup, where a special cleaning platform is situated along one or several tracks. Trains have to be moved towards the specific track to be cleaned, and have to clear the track afterwards to make room for the next train. Since each train should be cleaned every day, a lot of trains pass this location in a limited time window, creating a bottleneck in the process. The internal cleaning takes 24 to 46 minutes per railcar for one person, depending on the type of train. A double decker train for instance takes almost double the time compared to a small sprinter train. Cleaners commonly work in teams, speeding up the process. At the Kleine Binckhorst typically 5 cleaners simultaneously clean one train.

External washing

NS trains added together cover nearly 3 million km a week. During these kilometres a lot of dirt is collected on the outside of a train, so once a week a train needs to visit a washing machine. The washing machine is located on one track in the yard. Trains have to be moved to this track in order to be serviced. In some cases, the washing machine is located along a track that is also needed for shunting, which has two consequences:

- Trains that need to be washed already pass the track with the washing machine, so no extra moves would be necessary.
- Trains that do not need to be washed are blocked while another train is being washed.

The washing process is as follows: a train drives through the washing machine at a pace of 2kph (1 minute per wagon) to wash both sides of the train. The front of the train is washed manually at the same track, which takes 10 minutes for each end of the train. So, for instance, a train of 4 wagons takes $(4*1)+(2*10) = 24$ minutes to be washed.

Safety check

To keep trains safe, the function of some technical aspects have to be checked regularly. Safety checks cannot be executed along cleaning platforms, in the washing machine or at connecting tracks and switches, but it can be performed along most other tracks. The allocation of mechanics should be planned, but in this research it is assumed that each train will automatically stay long enough at a suitable track that it is not a significant factor in the assignment of tracks.

Maintenance

Technical maintenance is executed at various locations. A train regularly visits a maintenance depot at a central location for structural maintenance. Some small technical repairs can be executed within the yard, for example replacing the wearing strip of a pantograph. Some of these small repairs need special equipment situated along specific tracks, such as an aerial platform in case of the example. These tasks are however not scheduled during this research.

Shunting movements

Trains have to be moved along the yard, either to reach a specific location, or to re-order the consecution of trains along a track. Shunting movements use the switches and the tracks in between the origin and destination. Once an area is in use for one shunting movement, no other movements are allowed in the same area or any of the neighbouring tracks.

2.4 Flow through a shunting yard

A typical flow through the yard can be described as follows. The trains wait at their entry track until space is available along the cleaning platform. If enough space is available along the cleaning platform, and a path towards the platform is free, the train moves towards the platform and waits there until it is internally cleaned. When cleaned, a path towards a storage track is reserved, and when available the train moves to its new location, where it waits for a new task. This task can be either the start of a new passenger service, or a service activity somewhere in the yard.

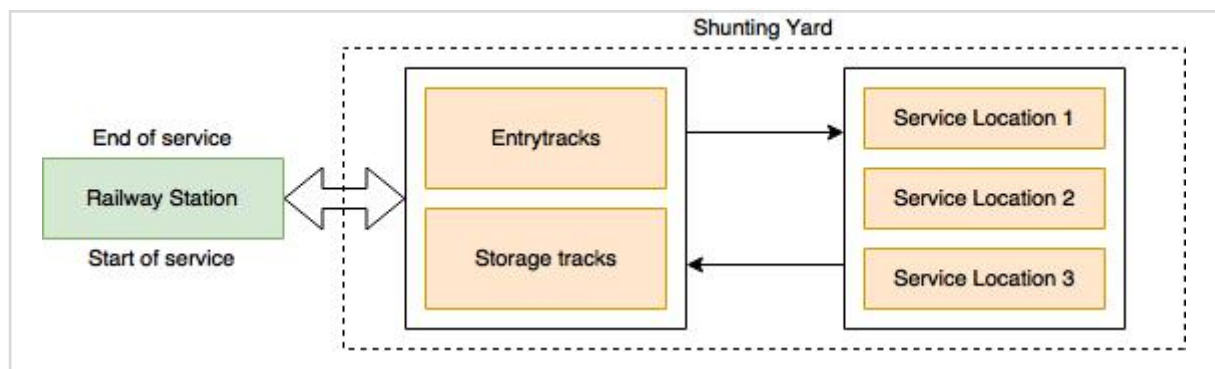


Figure 6-Flow through a shunting yard

External cleaning is done once a week, and is mostly done during a move that is planned anyway, such as driving from the cleaning platform towards the storage tracks. Both cleaning tasks often appear to be a bottleneck in the system, since these services have limited capacity, and a high percentage of trains in the yard should pass these locations in the yard.

Table 1-Characteristics of maintenance activities

Activity	Frequency	Processing time	Location
Internal Cleaning	Each day (before morning rush hour)	Depends on train type (24-46 minutes per rail car*)	Along cleaning platform
External Cleaning	Once in 7 days		In washing machine
Soap	8 out of 9 times	1 minute per rail car + 10 minutes per cab	
Oxalic	1 out of 9 times	4 minutes per rail car + 10 minutes per cab	
Safety check A	Once in 12 days	Depends on train type (8-27 minutes per train unit*)	All tracks (except cleaning platform and washing machine)
Safety check B	Once in 2 days	Depends on train type (38-90 minutes per train unit*)	All tracks (except cleaning platform and washing machine)
Maintenance	Incidental		

*Specific values for each train type can be found in Appendix B

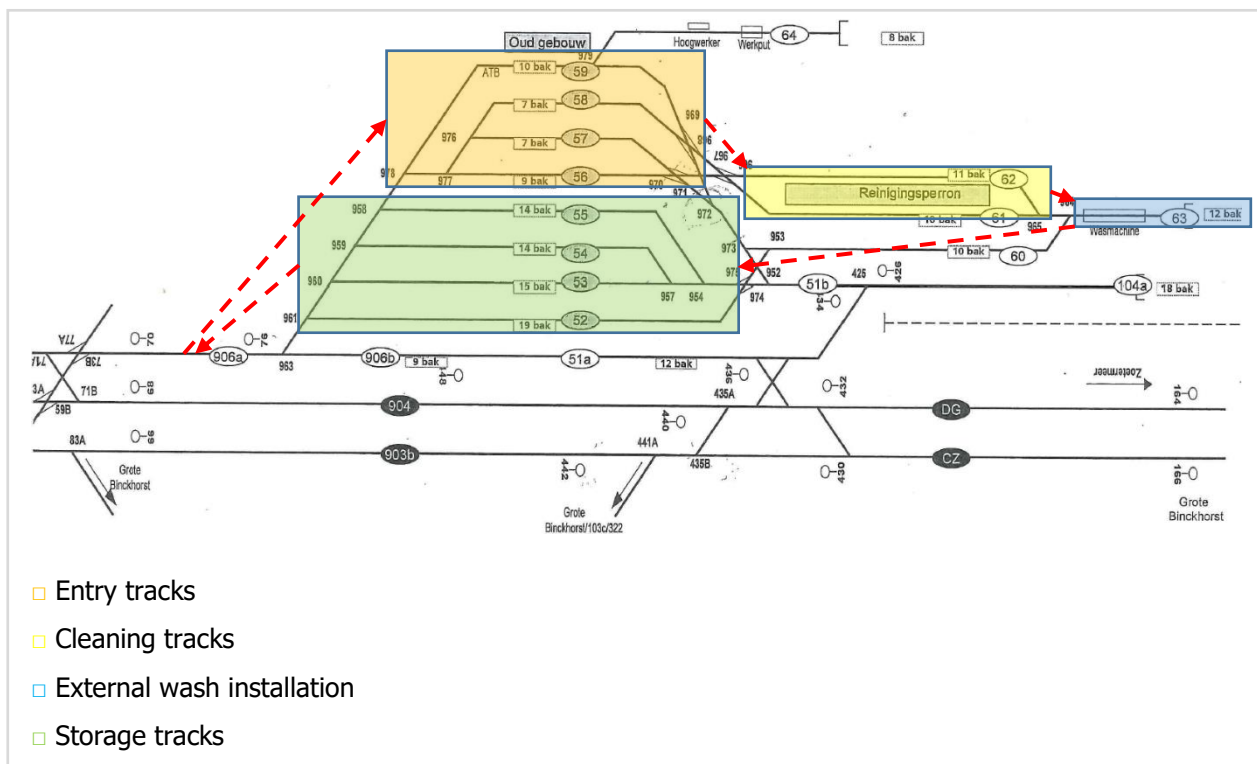


Figure 7-Flow through Bkh.KI

2.5 Number of trains entering the yard

The number of trains visiting the yard highly depends on a lot of factors. In the following paragraph we will discuss the most important factors and give an overview of the average flow entering one specific yard; the 'Kleine Binckhorst' near The Hague.

In an effort to reduce wear on equipment and improve energy efficiency, NS strives to avoid driving trains with too much rail cars. The assignment of train types and train length is different during the days of the week, in correspondence with the expected flow of passengers. The same service at the same time on a different day might be executed by a different train type or length of train. For instance: an Intercity on a specific line in the morning rush hour may consist of 10 rail cars on Monday, and only 8 rail cars on Friday. Another possibility: a service that is executed with bi-level trains on working days, might be executed with single level trains in weekends. These changes result in different trains ending their service at a specific rail station, and therefore different trains enter the neighbouring yard. These fluctuations are however in a weekly pattern.

Approximately each 2 months small changes are made to the timetable and the assignment of trains to services, resulting in a new circulation of trains. These modifications alter the weekly pattern of trains.

The previous two factors influence just the theoretical planning. For data analysis it is disadvantageous that there are already that much adjustments of the planning. The biggest factor however in diversity of trains entering the yard are caused by daily operations and disruptions in the planned service. The target is to execute 83% of all services in the exact composition as prescribed, but even if this target is met, 17 percent of all incoming trains contain at least one different trainset than planned.

At first an analysis is done after the differences between theory and practice. For this part of the analysis the entire 'Binckhorst'-yard is considered. In practice this yard consists of two individual yards, 'Kleine Binckhorst' and 'Grote Binckhorst' which are separated from each other by the main line crossing through the middle, but since theoretical values are based on final destination of a train and both yards serving the same station, it is considered as one. The theoretical data is compared with data from the trains GPS signals. From a database of all GPS coordinates of all trains in the Netherlands a sample is taken for each train's location at 2:00 AM. The GPS coordinates are compared to the location of existing service locations, and trains are assumed to be in the nearest yard.

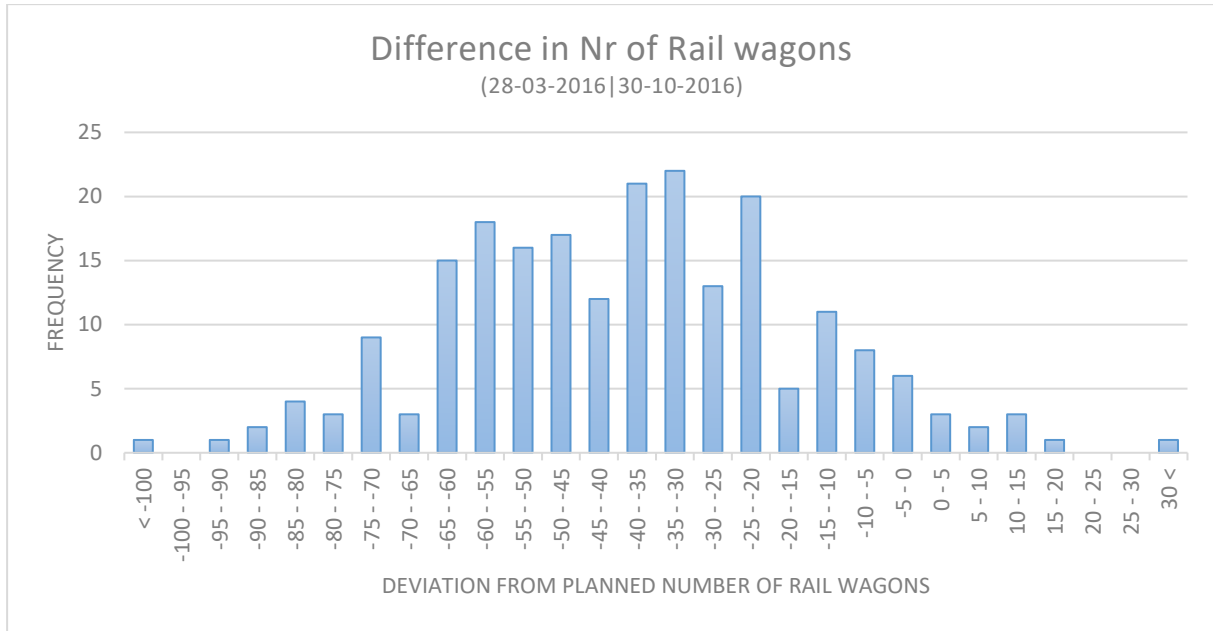


Figure 8-Rail cars per day at Bkh (multiple time table revisions)

By taking the difference between the actual and the planned number of rail cars, the weekly and monthly fluctuations are taken out of the equation. The deviation from the planned situation is depicted in Figure 8.

The sample mean is calculated by

$$\mu = \left(\frac{1}{n}\right) * \sum_{i=1}^n x_i \quad [1]$$

Where n is the sample size and x_1, \dots, x_n are the n sample observations.

Using the theoretical data and GPS data this leads to the following sample mean:

$$\mu_{\text{planned}} = 196,68$$

$$\mu_{\text{actual}} = 158,70$$

The sample standard deviation is calculated by

$$\sigma = \left(\frac{1}{n-1}\right) * \sum_{i=1}^n (x_i - \mu)^2 \quad [2]$$

Where μ is the sample mean, n is the sample size and x_1, \dots, x_n are the n sample observations.

Using the theoretical data and GPS data this leads to the following sample standard deviation:

$$\sigma_{\text{planned}} = 0$$

$$\sigma_{\text{actual}} = 2,685$$

When differentiating the separate days of the week, one can see a clear difference in the theoretical occupation of the yard, but reality seems to follow a similar trend, as is visualised in Figure 9.

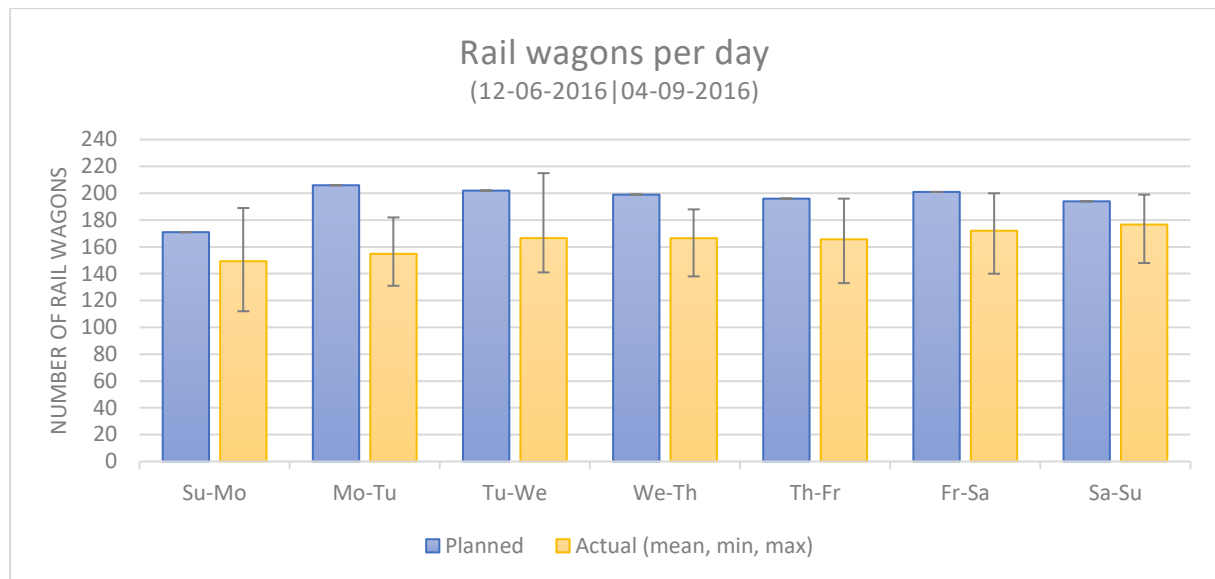


Figure 9-Rail wagons per day of the week at Bkh. (single time table revision)

The consistency in total volume of the demand is not very good, but since trains consist of fixed train units, the length of each entering train unit is also significant. One trainset cannot be divided over slots on different tracks. From now on we will turn the focus toward real data from the GPS instead of theoretical values. The focus will also be narrowed towards the 'Kleine Binckhorst', the part of the yard that will be subject of this research.

In Figure 10 the types of trains entering at Kleine Binckhorst are depicted. The bars represent the mean value of the amount of train units of one type in the yard at one night. The range shows the minimum and maximum amount of train units of that type in the yard at one night.

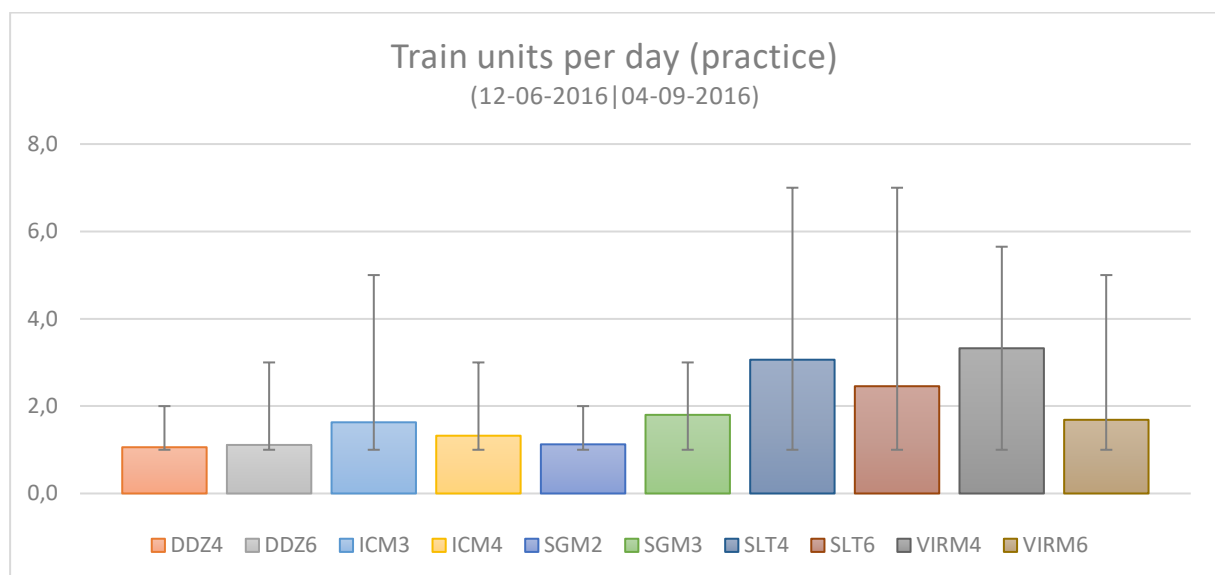


Figure 10-Traintypes at Bkh.KI

Figure 10 shows a large range of various train types. Especially for longer type of trains this results in a significant impact on the total amount of rail cars. Figure 11 confirms that the large range for one train type is not caused by one single outlier, but that there is really a widely spread occupation.

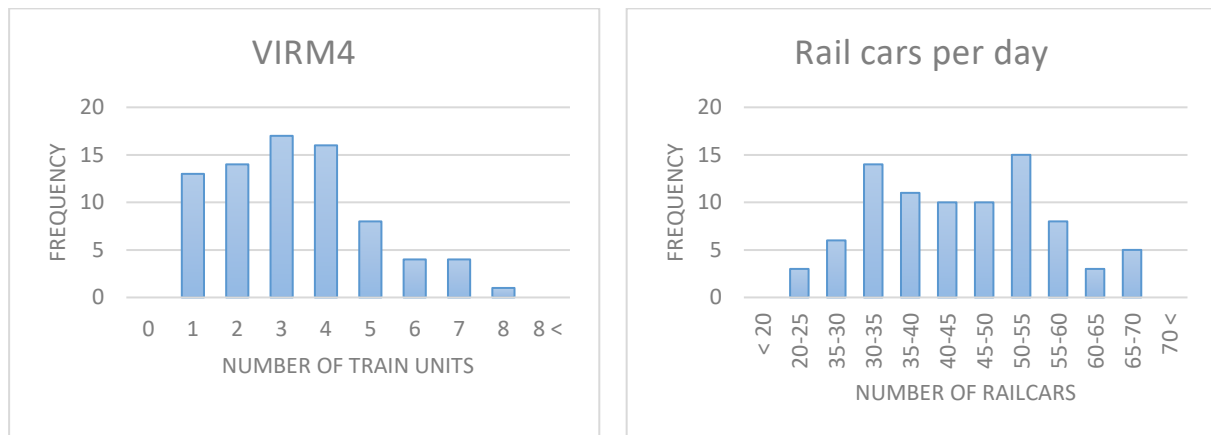


Figure 11-Distribution of VIRM4 (left) and total number of rail cars (right) at Bkh.KI, [12-06-2016 | 01-09-2016]

The arrival sequence of trains in the simulation is created by R. van den Broek [6]. The similarities and differences with the described statistics of reality will be elaborated further down in this report.

Table 2-Distribution of train types

	DDZ4	DDZ6	ICM3	ICM4	SGM2	SGM3	SLT4	SLT6	VIRM4	VIRM6
Bkh.kl	5,7%	6,0%	8,8%	7,1%	6,1%	9,7%	16,5%	13,2%	17,9%	9,1%
Ctw	5,8%	4,0%	6,5%	4,9%	6,7%	13,6%	20,5%	23,5%	7,3%	7,2%

2.6 Arrival times

Trains arrive after the evening rush hour. The first train comes in around six o'clock, and the flow of entering trains is assumed to be continue. From midnight until quarter to three the entering flow continues at a higher pace and around 2:45 AM nearly all trains are in the yard. While planning a service, a follow-up time of at least 5 minutes should be observed to meet the physical constraints of the yards entrance and the connecting tracks to the main line.

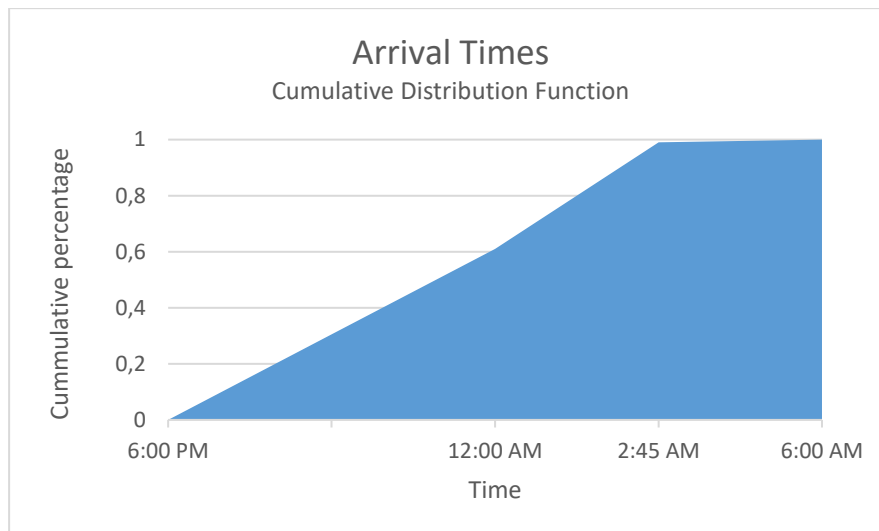


Figure 12-Cumulative distribution function of arrival times

2.7 KPI's

In many storage problems KPI's like reshuffles/reshuffle occasions, capacity shortage and transport equipment workload are mentioned in literature to be most important. Other factors could be computation time or equipment utilisation rate. For the TUSP the same type of KPI's could be applied. One main difference: train units have their own propulsion and do therefore not depend on any resource to move it towards its next destination. Trains do however use the same infrastructure to reach their destination. Only a limited number of vehicles can enter or leave an area of the yard per time unit, and movement inside the yard restricts movement of other vehicles. Besides, for each movement a driver is needed, which is a limited resource as well. These restrictions resemble the restrictions resulting from the use of one or more stacking cranes per block. The rail and driver can be seen as a limiting resource just like a crane.

The overall aim however is bigger than just reducing the number of drivers and shuffles. The main objective is reducing labour costs, decreasing processing time and maximising handling capacity. Also the time required for obtaining a solution is important, since it is beneficial to re-run a problem when the input changes due to disturbances. These objectives can be covered by the KPI's for this research:

- Dynamic storage capacity
- Processing time of trains
- Occurrence of unfinished maintenance
- Calculation time

2.8 Conclusion

Several maintenance tasks take place in a commuter train service yard. In this research internal cleaning, external cleaning and safety checks are considered to be most important with respect to their influence on the dynamic track capacity. Besides the maintenance tasks, the type of trains entering play an important role in the usable capacity. A lot of external factors, over longer and shorter time period influence what sequence of trains arrive at a yard, causing the problem to be less predictable. A distribution from real world data is given, which could be used as input for a model.

3. Literature

In this chapter the existing literature on related train parking systems and rail yard problems is reviewed. While reviewing literature, research questions 2 and 3 will be answered in this chapter. Some research on rail yard problems is performed by other students and staff at NS. The thesis of R van den Broek (2016) [6] on behalf of NS and Utrecht University gives a clear review of the existing literature. His review is gratefully used as a basis for this research.

3.1 General shunting problems

Shunting problems appear in multiple modes of transport. Often this kind of problems are rail based, since those systems are restricted by switches limiting free movement of vehicles, but the rush hour problem is strongly related to shunting problems. One of the earlier shunting problems discussed in literature according to van den Broek [6] is the problem of dispatching trams from a depot. Blasum et al. [2] have studied the assignment of tram storage in a depot with dead end tracks (first-in-last-out tracks) such that the number of shunt moves - moving a tram from one track to another – for the retrieval of trams is minimized. They have shown that this problem is NP-hard and have proposed a dynamic programming algorithm to solve small instances of the problem. Winter and Zimmer [27] investigated the tram retrieval problem extended with the assignment of arriving trams to tracks. Besides introducing an Integer Linear Program (ILP) to find the optimal solution, the authors focused on real-time decision-making to handle arrival delays of the trams. They have developed a number of heuristics that combine real-time information and the optimal solution computed with the ILP. These heuristics yield near-optimal results in less than three minutes for instances consisting of thirty to fifty trams.

The stored tram assignment problem introduced by Blasum et al. was shown by Eggermont et al. [12] NP-hard even if each track in the depot contains at most two trams. Furthermore, they have shown that the extended tram assignment problem described by Winter and Zimmer remains NP-hard when the instances are restricted to tracks that contain at most three trams.

A survey of research on train sorting problems in shunting yards is provided by Gatto et al. [15]. In these problems, a single arriving freight train is split into individual cars. The cars have to be parked on several tracks, each having a different final destination. The freight cars are not self-propelled, but are parked by pushing them one by one over a hump. As the cars roll downhill, they are guided through a tree of switches to arrive at their designated track. The problems have different objectives, such as minimizing the number of shunt moves or the number of tracks used. A broader overview of freight train sorting problems is presented by Boysen et al. [5]. Their survey includes practical aspects such as

solution robustness and recovery, as well as research on train sorting problems with departure lateness minimization objectives.

The topic of robust train sorting is central to the work of Cicerone et al. [6]. They propose recovery rules that can be applied to their generated feasible solution to cope with disturbances in the input data, such as track unavailability or an unexpected car order in the incoming train. The authors have shown the trade-off between robustness and optimality, as well as the difficulty of creating solutions that are robust to multiple types of disturbances.

G. Di Stefano and M. Love Koči (2004) [9] explicitly treats the problem of arranging the trains in a correct order in a commuter train shunting yard. The authors assume trains of the same type and equal length, and tracks of equal length, which enables them to describe the occupation of tracks with simple binary values. They propose some simple algorithms to cope with the track assignment problem with a binary search. The choice of track does not depend on trains arriving later. No measure is given for the performance of the algorithms.

Haijema (2001) [17] developed a heuristic for the shunting operations in a station. Prioritising takes place based on the 'flexibility' of a train; trains that have to perform critical moves around the yard are assigned a place first. Furthermore, the length of a train is considered important. Since long trains are more difficult to place on a track with limited capacity, long trains are assigned a place prior to short trains of the same flexibility level. The method results in less repositioning of vehicles.

3.2 Passenger train shunting and maintenance problems

According to multiple authors, the train unit shunting problem (TUSP) was first introduced by Freling et al. [14]. They match arriving train units to departing trains and assign these units to a track at a shunt yard. Train units are self-propelled and multiple train units can be coupled to form a single, longer train. The authors use a decomposition approach in which a train unit matching is constructed first. In the matching problem, parts of the arriving trains are assigned to corresponding blocks of train units in departing trains. The objective of this sub problem is to find a matching that minimizes the number of splits and assign each train unit to exactly one position in a departing train. The corresponding mathematical model is solved using the standard MIP solver CPLEX. A column generation approach, based on assignments of sets of train parts to each track, is used to find a feasible parking plan, then the parts assigned in the matching problem are parked on a track in the shunt yard. To solve the pricing problem - construct a set of train parts that fits on the track such that each train part can leave on time without being blocked by another train - the authors propose a dynamic programming algorithm. The routing of the train parts on the shunt yard is not taken into account. They generated a shunt plan for a typical weekday at the shunt yard in Zwolle, consisting of eighty train units to be parked, in roughly half an hour.

Although Lentink et al. [26] use a decomposition approach similar to Freling et al. to solve the TUSP, they include the routes taken by the trains, and decompose the problem in four steps. First a matching

is determined using the algorithm proposed by Freling et al. A graph representation of the shunt yard is presented in their study, which is used to estimate the routing costs from and to each shunt track. These estimates are used in the third step, the parking sub problem, to improve the column generation approach proposed by Freling et al. Finally, the actual routes are computed using the graph representation and the track occupation resulting from the previous step. Trains can move simultaneously and the entire path of a train movement is reserved for the duration of the move. The routes are computed sequentially and the order of evaluation is improved by a local search approach that swaps the order of two movements. The time needed to generate a feasible shunt plan for the shunting yard in Zwolle using their approach, including routing, was around twenty minutes.

Instead of solving all components of the TUSP sequentially, Kroon et al. [20] construct solutions for the matching and parking sub problem simultaneously. This greatly increases the complexity of the problem, resulting in a mathematical formulation for the integrated approach that contains a large number of crossing constraints. Testing the model on a realistic case of the shunting yard in Zwolle revealed that there were over 400.000 constraints, which proved to be too much for the CPLEX solver to find a feasible solution in a reasonable amount of time. To reduce the number of crossing constraints, the authors grouped them in clique constraints. This allowed them to find feasible solutions for their test case. Unfortunately, even with the reduction in constraints, the computation time increases rapidly with larger problems, taking several hours to complete.

An integrated approach has been investigated by van den Akker et al. [1] as well. They propose a greedy heuristic and an exact dynamic programming algorithm to the combined matching and parking problem. The heuristic uses track assignment and matching rules that select the locally best action on arrival and departure such that train units are parked in the correct order for the departing trains. The dynamic programming approach looks at all possible shunt track or matching assignments at each event on the shunt yard, and relies heavily on pruning nodes in the dynamic programming network that are unlikely to lead to the optimal solution as a way to reduce its computation time. In contrast to the model formulated by Kroon et al., both algorithms can include waiting time for the arriving and departing trains at the platforms. Furthermore, the exact algorithm is also capable of shunting a parked train unit to a different track, resulting in much more flexibility in the shunt plans. This property is difficult to include in the linear programming approaches proposed by other authors, due to the exponential increase in variables and constraints, even when allowing each parking interval to be split only once. The greedy heuristic is quite fast, but it is not capable of finding feasible solutions for complex problems. Even with the pruning rules, the exact algorithm requires more than ten minutes to find a plan for a dozen train units, making it hard to use in practice.

In the work of Lentink [23] a practical extension to the TUSP is studied. The train units on a service site have to be cleaned in addition to the normal matching, parking and routing problems. The cleaning sub problem is a crew scheduling problem, in which each train unit should be cleaned by a crew before departure from the site. The first three steps are solved using the methods proposed in earlier work

[14, 26]. The schedule for the cleaning crews is constructed last. The cleaning problem is modelled as a machine scheduling problem without pre-emption, where the machines correspond to the crews. The objective used for the machine scheduling problem is the minimization of the sum of the completion times. A mathematical model based on this formulation, in which the planning horizon is discretized into one-minute blocks, is solved using CPLEX.

Two genetic algorithms (GA's) for the integrated matching, parking and routing problem are presented by Jekkers [19]. Both algorithms include waiting time at the station platform and have genes for the parking locations and the arrival and departure waiting times. One variant of the GA has an extra gene for matching, whereas the other uses a greedy heuristic for determining the matching. The fitness of each member of the population is determined with a deterministic simulation. Routes are constructed during the simulation. This approach is applied successfully to generate shunting plans for shunt yards located near Rotterdam Central Station and Hoofddorp, two major stations operated by NS. The shunting yard in Rotterdam is the largest of the two. Its instance contained seventy train units that needed to be parked, and the computation time was fifty minutes.

An integral approach is used by Jacobsen and Pisinger [16] to solve a train parking and maintenance problem. Each train has to be maintained at one facility or workshop located on the service site and is parked in the yard during the remaining residence time. Three metaheuristics, Guided Local Search, Guided Fast Local Search and Simulated Annealing, are proposed by the authors to construct schedules without trains blocked by each other, no departure delays and minimized make span of the service tasks. Their results show that the local search approaches provide results close to shunt plans constructed by CPLEX, while taking only seconds of computation time compared to the twelve hours needed by the MIP solver. However, the largest instances contain no more than ten trains, with one maintenance task per train, which is not representative of real-world scenarios. Furthermore, the absence of routing and matching makes their approach not directly applicable to the scheduling problem for the service sites operated by NS.

Van Dommelen [9] considered the TUSP with the extension of service tasks for a specific service site of NS, the Kleine Binckhorst. Given a fixed matching, the goal is to generate feasible shunt plans that include servicing, parking and routing. The order of the service tasks is modelled mathematically as a flow shop problem, which is solved using CPLEX. The resulting parking intervals per train unit are used as input for a tool called the OPG, a tool developed internally by NS to determine both parking locations and routes. A feasible schedule for a test case with 35 train units was found after two hours of computation time. However, at that time the OPG was not guaranteed to generate shunt plans without routing conflict, thus the reported number of parked train units might be an overestimate of the actual service site capacity.

R van den Broek (2016) [6] uses a simulated annealing approach to solve the train unit shunting problem, including some aspects solved by other heuristics such as tabu search. An initial shunting plan is made, all resulting shunting movements are calculated and the cost function with weighted factors

determines the performance of the plan. A single change is made to the plan and all resulting changes to the shunting movements are calculated. The cost function determines if the solution performs better or worse. If the performance is better, the solution is accepted. If the performance is worse, a probability function determines if the solution is rejected or accepted. This iterative process continues until the runtime is over. The algorithms of R van den Broek outperformed the existing tool built by NS, named the OPG. It is able to schedule more trains than the OPG, even if no service task has to be performed. As a result, the incorporated the local search approach into their software. R van den Broek recommends to improve the robustness of the result. The algorithm provides an ideal plan for a given input, but is not able to cope with disturbances in the input.

According to Haahr et al (2017) [16]: *"Other authors have considered different variants of the problem of Freling et al (2005), including additional constraints and decisions such as maintenance operations or station routing. In some cases, the matching of train units between arrivals and departures is given as input and not part of the problem. Otherwise, part of the problem is also to specify which compatible (arriving) train unit is matched to every departure. The remaining part of the TUSP is to find a valid parking plan for the in and out movements specified by the train matching. With exception of Kroon et al (2008), all studies do not integrate the matching and parking problem, but solve them separately."* Haahr et al focusses on the core matching and parking problem and does not differentiate between distinct solutions.

Tomii and Zhou (2000) [29] regard the shunting scheduling problems at a railway depot as a resource constraint project scheduling problem (RCPSP). They propose a genetic algorithm (GA) that does take some practical issues into account, such as routing, maintenance planning, and shunt personnel planning. They assume however that just one train can be parked at one track at a time, which significantly reduces the complexity of the problem. The algorithm proved to successfully create a feasible solution in 5-30 minutes.

Haahr et al (2017) [16] propose 3 methods to solve the TUSP: Constraint Programming (CP), Column Generation (CG) and a Randomized Greedy Construction Heuristic (RGCH). The CP assigns compositions to tracks whenever an event occurs. All possible options are created and a CP solver searches for the combinations that result in a feasible solution. The CP method is able to solve small instances, but larger instances become impractical to solve because of the exponential growth of the possible options. The CG is a method that "only generates variables that have potential to improve the objective function while implicitly considering all non-basic variables included in the formulation". A pricing problem is used to find a favourable set from the selected variables. The CG method proved to be outperformed by all other methods. The RGCH selects a track based on the following criteria: (1) a track with a train of the same type as outmost train, (2) an empty track, or (3) any track with sufficient capacity. The RGCH was able to solve almost all instances within one second.

3.3 Container assignment and relocation problems

The general shunting problems have a lot of similarities with processes in other industries. The naval industry has to cope with storing a lot of assets on a limited area, while maintaining a certain order in the exiting flow. Because of the similarities we will discuss some researches in that area in the following paragraph.

3.3.1 Container stacking strategies

According to Dekker et al (2006) [9] there are three main objectives of a stacking strategy:

- Efficient use of storage space
- Efficient and timely transportation from quay to stack and further destination and vice versa
- Avoidance of unproductive moves

In literature several tactics for stacking of containers could be found. Duinkerken et al (2001) [12] differentiates a stacking method and a stacking strategy. The stacking method relates to the choice of a stacking lane (and thus of a stacking crane). The stacking strategy relates to the choice of a position within a stacking lane. The stacking strategy could refer to the choice of a specific area in the yard. The stacking method to a location within that area. In the track assignment problem only one area with several 'bays' and 'stacks' exist, so stacking methods will not be reviewed in detail, since it is a problem not occurring in the track assignment problem. We will therefore focus on the stacking strategy. Sculli and Hui (1988) [27] investigated a stowage plan on basis of categorisation of containers. They concluded that the number of categories, stacking policy and storage dimensions are the most important factors. The following part gives an overview of some stacking strategies.

Random stacking

The algorithm randomly selects a non-full pile. If the pile is empty or containers are the same size, the container is stacked here. If the containers in the pile are not the same size, a new random location is selected.

Levelling

The stack is filled level by level, first all ground positions are filled, then a second layer is started. The positions closest to the entry point are filled first.

Closest position

The pile closest to the entry point is filled to the maximum, then a next pile is created.

Category stacking

On basis of define categories, for instance outbound containers for a specific area with specific characteristics of a specific ship. The algorithm keeps track of a variable for every combination of lane, ship, and category. First search if non-empty, non-full piles of stacks with the same category exist. If exist, randomly search for another location.

The definition of the categories is based on the weight class, destination, and type of container (Dekker et al 2006 [9], Steenken et al 2004 [28]).

By using category stacking instead of random stacking, the results can be improved significantly. (Dekker et al 2006 [9]) proved a reduction of 80% in percentage of reshuffles, from 46.1% to 8.8%. Thereby, the workload of the stacking crane is significantly reduced.

Maximum remaining stack capacity

A variation on category stacking, developed by Duinkerken et al (2001) [12], using the same categorisation but with an added priority for the assignment of containers to a stack. A basic principle is that containers for different ships cannot be on the same stack-pile. Furthermore, the priority of pile selection is as follows:

- On top of containers for the same quay crane, of the same or a higher category.
- On top of containers for the same quay crane, of a lower category (creating a reversed pile)
- Randomly.

Duinkerken et al 2001 [12] proved a reduction of 10% in crane utilisation by using Maximum RSC. The stacking strategy and category loading have equal contribution to this improvement.

In residence time strategy

Dekker et al (2006) [9] proposed a method that is based on the arrival and departure time of a container. One stacks a container on others if its expected departure time is earlier than that of all containers below it. In this way a pile is created that can be unstacked in a chronological order, minimizing the expected number of repositioning of containers.

3.3.2 The block relocation problem

Container stacking problems generally assume an initial filling, which is not optimized. A significant part of the solution is therefore in the relocation of blocks.

Kim & Hong (2006) propose a location/relocation selection method. $E(s_i)$ represents the total expected additional relocations from stack i in state s_i . $r(c,i)$ represents the number of expected additional relocations if container c is moved. The following example illustrates the procedure for determining the location of a relocated block.

Consider stacks whose states are:

$$s_1 = \langle 6, 1, 4, 0 \rangle, \quad s_2 = \langle 2, 3, 0, 0 \rangle, \quad s_3 = \langle 7, 5, 0, 0 \rangle$$

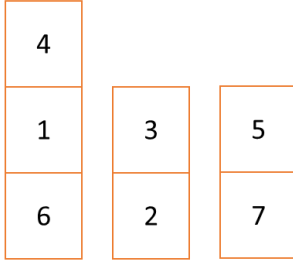


Figure 13-illustration of the stacks

Suppose that block 1 is retrieved from stack 1. Then block 4 must be relocated to either of the other stacks. If block 4 is relocated to stack 2, then the state of stack s_2 becomes $s'_2 = \langle 2, 3, 4, 0 \rangle$. If block 4 is relocated to stack 3, then the state of stack 3 becomes $s'_3 = \langle 7, 5, 4, 0 \rangle$.

Thus,

$$\begin{aligned} R[(4, 2)] &= \{E(s'_2) - E(s_2) + r[(4, 2)]\} \\ &= \{E(\langle 2, 3, 4, 0 \rangle) - E(\langle 2, 3, 0, 0 \rangle) + r[(4, 2)]\} \\ &= E(1, 2) - E(2, 2) + 2 \end{aligned} \tag{3.1}$$

and

$$\begin{aligned} R[(4, 3)] &= \{E(s'_3) - E(s_3) + r[(4, 3)]\} \\ &= \{E(\langle 7, 5, 4, 0 \rangle) - E(\langle 7, 5, 0, 0 \rangle) + r[(4, 3)]\} \\ &= E(1, 4) - E(2, 5) + 1 \end{aligned} \tag{3.2}$$

Then, stack j with a lower value of $r[(4, j)]$ will be selected as the storage location of the relocation for block 4. Kim and Hong (2006) [20] prove an average error rate of 4,7% of the heuristics with categorized blocks compared to a branch and bound method.

Caserta et al (2012) [7] developed an exact method and a heuristic to define the reshuffle and extraction of a box from a pile. Driving the crane in longitudinal direction is time expensive, so the problem is 2D considering only bays, not rows (only width and height).

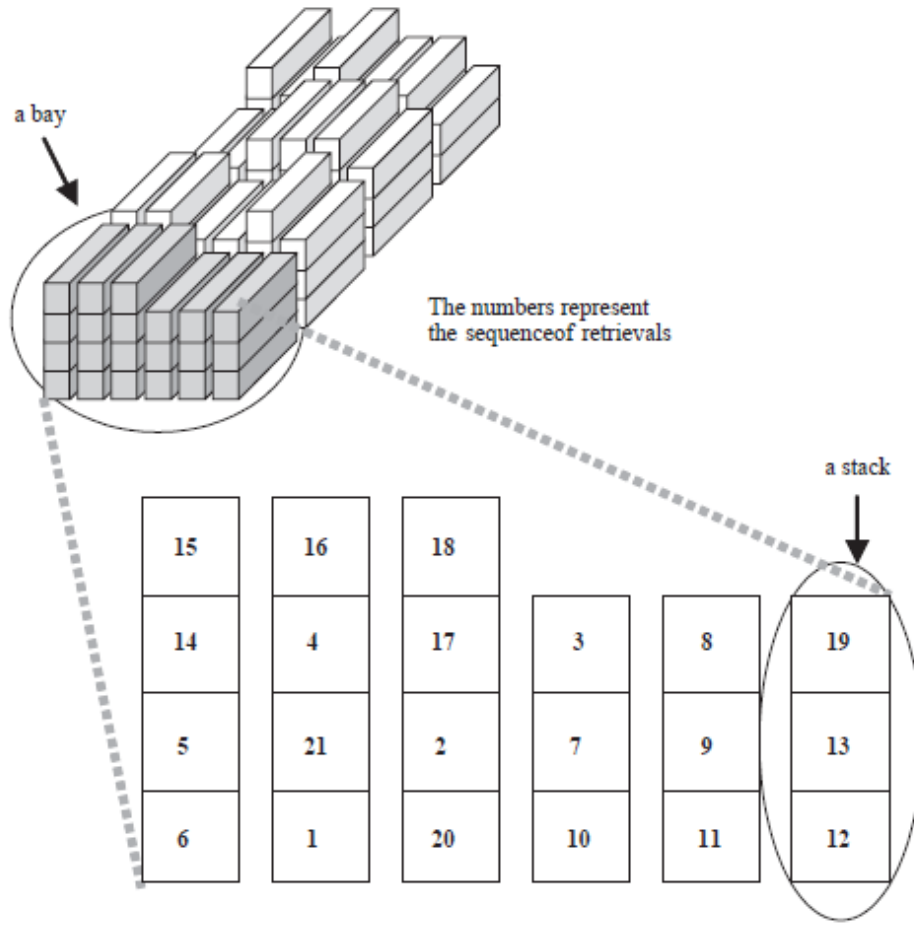


Figure 14-An illustration of a yard-bay of blocks with a fixed sequence of pick-ups (Kim and Hong, 2006)

In determining where block $r \in R$ should be placed, we measure the attractiveness of each stack in the following way: Let us define $\min(i)$ as the value of the block with highest priority in stack i , with $i = 1, \dots, W$. For empty stacks, set $\min(s) = N + 1$. We identify the stack s^* to which the uppermost element of R should be moved as

$$s^* = \begin{cases} \operatorname{argmin}_{i \in \{1, \dots, W\} \setminus \{s\}} \{\min(i) : \min(i) > r\}, & \text{if } \exists i : \min(i) > r, \\ \operatorname{argmin}_{i \in \{1, \dots, W\} \setminus \{s\}} \{\min(i)\}, & \text{otherwise} \end{cases} \quad (3.3)$$

The rule says: if there is a stack where $\min(i)$ is still greater than r (i.e., putting r there will cause no additional forced relocation), then choose such a stack where $\min(i)$ is minimized, since stacks with large $\min(i)$ are valuable. If there is no stack satisfying $\min(i) > r$, then choose that stack where $\min(i)$ is maximized as block r will cause a new forced relocation anyway.

The binary linear programming method takes a lot of time to reach an optimal solution; for a 4 tiers 6 row problem thus can take up to and over one day to compute. The same problem can be solved by the proposed heuristic in under a second, within an average error of 0-5%. Even instances of 100×100 can be solved within a few seconds.

The algorithm of Caserta et al (2012) [7] perform significantly better than the algorithm of Kim and Hong (2006) [20] on the same instances.

3.3.3 The block retrieval problem

Alessandri et al (2000) [2] propose a queuing model to simulate the terminal processes. Their model simulates the entire terminal process, from arriving of containers by various modes of transport, unloading, internal transport, storage and loading of various transport modes.

Lee and Lee (2010) [24] propose a heuristic for retrieving containers that consists of three phases executed one after the other, namely the initial phase that generates a feasible movement sequence with a simple rule, the movement reduction phase that cuts down the number of movements in the sequence, and the time reduction phase that decreases the total working time by adjusting the sequence. The heuristic terminates when the third phase ends.

Phase one achieves a feasible solution by retrieving non-blocked containers directly, and moving blocking containers to the nearest non-full pile. This might lead to a new blockage and rehandling of the same container, but at least result in a feasible solution with simple calculations.

Phase two selects the containers that are handled more than twice, and tries other options. If this option is feasible and results in less handling, it is accepted. If more combinations are available, a 'duplicate' container is produced with proposed new handling sequence, to avoid the possibility of combinations creating new blockages.

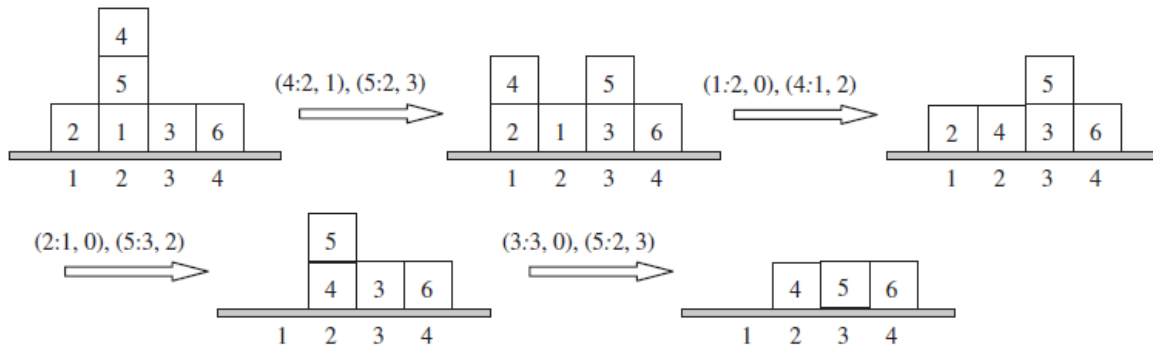


Figure 15-Illustration of the movement reduction phase (Lee&Lee, 2010)

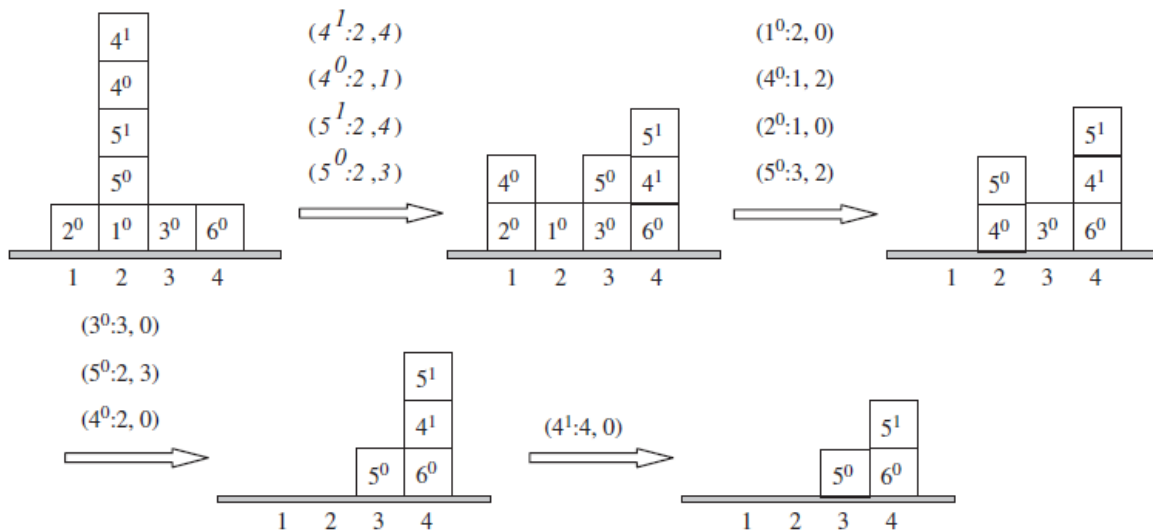


Figure 16-Illustration of an augmented yard and the execution of a super-sequence (Lee&Lee, 2010)

Alvarez (2006) [3] proposed a tabu search for vessel stowage planning. A suboptimal planning is taken as starting point and by making small changes to the plan the neighbouring solutions are explored. The best result is stored and returned after a predetermining number of iterations.

The method of Alvarez (2006) [3] recognizes the advantage of for instance 2 containers stored on top of each other in the stack that are preferably loaded after another. These moves are paired because it is not possible to further improve the sequence.

Kozan and Preston (2006) [21] created an integrated algorithm composed of two sub models to solve problem of container storage allocation and container transfer simultaneously. The integrated algorithm converges faster to an optimal value than the individual models separately. The solution using a generic algorithm for both models outperforms the hybrid algorithm, using tabu search and a generic algorithm.

Lee and Hsu (2006) [23] discuss a pre-marshalling problem based on a multi-commodity network flow model. The nodes and arcs correspond to the time–space structure of the container yard, and containers moving in time–space in the yard are represented with flows. Physical laws that containers have to comply with are specified with constraints. By allowing multiple movements to take place within the same time step, the solution time can be reduced significantly. Constraints are applied to ensure no infeasible loops exist in the final solution, and no more than the maximum number of containers is put into one bay during the time step.

The heuristic solves the problem in two phases. In the first phase, the heuristic attempts to solve for a set of movements that leads to a good final layout using a simple branch-and-bound method and no constraints. The movement arcs are re-used and unused arcs are replaced until a solution satisfies the constraints. This phase runs quickly, but the solution quality is low because it can contain multiple cycles. In the second phase, the heuristic sorts the movements and breaks all cycles by including additional movements. Both phases work iteratively by repeatedly solving variants of the basic model.

3.3.4 Differences between container and rail

Some differences can be obtained from rail and container operations. This paragraph will highlight the most important ones.

According to Dekker et al [9] container stacks operate at 50% average utilisation rate, while rail yards operate at 70% utilisation rate and aim for more. The reduction of remaining free slots makes the reshuffling and retrieval of blocked units significantly more challenging.

Container stacking is a 3D process, while train parking is only 2D. Some literature already reduces the container retrieval problem from lanes, rows and height to just rows and height in order to simplify the calculation. These simplified methods can easily be converted to the rows in which trains are placed 'on top' of each other.

A focus in stacking operations is often at stacking crane utilisation, since that is the bottleneck in the system. The stacking crane is the engine of a container, enabling it to move to another location. This

element is not existing in the rail problem since train units have their own propulsion, however the stacking crane can be seen as a resource needed to move units. The connecting tracks in a shunting yard fit this description of a stacking crane, and these also happen to be a bottleneck in shunting operations. In line with previous statement, the input and output of a container stack depends on the capabilities of the stacking crane and transportation. The input and output of a rail yard depends on free paths towards the yard.

Containers exist in a wide variety of sizes; different length, width and height, but they can all be reduced to the measurements of one or more Twenty foot Equivalent Units (TEU). Most containers fit within the height and width of this standardised unit measure, and else they are classified as exceptional load, which is dealt with in a separated process. Stacks of containers are generally made of containers with the same length, and in many cases even bays of containers only contain the same size, reducing the complexity of the system. So, containers do exist in a few different heights, but a fixed number of containers fit in a stack. Train units however come in various types with a wide range of different lengths, resulting in a wide variation of train units that fit on a track.

Containers are individual units, handled separately. Trains entering a yard can consist of multiple train units, with different 'destinations'. Coupling and decoupling might be needed.

Container stacks generally have a uniform maximum height, restricted by the specifications of terminal equipment such as stacking cranes. However, due to the branching characteristics of rail yards and the length needed for switches, each individual track in the yard might have a different length and capacity.

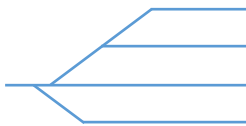


Figure 17-Branching characteristics at a railyard

Most literature on container retrieval problems assume an initial filling of the stack. There is much uncertainty in departure time at the moment of arrival of export containers, making it hard to create a pile with a smart order, so the initial filling in a yard is generally created randomly to replicate the result of this initial uncertainty.

Re-handling of containers is avoided because it is inefficient use of scarce resources. In carousel type yards, the trains move through the yard anyway because they need servicing at a specific location, so service stations can act as a way to ease small scale reshuffling of the yard.

Tracks can be open ended, enabling trains to enter and exit a track on more than one side. The 'pile' of trains can therefore be accessed from 2 sides, where in stacks of containers only the top one is accessible. This improved accessibility of a train track significantly increases the possibilities and complexity of the solutions.

3.3.5 Similarities between container and rail

In previous paragraph became clear that some differences are not that different at all. There are however also some really clear similarities between piles of containers and tracks full of trains. For the sake of completeness, we will summarize them in this paragraph.

For a start, trains are 'stored' on a few tracks, multiple train units on one track block each other's exit path, just like containers are stored on top of each other in one bay.

Train units could be divided in several groups according to type, in which the individual train units are interchangeable, in the same way containers can be grouped according to size, destination and weight group.

The exact arrival time of both a container and a train cannot be foreseen over a longer time span. An indication of the arrival time is available, but this can change until the real moment of arrival. These changes ask for recalculation to be performed periodically or event driven.

A pickup schedule problem in which the quay cranes schedule or a loading sequence is known, resembles a railyard from which trains should exit to be at a platform in time to start service.

3.4 Conclusion

Several general shunting problems and Train Unit Shunting Problems have been reviewed. The basic TUSP, described by Freling et al. [14], treats the matching of arriving and departing trains and the assignment of trains to tracks. Many authors explicitly treat the arranging of trains in a correct order, and do not take other tasks into account. Some authors do take other task into account, but treat the problems separately.

Because of the similarities in the processes, some container problems have been reviewed. Many container problems are solved by using smart heuristics and defining a smart sequence in which actions take place. Advantage of this approach is that it is easy to implement in a problem with regularly changing input.

In this research the heuristics approach of several container stacking problems will be used to create an alternative method for the extended train unit shunting problem as described in paragraph 3.2. By using these heuristics, an integral model of matching, assignment of tracks, assignment to service activities and shunting movements can be made. In next chapter we will go in further detail on how the literature could be used in a model.

4. Solution methods

In paragraph 3.2 the train unit shunting problem is defined as the matching of arriving and departing trains and the assignment of parking slots to trains. As an addition, the extended TUSP will also cover some maintenance tasks and shunting movements. In this chapter we will formulate several solving strategies on the extended train unit shunting problem, and construct a method for replicating the results of these strategies. The combined solving strategies for several partial problems, result in a solution method. These solution methods will be introduced in this chapter, thereby answering research question 4.

4.1 Approach

Jianbin Xin (2015) [32] differentiates two different approaches to solve an equipment assignment problem. The approaches are summarized shortly below.

4.1.1 Analytical approach

Analytical approaches use mathematical optimisation methods to model and optimize the operations. Equipment scheduling and vehicle management are typically considered separately.

4.1.2 Program approaches

The programming approaches use heuristics for weighting of options. Agent-oriented programming focuses the concept of “agent” and the cooperation of multiple agents, typically referred to as a multi-agent system. In the agent-oriented programming, an agent is a computer system that is capable of independent action on behalf of its user or owner and a multi-agent system consists of a number of agents which interact with each other, typically by exchanging messages. The agents are considered equal, and negotiate on behalf of its client in order to maximize the result of the entire system.

In the object-oriented approach, an object is an entity that contains a set of attributes and a set of methods. Attributes are factual descriptions of the object and the methods are functions that enable the object to manipulate its attributes and communicate with other objects. The object-oriented approaches focus on developing a decision support system, in which the effect of different operation policies and parameters on the performance of a system can be evaluated. The detailed control and optimization algorithm can be incorporated as the operation policy of the decision support system.

4.1.3 Approach selection

The three methods mentioned by J. Xin could be applied to terminal scheduling problems, as is done in his own research, but it could also be applied to train unit shunting problems. Literature (Haahr et al (2017) [16]) proved that heuristics generally are more successful in solving the TUSP than an analytical

approach, so the programming oriented approaches are preferred over analytical approaches in this research.

In the train unit shunting problem there is generally only one category of elements; trains. These elements desire some service, but there also exists a shared interest between the systems elements: getting as much as much trains serviced in time as possible.

An object-oriented approach could be setup in line with how a human planner would solve the problem, incorporating logistical processes that proved beneficial in practice. Besides the ability for using proven operation policies, it might also be beneficial for implementation in reality. In this research therefore the object-oriented approach will be used.

4.2 Assumptions

We will assume a general setup of the yard for all solutions. These assumptions are based on the layout of the yard and processes in practice. The system consists of methods and objects with their attributes. The methods are described in paragraph 4.3 to 4.5; the following objects, attributes and methods will be explained in the following paragraph. In the following overview '{ }' denotes the possible characteristics of the attributes, '[']' denotes the units.

- **Trains (active)**
 - Type : {SLT; VIRM; DDZ}
 - Length : [rail cars]
 - Arrival time : [seconds after 8:00AM]
 - Departure time : [seconds after 8:00AM]
 - Cleaning time : [seconds]
 - Train matching procedure
 - Track selection procedure
 - Track switching procedure
- **Tracks (passive)**
 - Type : {entry; cleaning; parking; connecting}
 - Preferred driving direction : {entry; exit}
 - Capacity : [rail cars]
 - Time to cross segment : [seconds]
- **Yard equipment (passive)**
 - Function : {internal cleaning; external cleaning}
 - Capacity : [rail cars]
 - Location
- **Train generator (active)**
 - Train creation procedure

4.2.1 Trains

The set of trains entering the yard is determined in advance. Train types with given length will arrive at a specific time. The set of trains leaving the yard is also given in advance. Departure times are used to determine if a train finished all service tasks in time to start its designated passenger service.

4.2.2 Tracks

One object in the yard are the tracks. Tracks are used by trains for movement around the yard or parking. Some side constraints arise from the characteristics of the tracks.

Tracks have a predefined maximum capacity, divided into slots of a standard wagon length. It is not possible to assign a train to a track if the remaining capacity is not sufficient.

From logistical viewpoint it is assumed to be beneficial to assign a preferred entry and exit point to a track. By obeying the preferred direction, a logic flow through the yard can be obtained in a way that proved to be efficient in practice. In this way existing process knowledge is incorporated in the problem solution. Open end tracks are assumed to have an entry point at one side and an exit point at the other side; the first train entering the track is the first to exit: First In, First Out or FIFO. We apply the FIFO strategy to the 'Kleine Binckhorst' yard. Dead end tracks have only entrance and exit points at one side of the track; the first train entering the track is the last to exit: First In, Last Out or FILO. 'Cartesiusweg' is an example of a yard with FILO tracks.

Tracks have a predefined function: buffer for incoming trains, storage, or maintenance activities. Tracks can temporarily be used for other purposes than pre-defined if the situation demands it. The preferred direction can change with the altered function of the track. Besides the entry tracks, cleaning tracks and storage tracks, there are also tracks that connect those specific tracks to one another. Connecting tracks cannot change function, cannot be used for parking and can only be occupied by one train at a time.

It takes time to move switches and cover distance. The assumed time is independent of the length of the track, the type of switch and the direction. The time it costs to cover a segment of track is set at one minute, and the time to cross a switch is set at 30 seconds. This assumption is common in shunting problems at NS.

For safety reasons all shunting operations should be separated in time and distance at all times. If one shunting operation is active, no other shunting operations are allowed on all neighbouring tracks to avoid collisions.

4.2.3 Yard equipment

A special type of track is the kind that accommodates yard equipment. The cleaning platforms in the yard and wash installation are basically the same as normal tracks, but with additional characteristics. Cleaning platforms have a maximum capacity along the platform, and a predetermined number of cleaners is available. External cleaning of a train takes a predetermined amount of time depending on characteristics of the train. See Table 1 for the range of the processing time.

4.3 Elaboration of solution methods

As mentioned before, the solution methods of this research are heuristics, or a series of assignment rules that create a shunting policy. The intention of this research is to minimize repositioning of units, because it is assumed better to prevent additional handling instead of solving the consequences. The basics of the chosen methods are therefore in line with the methods described in 3.3.1, that use some kind of categorisation to determine a stacking sequence of the units.

All solution methods use a train generator to create new trains. After creation, the trains monitor their own progress and assign tasks to itself when it is their turn.

Traingenerator PDL

At T = arrivaltime	0.	Incoming train is created by generator
		With characteristics (length, type, tasks...)

In the following part the specifics per method are described.

4.3.1 Solution method 1: Categorization on train type (Type Based Strategy)

The first method is based on categorised stacking with time constraint by Dekker [9]. The method purely separates train units of different types and lengths, and groups them type by type on the tracks. The track selection will be as follows:

- Train units with no activities are stacked on basis of train type (SLT, VIRM, DDZ).
- Train units with same 'destination' (activity in yard) are stacked on the same 'pile' (track).

Train units can be stacked on top of another type if no other options are available. Train PDL 1

At creation	1.	Assign category based on train type
When first claimer for task	2.	Select queue containing elements with the same categorization Check if available track length > train length If no queue meets requirement, search queue with service task (= early departure) If still no queue meets requirement, select empty queue
	3.	Enter selected queue
	4.	After finishing service activity, repeat from 2

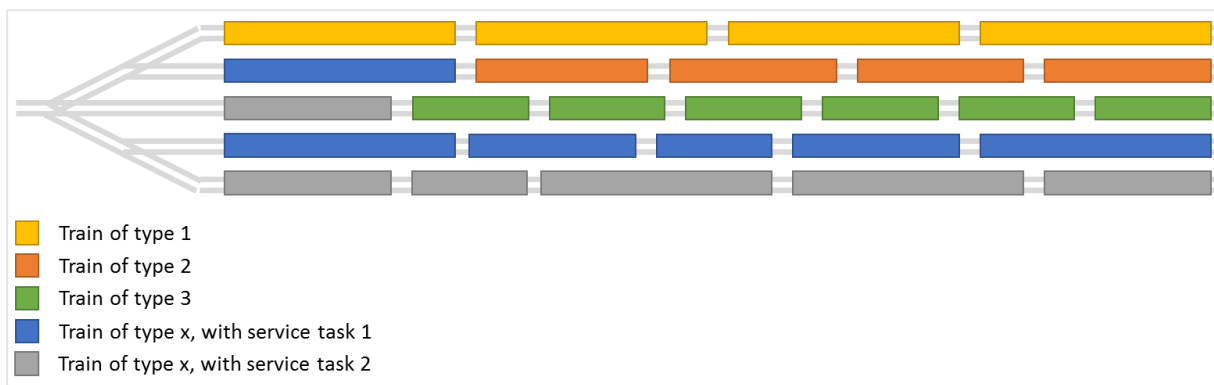


Figure 18-Example of filling of a yard using method 1

4.3.2 Solution method 2: Selection on departure time (In Residence Time Strategy)

This method is based on the retrieval problem by Kim and Hong (2006) (paragraph 3.3.2 The block relocation problem).

Assume a given sequence of trains leaving the yard with a predetermined length and type of train. Trains entering the yard are linked to the first leaving train of the same type. The entering train is assigned a priority corresponding to the departure sequence. The first train leaving gets priority 1, the

second train leaving gets priority 2, etc. Once a leaving train is linked to a train in the yard it will be marked, so it will not be linked twice.

Trains are assigned to their next location based on the assigned priority. For selection of a service activity only trains that are not blocked by other trains are considered. The train with the lowest priority is selected to go first to the next location, once this location becomes available. This method results in a small iterative process that improves the order of trains in the yard at each movement.

Train PDL 2

At creation	1.	Assign priority based on departure sequence
When first claimer for task	2.	Select queue which last element had a lower priority (FIFO)/higher priority (FILO) than itself, and has the least difference in priority. Check if available track length > train length If no queue meets requirement, select empty queue
	3.	Enter selected queue
	4.	After finishing service activity, repeat from 2

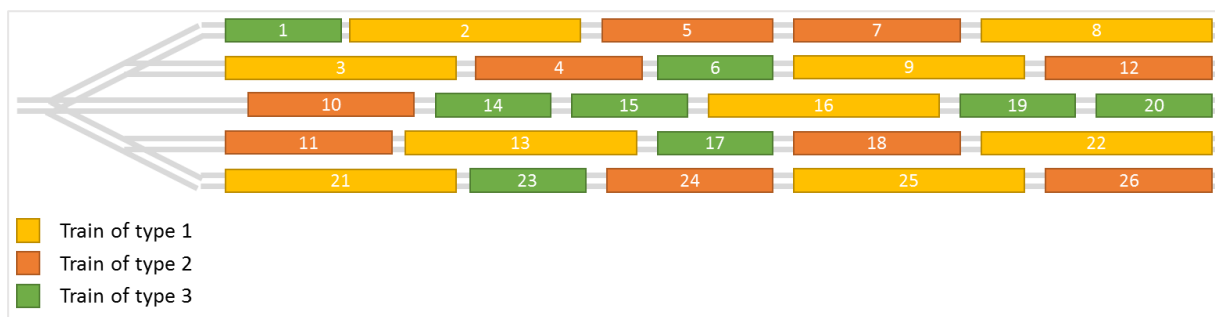


Figure 19- Example of filling of a yard using method 2

4.3.3 Solution method 3: Selection on departure time including reshuffle

The method largely resembles solution method 2, except at step 2: if no track available, take first in row out, place current train in that spot and place replaced train in front of current train. This results in the following PDL.

Train PDL 3

At creation	1.	Assign priority based on departure sequence
When first claimer for task	2.	Select queue which last element had a lower priority (FIFO)/higher priority (FILO) than itself, and has the least difference in priority. Check if available track length > train length If no track available, take first in row out, place current train in that spot and place replaced train in front of current train.
	3.	Enter selected queue
	4.	After finishing service activity, repeat from 2

4.4 Service activities

One of the contributions of this research is the combination of the track assignment problem with several service activities. All service activities are assigned on the same basis as the 'service activity' parking. Or in other words: trains are parked at each service location using the same priority as parking for storage. If necessary, trains can be split at several locations. The same is valid for combining two or more train units. Trains can enter and leave as a consist of up to four train units, so multiple splits or combines could be needed in one train.

Besides train matching, cleaning is a frequent occurring service activity. Trains will stay for a specific time at a specific track while being internally or externally cleaned.

Safety and maintenance checks generally can be executed at an arbitrary track. The task is mainly a personnel assignment problem, but it is not of large influence on the flow through the yard. It is therefore not taken into account in this research.

4.5 Conclusion

In this chapter the assumed setup of the system is determined. The system contains tracks, trains and yard equipment. Additionally, a programming based approach is chosen to solve the problem. Three different heuristics will be used to improve the solution of an extended train unit shunting problem with service activities. The first method is based on ordering on train type. The second method is based on ordering on departure time, and the third method is a variation on the second method, including reshuffling of vehicles.

5. Application of the solution methods in a model

In the following chapter, the methods of chapter 4 are applied to the actual setup of existing rail yards using a simulation. The setup for this simulation will be described, thereby answering research question 5. Two different types of rail yards will be reviewed. First the setup of the 'Kleine Binckhorst' yard will be treated, which will later be adapted for the 'Cartesiusweg' yard

5.1 General setup

The TUSP can be described like a 'job shop problem', which means that limited resources should process the jobs that enter the shop. H. P. M. Veeke and J. A. Ottjes developed a software tool specifically for these kinds of problems. We therefore create a discrete process simulation of a rail yard using TOMAS and Delphi software, in which tracks are resources and trains are jobs to be handled.

- Each track is modelled as a resource with queue
- Each train is modelled as a job with its own process
- Each maintenance task takes place at a track and delays the trains process
- Arrivals and departures are imported from an input file created by excel
- Problem processing procedure is hardwired in simulation
- Visual representation of track occupation
- Each track has a predetermined maximum capacity

The simulation is performed for one day. The planning period in the datasets spans from 8.00 AM till 8.00 AM the next day. The arrival and departure sequence of trains is determined in advance. In order to be able to make a comparison with other methods, the same dataset is used as R. van den Broek [6] and F. Wolfhagen [31]. The datasets contain all information about arriving and departing trains, including type, length and composition. Due to the predetermined arrival and departure times in the dataset, and the fixed assignment procedures of tracks, the process is entirely deterministic. As a result no replications of the same input are needed since it would result in the same solution.

There are 200 artificial instances in the dataset; 10 instances for each even number of arriving train units from 2 to 40 (2,4,6,...,38,40). Within the 10 instances per number of train units, the train units are the same in type and length, but the composition of trains are different, as are the arrival and departure times. The sets are named according to the number of arriving trains in the set and an index number (1 to 10), so for instance the first of ten sets with four arriving trainsets is called 4-1, and the fifth of ten sets with 20 arriving trainsets is called 20-5.

T in [s]	Train Type	Train set 1	Train set 2	Train set 3	T out [s]	Train Type	Train set 1	Train set 2	Train set 3
48131	SLT	4	6	0	75783	VIRM	4	4	0
53542	VIRM	4	4	0	79661	SLT	4	6	0

Table 3-Example of data; first of ten sets with four arriving trainsets (set 4-1)

All trains arrive and depart within the determined timespan. A selection has been made such that no trainset should depart before a trainset with the same type and length has arrived in the yard.

5.1.1 Trains

Trains are created by the train generator at a pre-determined time. The arrival and departure time is given by a dataset, which is identical to the one used by R van den Broek [6]. We consider various types of trains, as is explained in paragraph 2.1. Only a limited amount of train types occur in the dataset, namely: SLT, VIRM, and a few DDZ. The type of train and the consist is also pre-determined in the dataset. Each train is created as a Tomas Element with its own process. This will mean that each train will continuously monitor its own process and assign tasks to itself at appropriate times, based on decision parameters. The processes of the train are different for each solution method, and are therefore explained in paragraph 5.4 and 5.5.

5.1.2 Tracks

We consider two rail yards with a given track layout, see Figure 2 and Figure 3. The layout is fixed and cannot be altered. The capacity of the tracks is given in * Conflicting data available; shortest option is used

Table 4 and Table 5.

Track number (simulation)	Track number (reality)	Length of track [m]	Length of track [wagons]
1	59		10
2 (/42)	58		7
3 (/43)	57		7
4 (/44)	56		9
21	61	247	9*
22	62	247	9*
31	55		14
32 (/12)	54		14
33 (/13)	53		15
34	52		19

* Conflicting data available; shortest option is used

Table 4-Track capacity at Bkh.kl

Track number (simulation)	Track number (reality)	Length of track [m]	Length of track [wagons]
1	90	227**	8**
2	91	332**	12**
3	92	162**	6**
11	251	196	7
12	252	210	7
13	253	205	7
14	254	219	8
15	255	221	8
16	256	235	8
17	257	293	10
18	258	260	9
19	259	264	9
20	260	316	11
21	261	369	13
22	262	304	11
-	263	No parking allowed	-
23	264	224	8
24	265	161	5
25	266	162	5
26	267	178	6
27	268	146	5
28	269	147	5
29	270	162	5
30	271	163	5
-	272	No parking allowed	-

** No data available; measured using google maps

Table 5-Track capacity at Ctw

Tracks are simulated as waiting queues. The train arrival pattern is described in a file which is automatically loaded into the program. At the Binckhorst yard there are 4 tracks predetermined to receive arriving trains (track 56-59 in Figure 21). The tracks along the cleaning platform (tracks 61-62 in Figure 21) are reserved for internal cleaning, and 4 other tracks (track 52-55 in Figure 21) are intended to be used for storage of trains after executing maintenance tasks. The tracks are created as a Tomas Semaphore, which is a queue with limited capacity of which the capacity is claimed for undetermined amount of time. The train releases the capacity at departure from the track.

5.2 Train selection procedure

The main difference in the solution methods can be captured in the way arriving trainsets are matched to departing trains. We therefore discuss the train selection procedure separately per method.

5.2.1 Type Based Strategy (solution method 1)

At creation of the train, it is assigned a value based on its type. All trains of the same type and length receive the same number. Longer trains are assigned a lower number, e.g. a higher priority to move through the system (in accordance with Lentink et al. [26]), since existing evidence indicates these are more difficult to place.

Nr.	1	2	3	4	5	6
Type & Length	VIRM 6	DDZ 6	VIRM 4	DDZ 4	SLT 6	SLT 4

Table 6-Train types and their corresponding values in the simulation

5.2.2 In Residence Time Strategy (solution method 2 and 3)

At creation, the train checks a pre-determined array based on the dataset, from which it derives the order of leaving trains. According to this sequence it can assign priority to the train (10 for the first train leaving, $10 \cdot X$ for the X^{th} train leaving). The priority considers the type of train entering and leaving. When a leaving train is assigned, the value will be deleted from the array so it will not be assigned again. The next train of the same type will be assigned the priority corresponding to the next leaving train of that type.

Trains enter regularly as consist of multiple train units, and exit as consist of (other) train units. Trains can exist of at most 4 coupled train units. At entry is checked if the type of train unit in front appears in the corresponding position of a leaving train. If no leaving train has this type of train unit in the right position, the next train unit in the consist of leaving trains is checked. If the train unit exists in a leaving train, the train gets the priority of the first leaving train meeting this condition. Next is checked if any of the other coupled train units appear in the same order in the selected leaving train. If so, the train units can stay coupled. If not, the train is split so the last corresponding train unit stays in the consist.

Figure 20 gives a visual representation of the options that are checked when matching an arriving train to a leaving train. At first is checked if an arriving train is from the same type as the considered departing train. If not, the next departing train is regarded. The matching procedure checks each matched type from top to bottom according to their length. For instance, if a train with four trainsets enters, it checks if the considered train is equal for all four trainsets. If not, it checks if the first three trainsets are equal to the first three trainsets of the considered leaving train. If not, the first two and the first trainset are considered, regardless of the total length of the arriving and departing trains. If a arriving train is longer than the matched part of the train, it will be split as will be described in paragraph 5.6.

A trainset has length 0 if there is no trainset on that position in the train. As a result an arriving train with 2 trainsets can correspond with an departing train on all 4 positions.

It should be remarked that trains reverse direction in the yard, and therefore the rear trainset of the matched train should correspond to the front trainset of the leaving train. Since the trains at Ctw are retrieved in the opposite order in which they were stored, the matching is based on the rear part of the departing train instead of the front part, hence the difference in Figure 20.

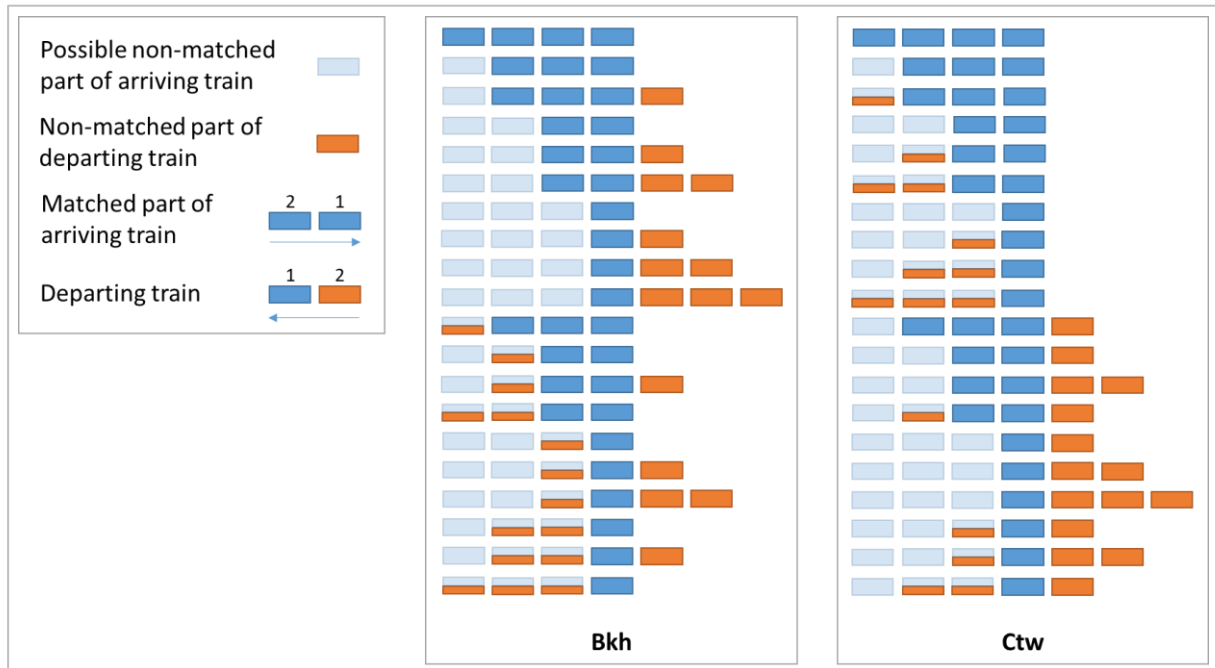


Figure 20-Options for combining and splitting incoming trains (left: Bkh, right: Ctw)

Each time a train unit is assigned to a consist of a leaving train, that part of the leaving train is deleted from the list of leaving trains (since it is assigned already). In this way, the method keeps trying to complete the first leaving train, before checking other options. This method is beneficial to preserve the optimal sequence in the yard, however it can also cause trains to be split unnecessary if a train leaving at a later timeslot is exactly identical to the leaving train, while the first leaving train only resembles one train unit.

In the initial setup (*IRTS – Limited Split*), an effort is made to minimize the number of splits. If an arriving train should be split to match the consist of the leaving train, the matching procedure is continued to check if any other match could be made that does not need a split action. In all other setups, the matching procedure is completed at the first match.

5.3 Track selection procedure

During movement through the yard, a train will assign itself a new location based on assignment rules. These rules vary per location and some strategies use variations of the same rules. In the following paragraph we will discuss them.

5.3.1 Entry track

If the train cannot be placed in the preferred order, the arriving trains are assigned to the track with the most remaining capacity left. If the entry tracks have insufficient capacity to cope with the number of entering trains, it is checked if any of the predetermined 'storage tracks' are unused. If so, the capacity is of the specific track set to zero, and a copy of the track is used as entry track. Trains entering by storage track get priority in assignment to the cleaning track, in order to restore the storage capacity as soon as possible.

If no space is available and no space can be created, then trains are parked on an imaginary track with infinite capacity.

5.3.2 Cleaning track

The priority of trains in front of a track is used to determine which one moves first to the next location. Trains have specific characteristics pre-assigned, such as a type, length and required internal cleaning time.

While waiting at the entry track; a train will check:

1. If there are no trains in front of it on the entry track
2. If it is the train with highest priority (lowest number) of all the trains in front of an entry track
3. If there is enough capacity available along the cleaning platform.
4. If the path towards the cleaning platform is not in use by another train.

If all four constraints are met, the train is moved. The move takes one minute per travelled piece of track and half a minute per crossed switch. During this time all connecting tracks are claimed by the train; constraint 4 makes sure that no other train moves in the same area during that time.

Trains are first assigned cleaning track 1 unless its capacity is not sufficient, then they will be assigned to cleaning track 2. Trains stop behind each other on the assigned track and are not moved until they are cleaned, so only if the last train leaves, the track can be used to full capacity again.

After a train is assigned to a cleaning track, cleaners are assigned. If the cleaners are not available; the train waits at the cleaning track. Cleaners are assigned according to the entering sequence. Once cleaners are assigned, the train stays for the cleaning time divided by the number of cleaners that where assigned. While at the cleaning track, the train will stay at its initial position, and will not be moved to the end of the track if the train in front leaves.

At the Ctw yard all storage tracks act as cleaning track. Trains will leave the entry tracks as described above, select a storage track as described below, and wait at the storage track until it is the train with the highest priority (lowest number) that has to be cleaned. After assignment of cleaners the train is standby for the cleaning time divided by the number of cleaners that where assigned. It does however not have to move after being cleaned, since it is already at the storage track.

5.3.3 Storage track

Once cleaned, trains are assigned to tracks for storage. Trains are assigned according to priority. The exact assignment method differs per strategy. First the general processing method is explained. We will discuss the assignment methods at the end of this paragraph.

If no storage capacity is left on the preassigned storage tracks, it is checked if entry track 4, 3 or 2 (track 56, 57 or 58 in Figure 21) is empty. If so, its capacity is set to zero and a copy of that track is used as a storage track. At these tracks no specific order is obtained.

When the storage track is determined, a path towards the track is claimed. All switches and connecting tracks that a train will cross are claimed by the train. The external wash installation is situated along the way towards the storage tracks. If external cleaning is taken into account, a normal distribution determines if a train receives external cleaning. There is a chance of 1 in 7 that a train need cleaning, in which case only the side of the train is cleaned and not the front.

The train reverses direction at the cleaning track. The turnaround time is determined by the type of train, and the walking time of the train driver which is based on the length of the train.

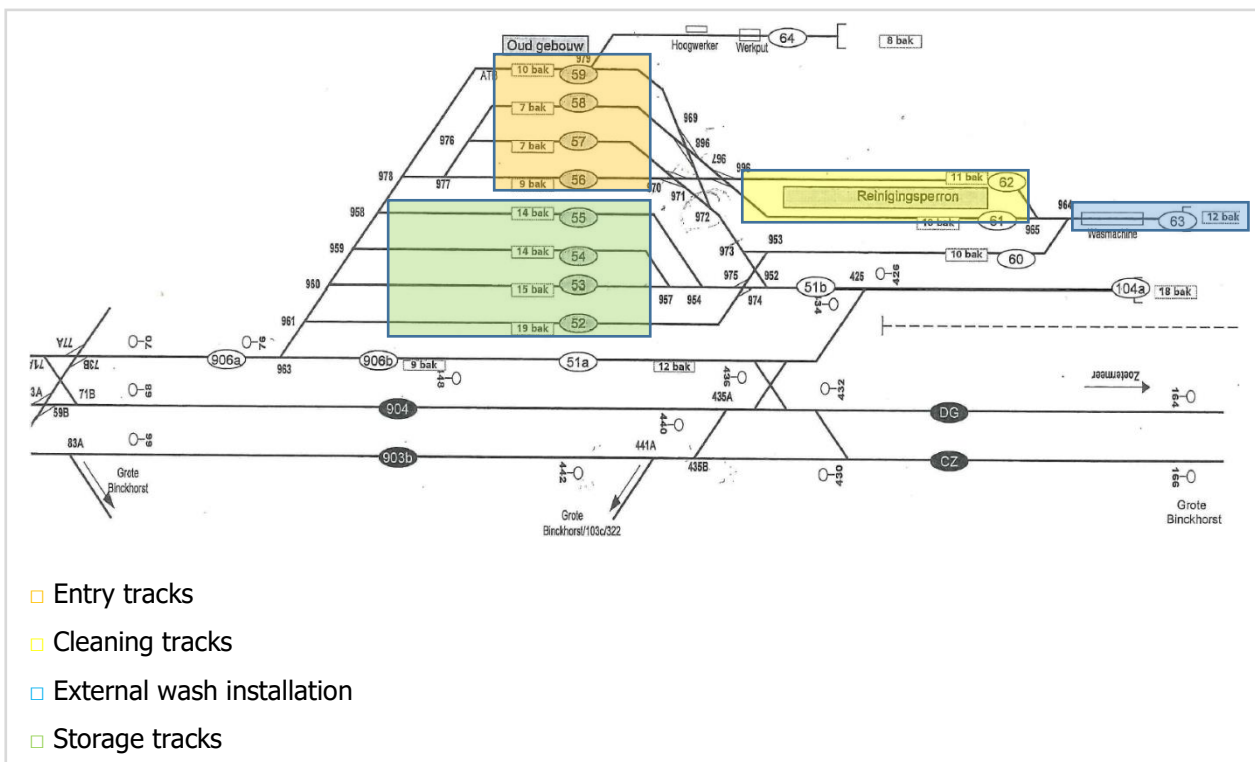


Figure 21-Service areas at Bkh.kl

5.3.4 Additional rules for parking

Some variations on the same solution method have been tested. Each variation adds some rules to the initial method. In the following part the specifics of each variation are explained.

Type Based Strategy

Trains are only allowed to park behind a train of the same type and length. If no such option is available, the train is parked on the first empty track. If no storage track is empty, the train will park in random order at an entry track.

In Residence Time Strategy

Trains prefer to park behind a train with a lower priority, but as close to its own priority as possible. If there exists a track with trains of lower priority that will be filled to maximum capacity by this particular train, it will be preferred over all other options. Trainsets of the same leaving train are allowed to park behind each other, even when they are not in the right order.

IRTS-LS

In the initial setup an effort is made to minimize the number of splits. If an arriving train should be split to match the consist of the leaving train, the matching procedure is continued to check if any other match could be made that does not need a split action.

IRTS

In the basic setup the same rules apply as mentioned above, except the train selection procedure is finished at the first possible match to promote chronology of trains.

IRTS-CT

The same rules apply as in the basic version of IRTS, except combining of trains is promoted by blocking tracks that contain uncomplete trains. Only trainsets of the same train are allowed to park on this track until the train is complete, except if there are no other parking options possible.

IRTS-RO

The same rules apply as in previous version of IRTS, but in addition trainsets of the same train are allowed to restore the correct order by temporarily shunting a train from the storage track to track 104a. In this way the new arriving trainset can be placed in front of the earlier arrived trainset. The shunting procedure blocks the connecting tracks for an extended period of time, in order to remove the parked train, reverse direction, and move it to its original track after the new train is parked.

IRTS-LTF

The same rules apply as in previous versions of IRTS, except now the longest tracks are assigned first instead of the shortest tracks.

Additional general rules

In order to improve the usage of the tracks, the system is able to assign all trains at one track to another track with retrospective effect. The trigger for this event is when the usage of a specific track is equal

to the maximum capacity of another track. In the simulation the trains will stay in the same queue, but the capacity of both queues is switched. For instance, if track 52 (max capacity 19) contains 14 wagons, and track 55 (max capacity 14) contains less than 14 wagons, then the capacity of track 52 is set from 19 to 14 and the capacity of track 55 is set from 14 to 19. At the same time the names of both queues are switched, so the queue with name 'track 52' has a capacity of 19 again, though be it with another content.

After trains are being parked at the storage tracks, they stay standby at that location until the end of the simulation, even though they depart earlier in practice. This is a simplification resulting from the initial assumption that all trains arrive before the first leaves. This assumption can be rectified by the fact that trains generally are not relocated to the end of the track anyway, because it would take too much effort to move every single remaining train on the track whenever a train left.

5.4 Special operations

Sometimes the composition of a train should be altered to enable matching; either a train should be divided in two parts to form two shorter trains, or two trains should be put together to form one new longer train. The procedure is referred to as splitting trains, and combining trains. Both procedures will be explained in more detail.

5.4.1 Split trains

As explained before, trains can enter the yard in a composition of multiple train units. For the matching of incoming trains to departing trains, it might be necessary to change the composition of a train. The action of dividing trains in two is referred to as splitting. The procedure of splitting trains is setup as follows. Which train unit should stay in the consist and which should not, is determined during the train selection procedure. At split, the length of the existing train is changed to meet the length of the front part of the train. A new train is created of the residual length of the train. This new train will go through the same routine of prioritising and determining splits as any other train, except that it starts at the exact location where the split is executed instead of at the entrance of the yard. It could occur that the new train needs to be split again. In that case the preceding procedure is repeated, including the creation of another new train.

Based on the setting, the trains can either be split at the arrival track, or the last track before entering the storage tracks. At entry it is checked if the train fits along the cleaning platform. If not, the train is split at the arrival track anyway. In the next paragraph, only results are displayed using the setting 'split at arrival track'.

While splitting, the priority is changed in order to identify the various parts of a train. The priority has the following definition: the first digit represents the index number of exiting train it is assigned to (1x is the first train exiting). The second digit represents the location of the consist in a train. For instance:

train 11 is the front part of the first leaving train. Train 12 is the second part of the same train; when combined they will form train 10.

5.4.2 Combine trains

Leaving trains can also contain several train units. While moving around the yard, trains check if the last train unit of the train in front of it on the same track, is equal to the one it should be coupled to. If so, the rear train will set its own length to zero and the front train will set the length to the combined length of both trains. The variable of the last train unit in the train is updated. Factual the last train will still exist, but it does not have any actions or capacity claim anymore.

In case of the IRTS-RO method, a train can also be combined when the trainsets are not yet in the correct order. As described in paragraph 5.4.3, a shunting movement is executed to restore the correct order of the train before the trainsets are attached to each other.

5.5 KPI's

The results of the previously described methods will be discussed in chapter 6. In paragraph 2.7 a preview is given of possible Key Performance Indicators for comparing the results of a TUSP. Now the exact setup of the simulation is known, the most important ones will be selected in the following paragraph.

Obviously, it is most important for a method to be able to solve as many instances of the TUSP as possible, for each number of trainsets. So the number of successful solutions -or the percentage of feasible solutions- is determined to be the most important KPI.

Let's recall the first three KPI's from paragraph 2.7: 'storage capacity', 'processing times of trains' and 'occurrence of unfinished maintenance'. All three of these KPI's can be covered by 'feasible solutions'. The feasibility of an instance with a high number of trainsets will prove a high storage capacity. The in-time finishing of trains suggests that the processing time of trains is fast enough for the system to cope with the demand. The not-in-time finishing of trains is covered by the breakdown of the not-feasible solutions.

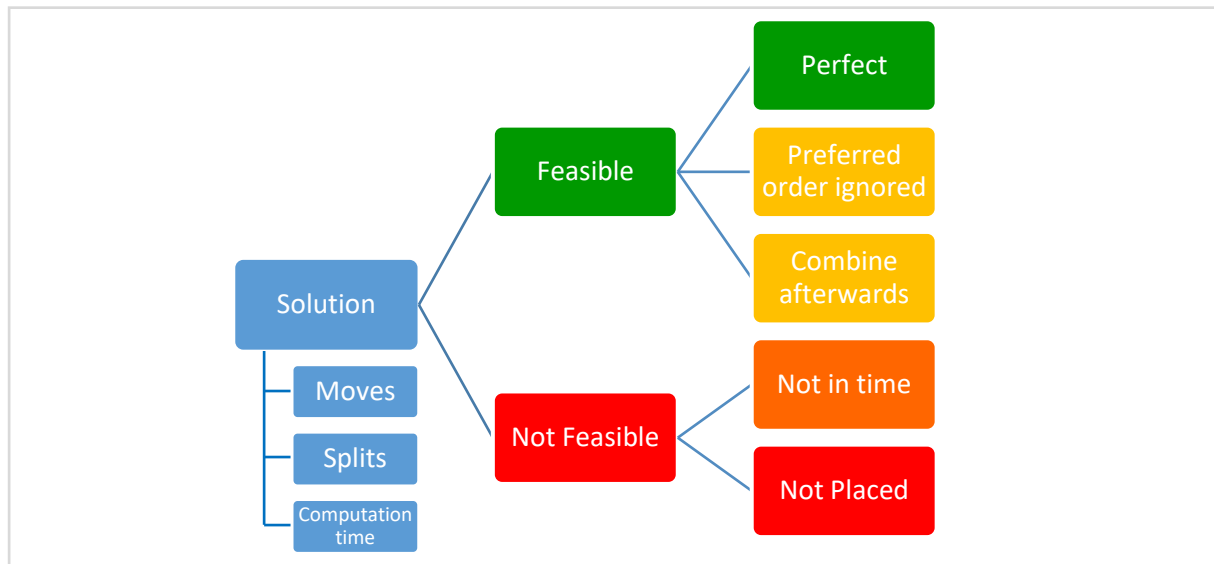


Figure 22-Breakdown of outcome of solutions

As can be seen in Figure 22, the solutions can be divided in two options: feasible and not feasible. A feasible solution can have some restrictions, depicted in the yellow boxes. These restrictions can occur multiple times per solution; the number of occasions is therefore a performance indicator of the method. The unfeasible solutions can have 2 causes for failure; both causes can occur once or more in a single solution. 'Not in time' is assumed to be less harmful than 'not placed', since there are other variables that influence the speed of processing, such as the number of available cleaners. Further motivation of the performance indicators are summarized below. At first, the performance of the method can be judged on basis of the following KPI's.

Feasible solutions	The most important KPI is the percentage of instances of a certain size that could be solved successfully, without making any further differentiation in the quality of the solution.
Calculation time	One of the intended contributions of this research is to construct a fast method of solving the extended TUSP, so the calculation should be fast.

Furthermore, the quality of the solutions can be determined based on the following KPI's. These KPI's can be counted for both feasible and unfeasible solutions.

Violation of preferred order	A train parked in the wrong order makes the solution more complex, and causes additional movements to retrieve trainsets. The solution is however feasible.
Combine at departure	A train ready to leave is more convenient than a train that needs handling before departure. It demands extra movements, needing additional train driver availability over a long time span. The extra

	movements are however less complex than with trains in the wrong order.
Moves	Less moves are preferred. Extra movements demand additional train driver availability, however over a smaller timespan than with combines afterwards.
Splits	Splitting is a time consuming activity needing a train driver, however less frequent occurring than additional moves.

As the explanation suggests, the KPI's are ordered from most important to least important. The KPI's summarized in the lists above will be used to assess the results in chapter 6.

5.6 Conclusion

A discrete process simulation model is made to assess the result of several heuristics. The model uses proposed heuristics to determine the preferred next client for each available slot, be it for entry, service of storage. On this basis, at every step in the process adds to improving the order in which trains are parked. Besides servicing, also matching including splits and combines are modelled. Input for the model will be the datasets composed by R. van den Broek [6]. The results will be compared on basis of the KPI's mentioned in this chapter.

6. Results

In previous chapter the setup of the methods is discussed. All methods are tested using the datasets provided by R. van den Broek [6]. The datasets are equal to those used in his research, and to those used by F. Wolfhagen [31], so the results of those methods could be compared to the methods in this research. In this chapter an insight is given into the results of the proposed methods and they are compared to other methods developed under supervision of NS.

6.1 Location Kleine Binckhorst (Bkh)

At first some remarks will be made with respect to each individual method. From each method is explained how they originated, and the big advantages and disadvantages are summarised. At first the most important KPI -feasibility- will be discussed. The results for this KPI are visualized in Figure 24 and Figure 25. After the comparison on feasibility, the results based on other KPI's will be discussed. These results are summarized in Figure 26.

6.1.1 Results per strategy

Type Based Strategy

The type based method results in a very robust solution, if a feasible solution is possible. At any time a trainset of a specific type is parked in front to compose a new train. Most trains are however not in the right composition to leave, and have to be combined at a fairly late stage in the process. Because of the setup, it might be possible that there is no space available to combine the trains. Another disadvantage is that the capacity is not used very efficiently. As is visualised in Figure 23, some tracks will never be used to maximum capacity, since the remaining space is not enough to accommodate another trainset of the same type. It is therefore not able to solve large instances. This method will not be very efficient if there is a high variability in train types and a restricted number of tracks.

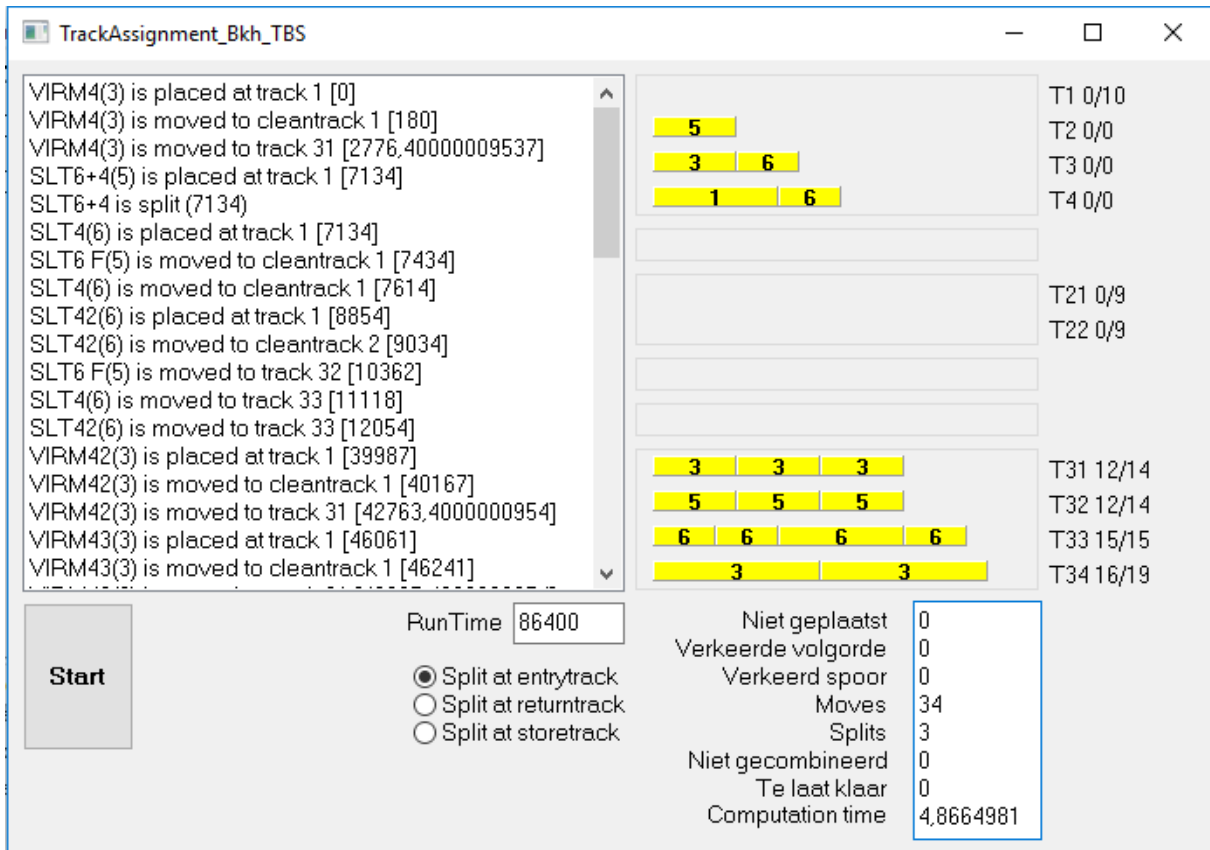


Figure 23 - Example of simulation result (TBS test number 20-5)

IRTS-LS

The first setup of the In-Residence Time Strategy obeys the strive for limiting the number of unnecessary splits. If there exists a leaving train with the same composition as an arriving train, it is matched even if it is the first train arriving and the last train leaving. A drawback of this rule is that it might be necessary to assign a part of an early arriving train to an early leaving train, in order to have that train ready to leave in time for departure. In many cases a second trainset of the same type has not even arrived in the yard when the first train should leave. As a result, many trains in the IRTS-LS setup are late. In order to solve this problem, the second setup is created

IRTS

In contrast to the previous setup, now trains are matched if the first trainset of a train matches a part of a leaving train. As a result, some more splits and moves are needed to process all trains in the yard. The number of instances of a train being late drops however from 85 to 35. The number of trains leaving in the 120 instances is equal to 1130, so this means a reduction of late finishes drop from 7.5% to 3.1%.

During the simulations of this strategy a new inefficiency is remarked. In quite some instances, a part of a leaving train becomes blocked by other trains, before the consist is complete. Although the order of trains per track is correct, and thus all trainsets of a train are unblocked before they have to leave,

it might be preferred to complete the trains in an earlier stadium. For this purpose, the third method is proposed.

IRTS-CT

In this method a track is blocked for other trains until a train is completed. The feasibility for the datasets up to 24 trainsets is equal to previous setup. The difference in number of feasible solutions is in de simulations with 26 trains. Figure 26 shows however that while more trains are combined at an early stadium, also more trains are forced to park in a wrong order. This could be explained by the fact that the higher percentage of long trains reduces the likeliness of a track having enough capacity for the entire consist. Besides, the long consists reduce the chance of a track being filled to maximum capacity, because less variations are possible.

IRTS-RO

In all three previously described setups it occurs that a second part of a train is cleaned and parked before the first part of the same train. As a result, these trains do not end up in the yard in the right order, and cannot be combined in the right order until other trains left the yard. There is however a track (track 104 in Figure 21) that enables a shunting movement to restore the correct order of parking. This ability is used in the fourth setup of the method.

The method is able to solve 1 instance less than the IRTS-CT variant. It does however succeed in its intention; the number of trainsets combined in an early stadium increases significantly. On the downside, just like the IRTS-CT setup there is a higher percentage of long trains stored on the tracks, resulting in less flexibility and less efficient track use. The reduced flexibility leads to more trains that cannot be stored in the right order.

IRTS-LTF

The intention of the last setup is to increase the flexibility. By allowing the trains to park on the longest tracks first, the theory is that there is a higher chance of the longest track containing the maximum capacity of a shorter track. Due to the possibility of switching the content of tracks, this should increase the number of tracks filled to maximum capacity.

The method performs quite different when compared to the others setups. In two sets the method performs worse, but in two other sets it performs better, including two sets with 24 trains in which no setup had found a solution yet.

The setup achieves another reduction in the number of splits at a late stadium, but just like in previous occasions it is at cost of the number of trains that are parked in the right order, as can be seen in Figure 26.

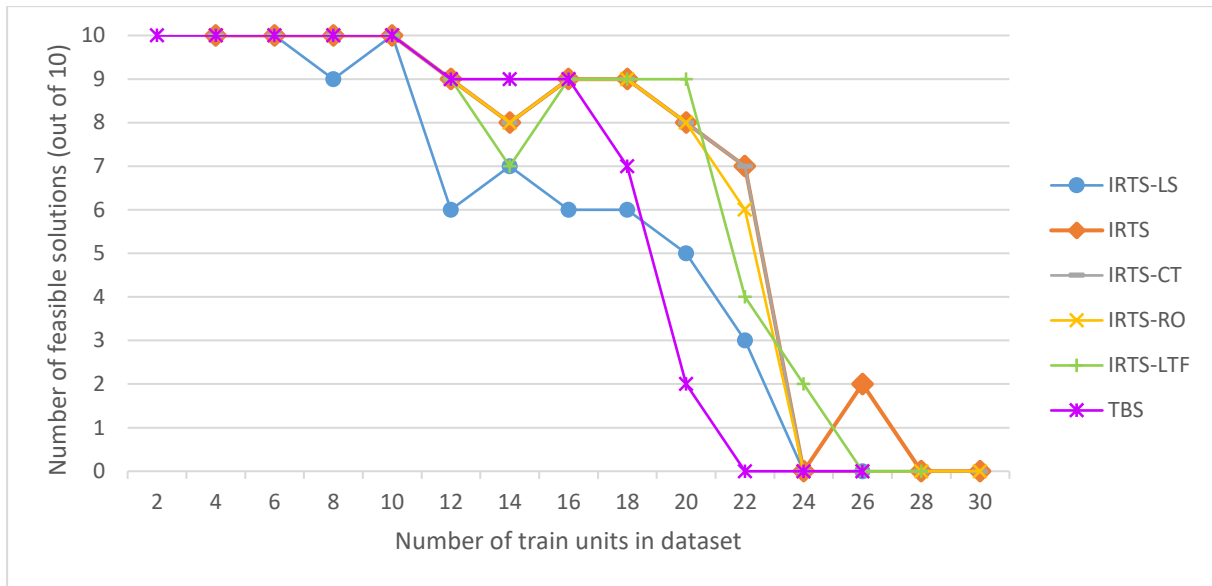


Figure 24-Feasibility of the proposed methods per dataset

6.1.2 Comparison of proposed methods

Now we have reviewed the individual results of the proposed methods, they can be put together and weighed against one another. The weighing can be done based on different factors. First we compare the methods purely on feasibility. An overview is given in Figure 25.

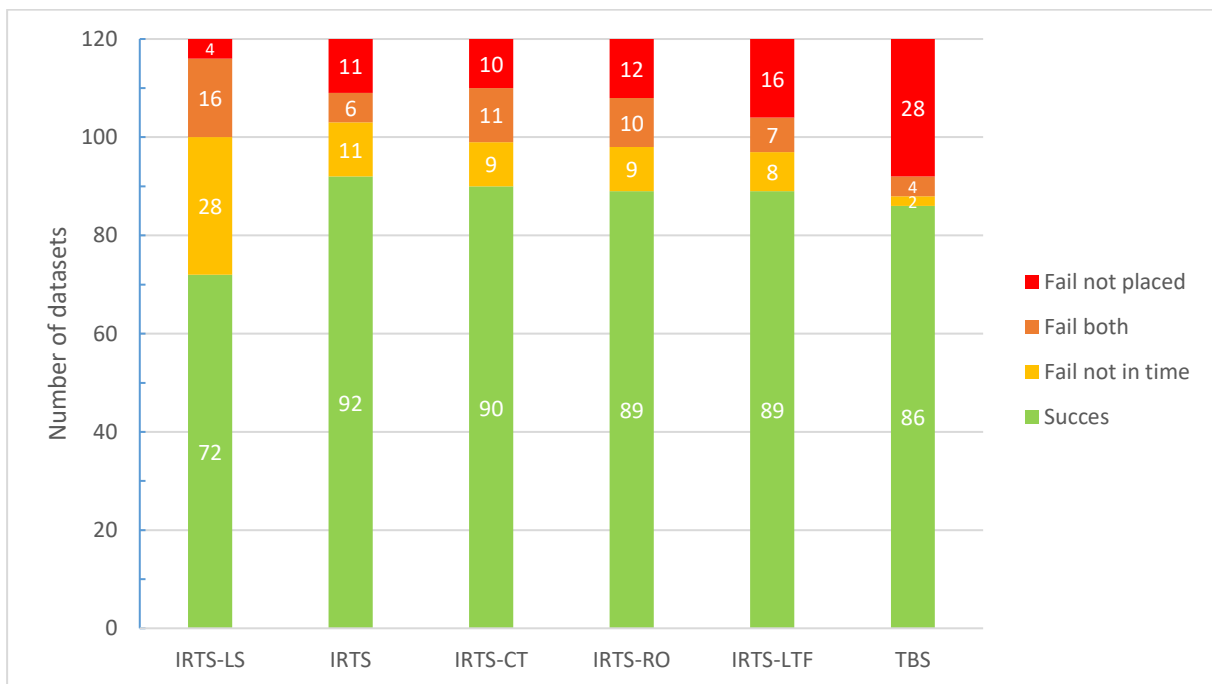


Figure 25-Causes of failure at datasets with 4 to 26 trainsets

From Figure 24 it could already be derived that the basic IRTS method is performing above average at nearly all datasets. Figure 25 confirms this observation, since the IRTS method has the most successful solutions in the 120 runs. A failure due to not placing a train could be regarded more harmful than failure due to late finishing, since the speed of the processing of trains could be increased by other

relatively simple factors, such as increasing the number of cleaners. Figure 24 shows that the basic IRTS setup results in the least amount of failures due to lack of useable space, namely 17.

The proposed methods can also be evaluated based on other factors, as is summarized in Figure 26. The figure illustrates the performance of 5 factors scaled from 0% errors to 100%, where 100% corresponds to the maximum number of instances that a factor occurs. For all factors applies: the closer to 0, the better the result. The most important factor is of course the feasibility. Recall from paragraph 5.8 the ranking of the other:

- 1) NOT feasible solutions
- 2) Violation of preferred order
- 3) Combine at departure
- 4) Moves
- 5) Splits

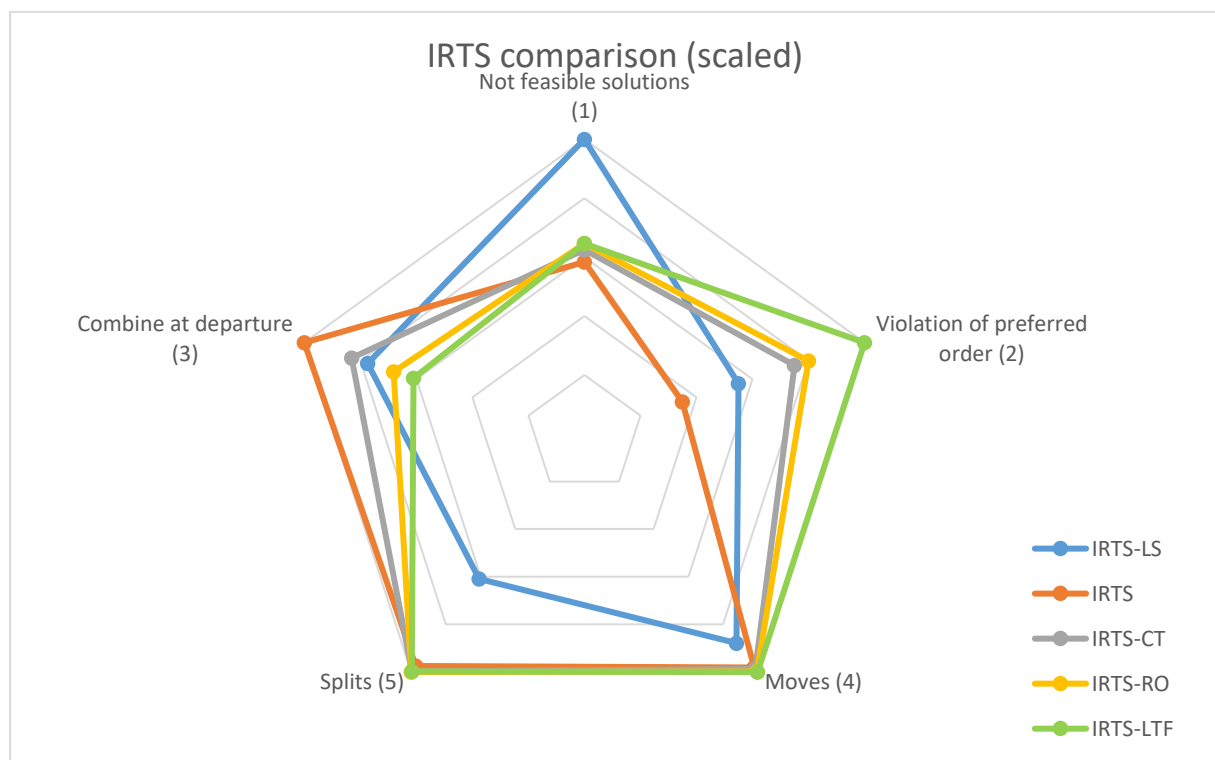


Figure 26-Comparison of the IRTS setups, scaled from 0 to the worst performing score

The IRTS-LS method performs well on the least important factors, but terrible at the most important one. IRTS performs best on all factors, except the third important factor. Each attempt to improve the third factor, results in a significant downturn of the more important factors. On this basis, the IRTS method proves to be the best performing method. In future it might be interesting to apply the reversing order changes to the basic IRTS setup, since it significantly improves the performance on third factor while only slightly reducing the performance on the second factor, but this will be for further research.

The results of Figure 24, Figure 25 and Figure 26 are a visualisation of the data in Table 7.

	Type Based Strategy	Limited Split (match trainset with longest possible leaving train)	(match trainset with first leaving train)	Combine Trains (Block track if last train is not complete)	Reverse Order (Shunt last train in order to improve order)	Longest Track First (Assign trains to longest track)
	TBS	IRTS-LS	IRTS	IRTS-CT	IRTS-RO	IRTS-LTF
Instance succesfull (out of 120)	86	72	92	90	89	89
Instance not feasible [1]	34	48	28	30	31	31
Violation of preferred order [2]	0	11	7	15	16	20
Combine at departure [3]	-	284	367	305	250	224
Moves [4]	2870	2824	3153	3173	3206	3215
Splits [5]	248	232	371	380	381	380
Calculation time [s] (avg)	2,32	3,15	2,82	2,93	2,99	2,83
Calculation time [s] (max)	4,39	5,87	5,37	5,43	5,45	6,39
Trains not placed	59	37	22	36	37	33
Trains not in time	8	85	35	40	37	33
Instance fail not placed	28	4	11	10	12	16
Instance fail both	4	16	6	11	10	7
Instance fail not in time	2	28	11	9	9	8
4 trainsets – succes (out of 10)	10	10	10	10	10	10
6 trainsets – succes (out of 10)	10	10	10	10	10	10
8 trainsets – succes (out of 10)	10	9	10	10	10	10
10 trainsets – succes (out of 10)	10	10	10	10	10	10
12 trainsets – succes (out of 10)	10	6	9	9	9	9
14 trainsets – succes (out of 10)	9	7	8	8	8	7
16 trainsets – succes (out of 10)	9	6	9	9	9	9
18 trainsets – succes (out of 10)	9	6	9	9	9	9
20 trainsets – succes (out of 10)	7	5	8	8	8	9
22 trainsets – succes (out of 10)	2	3	7	7	6	4
24 trainsets – succes (out of 10)	0	0	0	0	0	2
26 trainsets – succes (out of 10)	0	2	0	0	0	2

Table 7-Bkh, cumulative results for 120 datasets (10 instances for 4,6,8,...,24,26 trainsets)

6.1.3 Comparison with other methods

In the following paragraph we will compare the best performing proposed method (IRTS) to the other methods developed under supervision of NS. All methods in this paragraph are tested using the exact same datasets, and are therefore perfectly comparable.

One major difference between the methods becomes clear in the results visualized in Figure 29. In IRTS the claimed capacity at a track is not returned after departure of the train. The maximum number of wagons during a day can therefore not exceed the total capacity of the yard. Simulated Annealing

obviously does re-assign capacity, since the total number of wagons in the dataset with 30 trainsets and beyond exceeds the total length of the tracks in the yard, as is visualized in Figure 27.

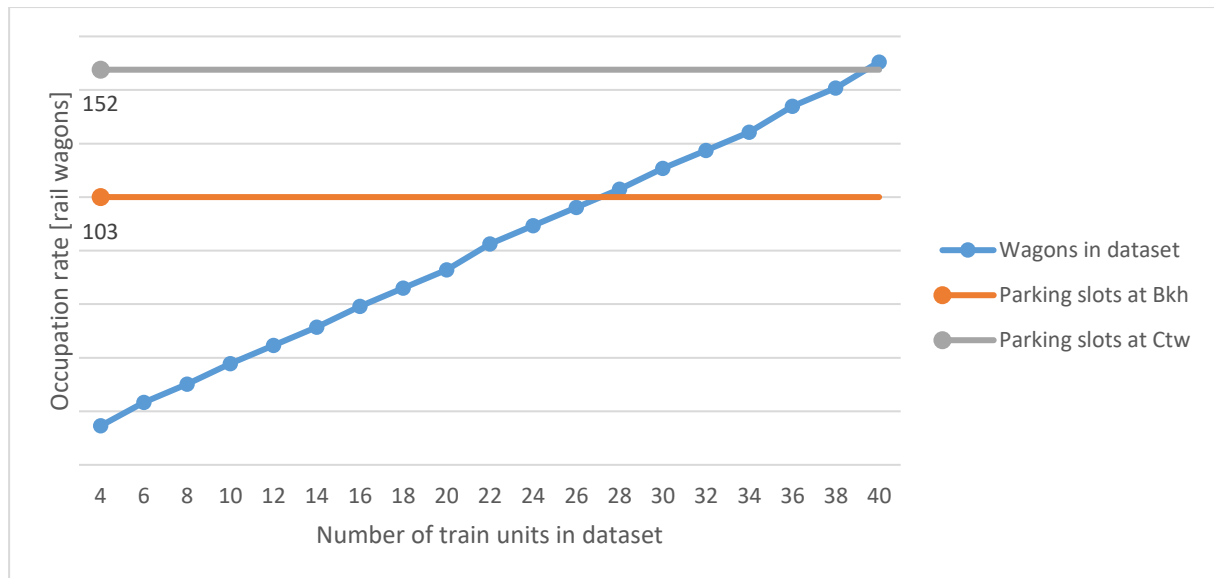


Figure 27-Theoretical capacity vs. input

Apart from the limitation of not releasing capacity, IRTS does always keep one track clear for the arrival of new trains. The capacity for parking processed trains is therefore even lower, namely 103 wagons. Due to the constraints it is thus theoretically impossible for the IRTS method to find solutions for the datasets with more than 26 trainsets.

For a comparison a new simulation has been built in which trains do return capacity after they leave, but only if no trains are parked behind it; or to translate it to reality: trains are not moved when parked, because it would result in extra movements. Some datasets have been tested again in the new setup, proving that instances of up to 28 are possible in most occasions. Only few instances of 30 train units were feasible.

If all trains would be moved to the end of the track when a train in front leaves, the number of moves increases by 36 for an instance of 30 trains, resulting in 1.6 times the number of moves needed for the initial problem. It does however not lead to the feasibility of any of the bigger instances since a new bottleneck arises: the time needed for cleaning and shunting creates a backlog for the cleaning platform, causing shortage of entry space or trains being late. Increasing the number of cleaners proves not to solve this problem either. This illustrates the complexity of the entire logistical system: changing one parameter will not always lead to better results, since a new problem will arise at some point.

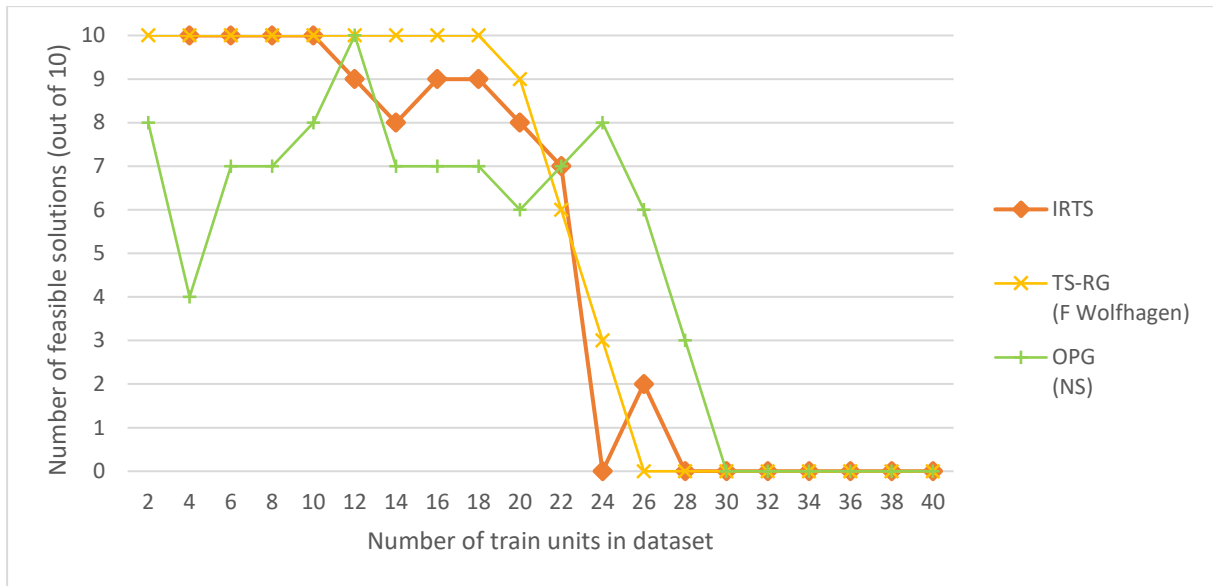


Figure 28-Comparison between proposed method and various other methods (Bkh.kl)

Figure 28 shows that IRTS is able to find a solution in the same range as TS-RG, developed by F. Wolfhagen [31]. TS-RG is able to successfully solve 6 instances more than IRTS, mainly in datasets containing less trainsets. Although the TS-RG is much more sophisticated and takes a significant amount of time to compute, up to 1800 seconds per solution compared to 6 seconds with IRTS, it is not performing significantly better.

OPG, one of the earlier developed tools by NS, is able to successfully solve 7 instances less than IRTS. OPG has a higher uncertainty of solving successfully the datasets with less trainsets, but it is able to solve some of the instances with more trainsets than IRTS.

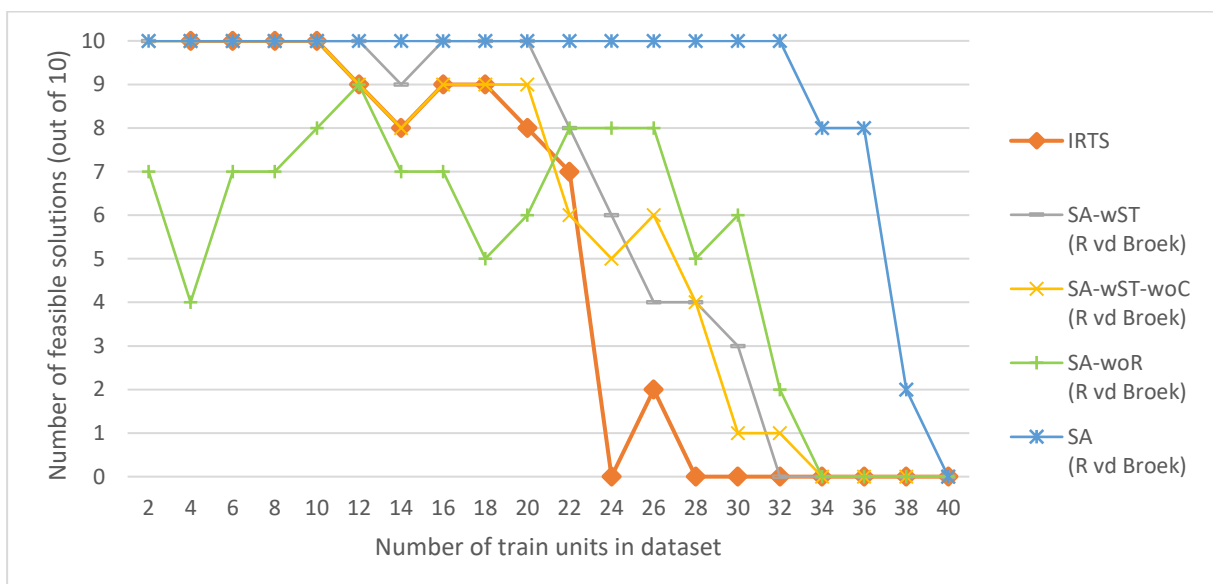


Figure 29-Comparison between proposed method and R vd Broek (Bkh.kl)

R. van den Broek [6] developed various solutions based on Simulated Annealing. The results of the four variations are depicted in Figure 29. The first two methods include service tasks, namely: internal

cleaning, external cleaning and safety checks. The second method excludes the safety checks and only considers cleaning tasks. The last two methods exclude all service tasks and the last variation introduces repositioning of trains after parking. The Simulated Annealing solution has a perfect result in instances with up to 34 trainsets, calculated in approximately under 60 seconds. SA uses extra repositioning and splitting of trains to fit trains along the remaining space, which demand additional labour. It does also not include any service task, in contrast to all other methods. IRTS performs not as well as any SA method, however when capacity at tracks is returned at departure of a train, IRTS nears the result of SA-with-Service-Tasks. When comparing the computation times of all solution methods as depicted in Table 8, IRTS proves to be significantly faster than all other methods that include service tasks.

	<20 trainsets	<28 trainsets	>26 trainsets
IRTS	3,9s	4,5s	5,9s
TSRG	200s	> 1000s	
OPG	40s	> 300s	
SA-wST	10s	70s	200s
SA-wST-woC	5s	50s	110s
SA-woR	4s	7s	15s
SA	5s	13s	110s

Table 8-Calculation time of various methods for Bkh

IRTS is able to solve most instances of up to 20 and 22 trainsets, corresponding to 75 and 82 wagons. If we recall Figure 11, we can see the maximum expected number of rail cars currently at Bkh.kl equals 70, so we can conclude that the method is able to solve realistic size instances. It has to be remarked however that the variety in train types does not correspond to reality. Also, external cleaning and technical checks are not taken into account in the performed simulation.

6.2 Location Cartesiusweg (Ctw)

In this paragraph the second service site will be reviewed. The same datasets will be used as in previous paragraph. There is no data available from other NS studies for this service site, so the results will be compared with the results of previous paragraph, in which a comparable method was applied to another service site. The theoretical maximum capacity is 152 rail cars, which means a solution with 38 train units should be possible as is visualised in Figure 27.

6.2.1 Results per strategy

IRTS

In the first runs of the Cartesiusweg simulation, it is not allowed to park in front of a train that is scheduled to leave earlier. At the same time, arriving trains are matched to the first leaving train, so early leaving trains are parked at the rear of the tracks. As a result of this, the method is only able to solve instances of up to 34 trainsets and the number of feasible solutions drop dramatically at 30 trainsets. Most instances fail because of lack of useable space. At this stage there are many unused slots in the yard, but they are blocked by earlier leaving trains at the rear of the track.

IRTS WO

In this version of the method a train is allowed to park in front of a train that is scheduled to leave early in case there are no other options. In short: a train is allowed to violate the preferred parking order if necessary. This policy results in some additional shunting for retrieving the rear trainsets, but enables to use the available space to full capacity. The method is able to find a storage location in 83% of the instances. In the runs with more trainsets it happens however regularly that all trains can be assigned a slot, but not all trains are finished servicing in time.

IRTS 6CL

The problem of late finishing might well be caused by the absence of enough cleaners. Previous 2 methods used the same number of cleaners that are commonly available at an average day at the Bkh.kl. The Ctw yard is however significantly larger, with its theoretical capacity of 152 compared to the 103 of Bkh.kl. To compensate for the bigger number of trains that can be accommodated, the number of cleaners in the next runs is increased from 5 to 6. The result is clearly visible in Figure 30, since the number of successful results are increased significantly. The addition of a seventh cleaner only results in just one additional successful run. Because of the minor improvement, the result of 7 cleaners is not visualized in Figure 30, but the additional successful result is in the dataset with 36 trainsets.



Figure 30-Feasibility of the proposed methods per dataset (Ctw)

6.2.2 Comparison of proposed methods

Using Figure 31 the feasibility of the various versions of the IRTS method can be compared. The graph of IRTS Ctw shows 41 failed instances due to trains that are not placed. By allowing to park in the wrong order, the number of infeasible solutions caused by lack of usable capacity drops to 8.

Remarkably, the number of not placed trains increase when the number of cleaners is increased from 5 to 6. The earlier availability of finished trains leads to small changes in the timing of track capacity

switches, causing 2 instances in which a train was late in the Ctw WO method to change towards a situation in which all trains are in time, but not enough space is left.



Figure 31-Causes of failure at measured datasets

Just like in paragraph 6.1 the results are summarized in a spider chart (Figure 32). The results of 2,3,4 and 5 are linear scaled from 0% to 100%, in which the worst performing method represents 100%. In order to value the performance of both locations despite of the different sizes of the yards, the feasibility is scaled by the number of datasets that theoretically fit the physical maximum capacity. 'Not feasible' is determined by:

$$NF_{percentage} = (n_{worst\ performing} / NF_{worst\ performing}) * (NF / n) \quad (6.1)$$

In which NF is the number of not feasible solutions in the method, n the number of instances tested with the method, $n_{worst\ performing}$ the number of instances tested by the worst performing method, and $NF_{worst\ performing}$ the maximum of not feasible solutions in the worst performing method.

Based on the spider chart one can conclude that the strive for better performance on one criterium leads to a decrease of the performance of at least one other criterium. In this simulation three of the criteria stay relatively constant. Reducing the number of infeasible solutions has mainly effect on the number of trains that are parked in the wrong order. The number of additional moves for the recovery of the rear trainset is not taken into account in this simulation setup, but this would cause an increase of moves of approximately 10%.

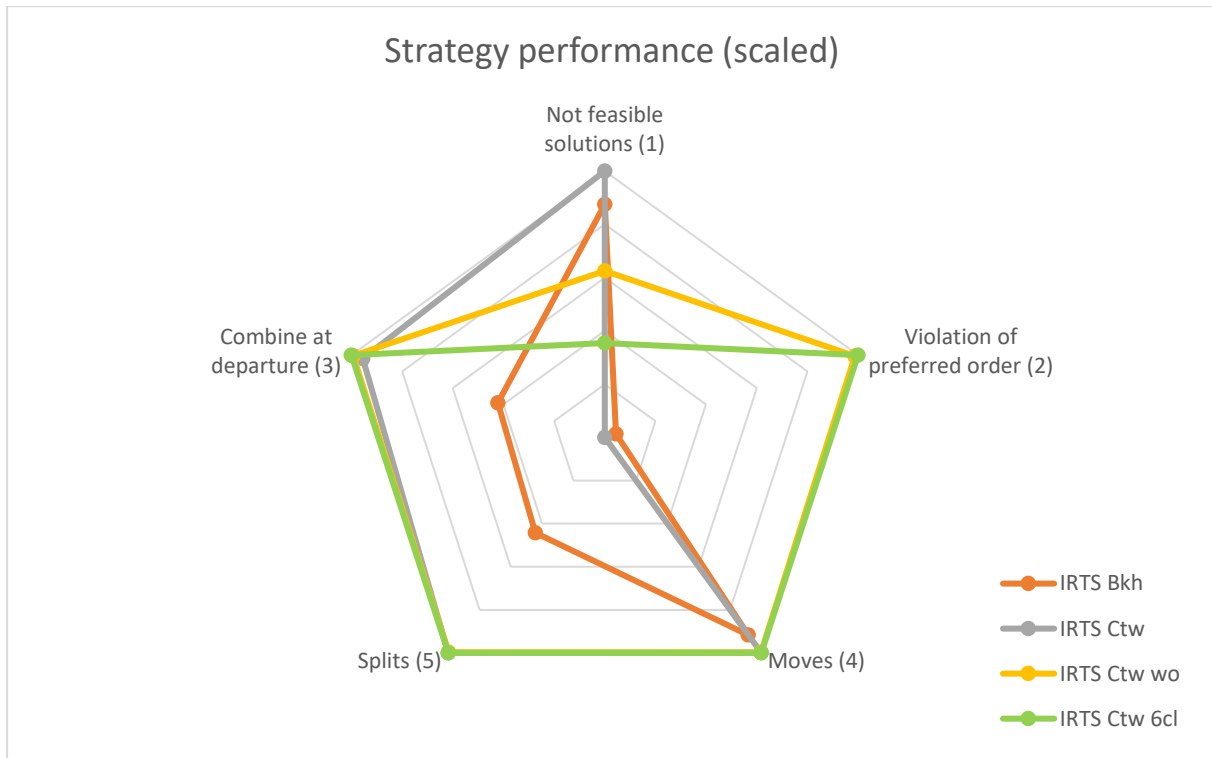


Figure 32-Comparison of the IRTS setups, scaled from 0 to the worst performing score

Just like remarked in previous paragraph, the type of trains in the dataset are not representative for a real-world scenario. Unlike the Bkh yard, this might have a significant influence on the performance of the method. The dataset contains quite some large trains, and has little variation in train length containing only trainsets of 3,4 and 6 wagons. Ctw contains many tracks of limited length; for instance, there are 6 tracks with a length of 5 wagons. In reality these can be occupied by various train types, such as DDM1 and SGM in order to use the full length of the track. With the train types in the datasets however there will be at least 1 slot empty per track, resulting in the loss of at least 6 usable slots in total just because of the selected input of the model. Using a dataset with higher variety of train types, or even a dataset with just train types of other lengths, might therefore improve the occupation rate of the yard and thereby possibly improve the results of the method. Figure 30, Figure 31 and Figure 32 are based on the data summarized in Table 9.

	IRTS Bkh	IRTS Ctw	IRTS Ctw wo	IRTS Ctw 6cl
Instance succesfull (out of 120/180)	92	132	150	163
Instance not feasible [1]	28	48	30	17
Violation of preferred order [2]	7	0	157	159
Combine at departure [3]	367	831	864	870
Moves [4]	3153	3434	3430	3438
Splits [5]	371	837	834	837
Calculation time (avg)	3,15	2,60	2,53	2,54
Calculation time (max)	5,87	5,00	5,81	5,43
Trains not placed	22	163	11	13
Trains not in time	35	68	64	20
Instance fail not placed	11	20	4	6
Instance fail both	6	21	4	4
Instance fail not in time	11	7	23	8
4 trainsets – succes (out of 10)	10	10	10	10
6 trainsets – succes (out of 10)	10	10	10	10
8 trainsets – succes (out of 10)	10	10	10	10
10 trainsets – succes (out of 10)	10	10	10	10
12 trainsets – succes (out of 10)	9	10	10	10
14 trainsets – succes (out of 10)	8	9	9	9
16 trainsets – succes (out of 10)	9	10	10	10
18 trainsets – succes (out of 10)	9	10	10	10
20 trainsets – succes (out of 10)	8	10	10	10
22 trainsets – succes (out of 10)	7	10	10	10
24 trainsets – succes (out of 10)	0	9	9	9
26 trainsets – succes (out of 10)	2	9	9	9
28 trainsets – succes (out of 10)	0	8	8	8
30 trainsets – succes (out of 10)		0	10	10
32 trainsets – succes (out of 10)			5	9
34 trainsets – succes (out of 10)			4	9
36 trainsets – succes (out of 10)			4	8
38 trainsets – succes (out of 10)			2	2

Table 9-Ctw, cumulative results for 180 datasets (10 instances for 4,6,8,...,36,38 trainsets)

6.3 Conclusion

The basic version of IRTS proves to be the best performing method of the various proposed methods. When compared to other methods developed at NS, IRTS performs nearly as well as TS-RG, or at least it is able to solve nearly as many cases, but with much less computation time. IRTS performs more consistent than the earlier versions of OPG and is able to find more solutions in total, but OPG is able to find solutions for some datasets with a larger amount of trains. SA outperforms IRTS in a reasonable amount of time, but mainly because of using actions that require a lot of additional labour in practice or omitting service tasks. In addition, IRTS could perform better if capacity was returned after departure of a train.

In a yard with a less complicated setup such as Ctw, the IRTS method is able to find an even higher percentage of solutions. Over 90% of all the cases that are theoretically possible are solved by the IRTS Ctw 6cl method. By analysing 2 different locations of different sizes and setups, an insight is given in the effect of different factors. In general, the more possible combinations of trains and tracks, the better the performance. Dead end tracks with build in cleaning platforms makes the system significantly less complex. Dead end tracks are however not beneficial for maintaining the prescribed sequence when the yard nears its maximum capacity, since early leaving trains and late arriving trains tend to block each other.

7. Conclusion and recommendations

In this thesis various strategies have been proposed to solve the Train Unit Shunting Problem (TUSP). The strategies are based on methods used in the container industry to stack containers. The exact arrival and departure time of containers is nearly impossible to predict, so the methods use specific characteristics for categorization of arriving containers, and form logical piles based on this categorisation. This method is translated to the TUSP, resulting in two policies: the 'Type Based Strategy' (TBS) and the 'In Residence Time Strategy' (IRTS).

Both TBS and IRTS use simple decision rules, based on the current state of the yard. It does not take future events into account and it will therefore not be able to outperform most sophisticated mathematical methods. A downside of the proposed methods is the lack of flexibility. For each instance, the solution method is successful or not, and it is not able to further optimize a result. Nevertheless, IRTS approaches the number of successful solved instances of various other methods that include service activities, but in a fraction of the calculation time. Besides the performance of the method, it gives a good representation of what is possible in a real service yard, even when uncertainty of future events is high. This might be the biggest advantage of IRTS: it generates a result that is robust, even in a chaotic system. It is able to cope with changing circumstances, which might be useful in one of the world's busiest rail networks. So above all the solution method illustrates a scenario that is easy to implement and execute in real world.

By analysing two different locations, of different sizes and setups, an insight is given in the effect of different factors. In general, the more possible combinations of trains and tracks, the better the performance. Dead end tracks with build in cleaning platforms makes the system significantly less complex. Dead end tracks are however not beneficial for maintaining the prescribed sequence when the yard nears its maximum capacity, since early leaving trains and late arriving trains tend to block each other.

Overall can be concluded that IRTS is a fairly simple method that performs remarkably well compared to more sophisticated but time consuming alternatives. These results suggest that IRTS offers an easy to implement opportunity to improve train yard logistics that are currently manually performed. The key features of these algorithms allow operators to react swiftly and flexibly to the hectic and ever changing environment of train yards.

7.1 Future research

As mentioned before, each solution method for itself has little flexibility. At the same time, each variation creates new possibilities, but regularly with a negative effect on other aspects of the solution. It might be beneficial to apply the artificial instances to multiple solution methods, and select the best result. In this way one would be able to create more optimised results, without a reduction in the number of feasible solutions.

Yards are not by definition empty at the start of the day, or even at any moment of a day, so the assumption of an empty yard is not always realistic. It might therefore be interesting to create a simulation that is able to cope with initial filling of the yard. The policy is perfectly able to solve problems with trains already present in the yard, but the influence is not tested yet. The addition of initial filling also makes it possible to rerun the problem when the input changes due to disturbances in the network, which is earlier mentioned to be a potential benefit of this policy.

Another remark that was made during this research, is the small variation in the datasets. Only a few train types are represented in the artificial instances, which makes the solution a lot less complex, but also a little less realistic. It might be interesting to construct datasets with other types of trains or more types. The higher variety of types might change the performance of the various methods, since the matching of trains leaves less room for optimization. At the same time, the higher variety of train lengths, especially the addition of smaller train units, will probably increase the possibilities of filling a track to full capacity.

The last suggestion for further research also deals with the content of the dataset. In this research only a limited amount of service activities is tested. Especially the addition of external cleaning might be interesting, since it is a time consuming activity using a scarce resource. The addition of external cleaning and other activities such as safety checks could be implemented using the same assignment rules as proposed in this research. It might reveal new bottlenecks, in the same way the cleaners at the cleaning platform proved to be a bottleneck during this research. This issue shows again that the TUSP is a complex system with a lot of variables, all of which are tied together. By using simple assignment rules and incorporate known logistical processes, as is done in this research, can easily give an impression of the effect in a real world system.

References

1. J. M. van den Akker, H. Baarsma, J. Hurink, M. Modelski, J. J. Paulus, I. Reijnen, D. Roozemon, and J. Schreuder. Shunting passenger trains: getting ready for departure. Technical report, Universiteit Utrecht, 2008.
2. A. Alessandri, S. Sacone, S. Siri, Modelling and optimal receding-horizon control of maritime container terminals, *Journal of Mathematical Modelling and Algorithms* (2-2007)
3. J. F. Álvarez, A heuristic for vessel planning in a reach stacker terminal, *Journal of Maritime Research* Vol. III No. 1 (2006) 3-16
4. U. Blasum, M. R. Bussieck, W. Hochstättler, C. Moll, H.-H. Scheel, and T. Winter. Scheduling trams in the morning. *Mathematical Methods of Operations Research*, 49(1):137-148, 1999.
5. N. Boysen, M. Fliedner, F. Jaehn, and E. Pesch. Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, 220 (1):1-14, 2012.
6. R. van den Broek, Train Shunting and Service Scheduling: an integrated local search approach, Master's Thesis, Department of information and computing science, Utrecht University, 2016.
7. M. Caserta, S. Schwarze, S. Voß, A mathematical formulation and complexity considerations for the blocks relocation problem, *European Journal of Operational Research* 219 (2012) 96–104
8. S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Recoverable robustness for train shunting problems. *Algorithmic Operations Research*, 4(2): 102-116, 2009.
9. R. Dekker, P. Voogd, E. van Asperen, Advanced methods for container stacking, *OR Spectrum* 28 (2006) 563–586
10. G. Di Stefano, M. Love Koči, A graph theoretical approach to the shunting problem, *Electronic Notes in Theoretical Computer Science* 92 (2004) 16–33
11. M. T. van Dommelen. Scheduling train service activities at service sites to determine the capacity. Master's thesis, School of Econometrics and Operation Research, Vrije Universiteit Amsterdam, 2015.
12. M. Duinkerken, J. Evers, J. A. Ottjes, A simulation model for integrating quay transport and stacking policies on automated container terminals, *Proceedings of the 15th European Simulation Multiconference* (2001)
13. C. Eggermont, C. A. J. Hurkens, M. Modelski, and G. J. Woeginger. The hardness of train rearrangements. *Operations Research Letters*, 37(2):80-82, 2009.

14. R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 186(2):261-272, 2005.
15. M. Gatto, J. Maue, M. Mihalak, and P. Widmayer. Shunting for dummies: An introductory algorithmic survey. In *Robust and online large-scale optimization*, pages 310-337. Springer, 2009.
16. J. T. Haahr, R. M. Lusby, J. C. Wagenaar, Optimization methods for the Train Unit Shunting Problem
17. Haijema, R. Train shunting: A practical heuristic inspired by dynamic Programming, University of Amsterdam. Diss. Master Thesis in operational Research, 2001.
18. P. M. Jacobsen and D. Pisinger. Train shunting at a workshop area. *Flexible services and manufacturing journal*, 23(2):156-180, 2011.
19. D. Jekkers. Train shunt planning using genetic algorithms. Master's thesis, School of Economics, Erasmus University Rotterdam, 2009.
20. K. H. Kim, G.-P. Hong, A heuristic rule for relocating blocks, *Computers & Operations Research* 33 (2006) 940–954
21. E. Kozan, P Preston, Mathematical modelling of container transfers and storage locations at seaport terminals, *OR Spectrum* 28 (2006) 519–537
22. L. G. Kroon, R. M. Lentink, and A. Schrijver. Shunting of passenger train units: an integrated approach. *Transportation Science*, 42(4):436-449, 2006.
23. Y. Lee, N.-Y. Hsu, An optimization model for the container pre-marshalling problem, *Computers & Operations Research* 34 (2007) 3295 – 3313
24. Y. Lee, Y.-J. Lee, A heuristic for retrieving containers from a yard, *Computers & Operations Research* 37 (2010) 1139–1147
25. R. M. Lentink. Algorithmic Decision Support for Shunt Planning. PhD thesis, School of Economics, Erasmus University Rotterdam, 2006.
26. R. M. Lentink, P.-J. Fioole, L. G. Kroon, and C. van't Woudt, Applying operations research techniques to planning of train shunting. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*, pages 415-436, 2008.
27. D. Sculli, C.F. Hui, Three dimensional stacking of containers, *OMEGA Int. J. of Msmt Sci.*, Vol. 16. No. 6 (1988) 585-594
28. D. Steenken, S. Voß, R. Stahlbock, Container terminal operations research – a classification and literature view, *OR Spectrum* 26 (2004) 3–49
29. N. Tomii, Li J. Zhou, Depot shunting scheduling using combined genetic algorithm and PERT, *Computers in Railways VII* (2000) 437-446

30. T. Winter and U. T. Zimmermann, Real-time dispatch of trams in storage yards. *Annals of Operations Research*, 96(1-4):287-315, 2000.
31. F. Wolfhagen, The Train Unit Shunting Problem with Reallocation, Master's Thesis, Department of Econometrics & Management Science, Erasmus University Rotterdam, 2017.
32. J. Xin, Control and Coordination for Automated Container Terminals, PhD Thesis TU Delft, 2015

Appendix A

Scientific paper

OPTIMIZING TRAIN PARKING AND SHUNTING AT NS SERVICE YARDS

E. Beerthuizen, ir. M.B. Duinkerken, ir. B. Huisman, prof. dr. R.R. Negenborn

Delft University of Technology
Department of Transport Infrastructure and Logistics, Delft, The Netherlands
NS Techniek, Utrecht, The Netherlands
e.beerthuizen-1@student.tudelft.nl

ABSTRACT

This paper proposes several methods to solve the planning of parking and servicing of trains in a shunting yard. Management of temporary superfluous resources and maintenance is a recurring topic in many fields. Trains visit a service yard regularly to be cleaned, maintained, and to store them outside rush hours. These activities are currently scheduled by hand, which is difficult and time consuming. This paper proposes several heuristics to construct an assignment method to solve this problem, which is referred to as the extended Train Unit Shunting Problem. A discrete process simulation is constructed for two real life service yards, namely 'Kleine Binckhorst' and 'Cartesiusweg', to assess the methods. The methods are not capable of obtaining a perfect solution, but most proposed methods are able to solve problems of a reasonable size in a few seconds. The methods incorporate logistical knowledge obtained in practice to optimize flow through the yard, resulting in a shunting policy that is effective and easy to implement in daily practice.

Key Words – TUSP, Discrete process simulation, Shunting policy, Rail yard, Train parking, Train servicing

1. INTRODUCTION

The Dutch national railway operator -NS- owns a fleet of trains of which a major part is not in use during most of the day, causing a parking problem. The national railway network has a high demand of passenger transport during peak hours while only a few trains run during the night, making a big part of the fleet temporarily redundant. Trains that are not in use are parked at dedicated rail yards.

1.1. About our research

The main functions of rail yards comprise the parking of trains and executing small maintenance tasks, such as cleaning, safety checks and small repairs. Maintenance tasks generally require a dedicated track at the yard, so additional movements are required to get trains at specific locations in the yard. These additional movements cause a reduction of the usable parking capacity, since space has to be reserved for the movement of trains.

NS is expanding its fleet of approximately 3000 rail wagons with an additional 1000 rail wagons within the next 5 years. It is therefore important to determine if the existing yards are able to cope with the extra demand by determining the capacity while performing maintenance tasks.

Several maintenance tasks take place in a passenger train service yard. Internal cleaning, external cleaning and safety checks are considered to be most important services with respect to their influence on the dynamic track capacity. Besides the maintenance tasks, the type of trains entering play

an important role in the usable capacity. A lot of external factors, over longer and shorter time periods influence in what sequence trains arrive at a yard, impeding the logistical planning.

1.2. Related work

The basic Train Unit Shunting Problem (TUSP), described by Freling et al. [1], treats the matching of arriving and departing trains and the assignment of trains to tracks. Many authors explicitly treat the arranging of trains in a correct order, and do not take other tasks into account. Some authors do take other task into account, but treat the problems separately.

To improve on this methodology, container logistics offer inspirational insights because of the similarities in the processes and their pragmatic approach. Many container problems are solved by using heuristics and defining a sequence in which actions take place. Advantage of this approach is that it is easy to implement in a problem with regularly changing input.

1.3. Approach

The focus in this paper will be on processes inside a shunting yard. The heuristics approach of several container stacking problems is used to create an alternative method for a TUSP with service tasks; the extended train unit shunting problem. By using these heuristics, an integral model of matching, assignment of tracks, assignment to service activities and shunting movements can be made.

2. MODEL DESCRIPTION

2.1. General

The two most important objects in the TUSP are tracks and trains. Yard equipment such as wash installations could be considered as a third, but their characteristics are covered by the characteristics of a normal track, except they are reserved for specific activities.

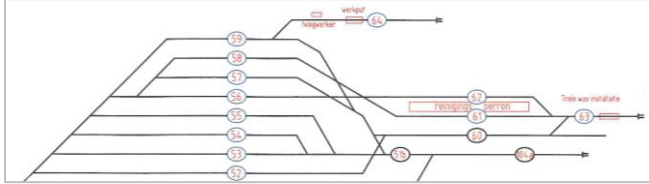


Fig. 1-layout of a railyard (Kleine Binckhorst)

Trains appear at the entrance at a specific time and have to be ready to leave at a predetermined time. The parking capacity of the yard is determined by the combined length of tracks. The real capacity is however influenced by the distribution of different train lengths along tracks and the required space for movements.

2.2. Tracks

Tracks are used by trains for movement around the yard or parking. They have a fixed lay-out and can only be used by one train at a time. Some side constraints arise from the characteristics of the tracks and some assumptions are made, as will be explained next.

Tracks have a predefined maximum capacity, divided into slots of a standard rail wagon length. It is not possible to assign a train to a track if the remaining capacity is not sufficient.

From a logistics perspective it is assumed to be beneficial to assign a preferred entry and exit point to a track. By obeying the preferred direction, a logic flow through the yard can be obtained in a way that proved to be efficient in practice. In this way existing process knowledge is incorporated in the problem solution. Open end tracks are assumed to have an entry point at one side and an exit point at the other side; the first train entering the track is the first to exit: First In, First Out or FIFO. We apply the FIFO strategy to the 'Kleine Binckhorst' yard. Dead end tracks have only entrance and exit points at one side of the track; the first train entering the track is the last to exit: First In, Last Out or FILO. 'Cartesiusweg' is an example of a yard with FILO tracks.

It takes time to move switches and cover distance.. The assumed time is independent of the length of the track, the type of switch and the direction. The time it costs to cover a segment of track is set at one minute, and the time to cross a switch is set at thirty seconds This assumption is common in shunting problems at NS.

For safety reasons all shunting operations should be separated in time and distance at all times. If one shunting

operation is active, no other shunting operations are allowed on all neighbouring tracks to avoid collisions.

2.3. Trains

Train units are rail vehicles consisting of multiple rail wagons in a fixed composition that can be moved by a train driver. Train units of the same type and various lengths can be coupled to form longer trains, this procedure is referred to as 'combining'. The reversed procedure of disconnecting multiple trainsets, is referred to as 'splitting'. Trains have a maximum total length, depending on the type.

A dataset of arriving trains is determined in advance of the simulation. Trains of a specific type with given length will arrive at a given time. The set of trains leaving the yard is also given in advance. Departure times are used to determine if a train finished all service tasks in time to start its designated passenger service.

2.4. Input

Datasets containing all information about arriving and departing trains, including type, length and composition were constructed by R. van den Broek [2]. There are 200 artificial instances in the dataset; 10 instances for each even number of arriving train units from 2 to 40 (2,4,6,...,38,40). Due to the predetermined arrival and departure times in the dataset, and the fixed assignment procedures of tracks, the process is entirely deterministic, so no replications of the same input are needed since it would result in the same solution. The planning period in the datasets spans from 8.00 AM till 8.00 AM the next day.

2.5. KPI's

Important factors of the problem are: 'storage capacity', 'processing time of trains' and 'occurrence of unfinished maintenance'. All three of these factors can be covered by 'feasible solutions' to some degree. The feasibility of an instance with a high number of trainsets will prove a high storage capacity. The in-time finishing of trains suggests that the processing time of trains is fast enough for the system to cope with the demand. The not-in-time finishing of trains is covered by the breakdown of the not-feasible solutions.

The performance of the method can be judged on basis of the following Key Performance Indicators:

- Feasible solutions : Number of instances with a solution.
- Calculation Time : Time needed to calculate solutions.

The quality of the solution can be determined using other KPI's, which can be counted for both feasible and unfeasible solutions:

- Violation of preferred order : number of trains that were not parked in the preferred order.
- Combine at departure : number of trainsets in which the train is not complete while parked.
- Moves : number of train moves needed for solution.
- Splits : number of trains that were split.

3. SOLUTION METHODS

3.1. General

The TUSP can be described as a ‘job shop problem’, which means that limited resources should process the jobs that enter the shop. A discrete process simulation of a rail yard is created using the TOMAS and Delphi software, in which tracks are resources, and trains are jobs to be handled. The simulation is performed for one day.

Three different heuristics are used to improve the solution of an extended train unit shunting problem with service activities. The first method is based on ordering on train type. The second method is based on ordering on departure time, and the third method is a variation on the second method, including reshuffling of vehicles.

3.2. Train selection procedure

The main difference in the solution methods can be captured in the way arriving trainsets are matched to departing trains.

In method 1, Type Based Strategy (TBS), at creation of the train it is assigned a value based on its type. All trains of the same type and length receive the same number. Longer trains are assigned a lower number, i.e. a higher priority to move through the system (in accordance with Lentink et al. [3]), since existing evidence indicates these are more difficult to place.

In method 2 and 3, both variations on the In Residence Time Strategy (IRTS), at creation the train checks a pre-determined array based on the dataset, from which it derives the order of leaving trains. According to this sequence it can assign priority to the train (10 for the first train leaving, $10 \times X$ for the X th train leaving). The priority considers the type of train entering and leaving. When a leaving train is assigned, the train will be deleted from the array so it will not be assigned again. The next train of the same type will be assigned the priority corresponding to the next leaving train of that type and length.

If a matched part does not correspond to the entire length of the train, an additional train is created for matching of the remaining part of the train, except in the ‘Limited Split’ variation; then the train selection procedure continues in search of a match for the entire train.

3.3. Track selection procedure

During movement through the yard, driven by the required service activities, a train will assign itself a new location based on some assignment rules. At arrival, trains are assigned to the track with the most remaining capacity. For cleaning and parking, the sequence of train movements is based on priority; trains with a high priority are allowed to move first (TBS: long trains, IRTS: trains that are planned to leave early).

In general trains are only allowed to park in sequence according to priority:

- TBS: Park only behind same type or at empty track
- IRTS FIFO: Park behind highest lower priority
- IRTS FILO: Park in front lowest higher priority

In addition to the basic policy some optimisation is included in the simulation: if a track happens to contain a number of wagons equal to the maximum capacity of another track, the content of both tracks is swapped. In IRTS it is preferred to park at a track if the capacity will be used to the maximum, regardless of the gap in priority but provided that the prescribed order is not violated.

For IRTS several variations with small additions to the policy have been made.

- ‘Combine Trains’: Only trainsets of the same train are allowed to park on a track until the train composition is complete, except if there are no other parking options possible.
- ‘Reverse Order’: Trainsets of the same train are allowed to restore the correct order by temporarily shunting a train from the storage track to a side track, in order to acquire the desired order of the train units.
- ‘Longest Track First’: Assign the longest tracks first instead of the shortest tracks.

These algorithms aim to optimize the order of trains at a track, while maintaining a certain flexibility.

4. EXPERIMENTS AND RESULTS

The simulation is performed for two different locations with a completely different layout and size. At first the ‘Kleine Bickhorst’, a small carousel type yard near The Hague is assessed. Then the ‘Cartesiusweg’, a large shuffleboard type yard near Utrecht is reviewed.

4.1. Kleine Binckhorst

TBS proves to be inefficient with the use of track length, since trains of the same length rarely add up to the full length of a track. TBS is able to solve instances up to 20 train units.

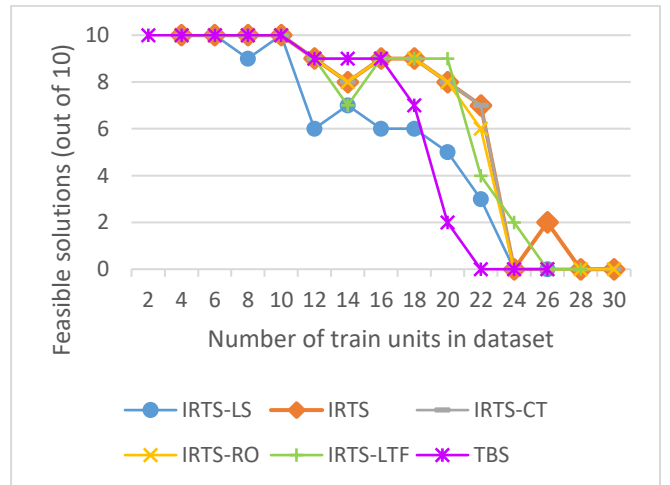


Fig. 2-Feasibility of the proposed methods per dataset

The IRTS setup is able to introduce more differentiation in train lengths at a track, and is therefore generally better performing than TBS in terms of track capacity efficiency. IRTS-LS (Limited Splits) strives to limit the number of splits. If there exists a leaving train with the same composition as an arriving train, it is matched even if it is the first train arriving and the last train leaving. A drawback of this rule is that it might be necessary to assign a part of an early arriving train to an early leaving train, in order to have that train ready to leave in time. In many cases a second trainset of the same type has not even arrived in the yard when the first train should leave. As a result, many trains in the IRTS-LS setup are late, leading to a low success rate in small datasets. The second setup should solve this issue.

In the basic version of IRTS trains are matched if the first trainset of a train matches a part of a leaving train. As a result, some more splits and moves are needed to process all trains in the yard. The number of instances of a train being late drops however from 85 to 35 (from 7,5% to 3,1%). IRTS is able to solve instances up to 26 train units. In quite some instances a part of a leaving train becomes blocked by other trains, before the consist is complete. Although the order of trains in the yard is correct, and thus all train units of a train are unblocked before they have to leave, it might be preferred to complete the trains in an earlier stadium. For this purpose, the third method is proposed.

In IRTS-CT (Complete Trains) a track is blocked for other trains until a train is completed. The feasibility for the datasets up to 24 trainsets is equal to previous setup. However, in the simulations with 26 trains the number of feasible solutions is reduced compared to IRTS. Figure 3 shows however that while more trains are combined at an early stadium, also more trains are forced to park in a wrong order. This could be explained by the fact that the higher percentage of long trains reduces the likeliness of a track having enough capacity for the entire consist. Moreover, the long consists reduce the chance of a track being filled to maximum capacity, because less variations of trains on a track are possible.

In all three previously described setups it occurs that a second part of a train is cleaned and parked before the first part of the same train. As a result, these trains end up in the yard in the wrong order, and cannot be combined in the right order until other trains left the yard. There is however a track (bottom right in Figure 1) that enables a shunting movement to restore the correct order of parking. This ability is used in IRTS-RO (Reverse Order). The method is able to solve one instance less than the IRTS-CT variant. It does however succeed in its intention; the number of trainsets combined in an early stadium increases significantly. On the downside, just like the IRTS-CT setup there is a higher percentage of long trains stored on the tracks, resulting in less flexibility and less efficient track use. The reduced flexibility leads to more trains that cannot be stored in the right order.

The intention of the last setup, IRTS-LTF, is to increase the flexibility. By allowing the trains to park on the longest

tracks first, the theory is that there is a higher chance of the longest track to contain the maximum capacity of a shorter track. Due to the possibility of switching the content of tracks, this should increase the number of tracks filled to maximum capacity. The method is able to solve other sets than before, including the set with 24 trains in which no setup had found a solution yet, but fails more sets in total. The setup achieves another reduction in the number of splits at a late stadium, but just like in previous occasions it is at cost of the number of trains that are parked in the right order, as can be seen in figure 3.

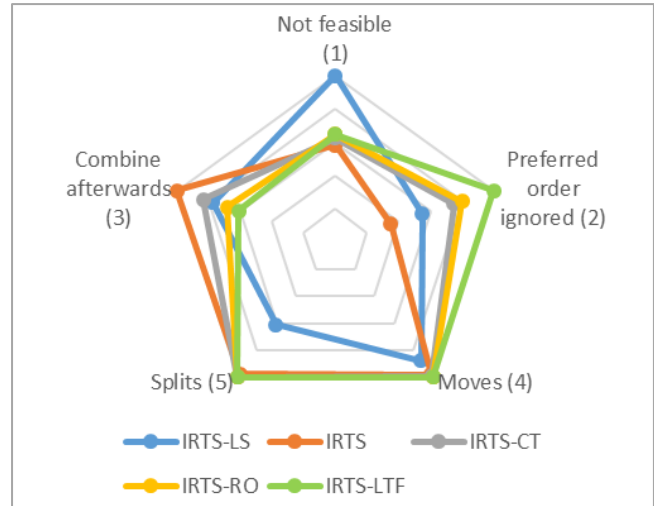


Fig. 3-Comparison of the IRTS setups, scaled from 0 (center) to the worst performing score

Overall, the IRTS-LS method performs well on the least important KPI's, but terrible at the most important one. IRTS performs best on all factors, except the third important factor. Each attempt to improve the third factor, results in a significant downturn of the more important factors. On this basis, the IRTS method proves to be the best performing method. Therefore we use this method during the rest of this research.

4.2. Comparison with other methods at Bkh

In figure 4 and 5 a comparison is made between IRTS and various other methods developed under supervision of NS. IRTS is able to find a solution in the same range as Tabu Search (Row Generation), developed by F. Wolfhagen [4]. TS-RG is able to successfully solve 6 instances more than IRTS, mainly in datasets containing less trainsets. Although the TS-RG is much more sophisticated and takes a significant amount of time to compute, up to 1800 seconds per solution compared to 6 seconds with IRTS, it is not performing significantly better.

OPG, one of the earlier developed tools by NS, is able to successfully solve 7 instances less than IRTS. OPG has a higher uncertainty of solving successfully instances with less trainsets, but it is able to solve some of the instances with more trainsets than IRTS.

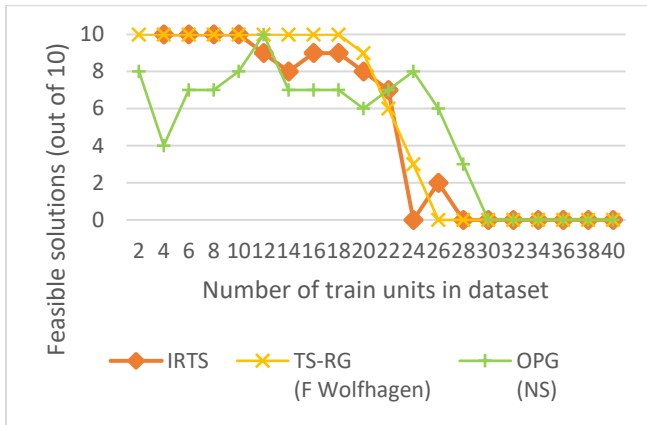


Fig. 4-Comparison between proposed method and various other methods (Bkh.kl)

R. van den Broek [Fout! Verwijzingsbron niet gevonden.] developed various solution methods based on Simulated Annealing. The first two methods include service tasks, namely: internal cleaning, external cleaning and safety checks. The second method excludes the safety checks and only considers cleaning tasks. The last two methods exclude all service tasks and the last variation introduces repositioning of trains after parking. The Simulated Annealing solution has a perfect result in instances with up to 34 trainsets, calculated in approximately under 60 seconds. SA uses extra repositioning and splitting of trains to fit trains along the remaining space, which demand additional labour. It does also not include any service task, in contrast to all other methods. IRTS performs not as well as any SA method, however when capacity at tracks is returned at departure of a train, IRTS nears the result of SA-with-Service-Tasks. When comparing the computation times of all solution methods as depicted in table 1, IRTS proves to be significantly faster than all other methods that include service tasks.

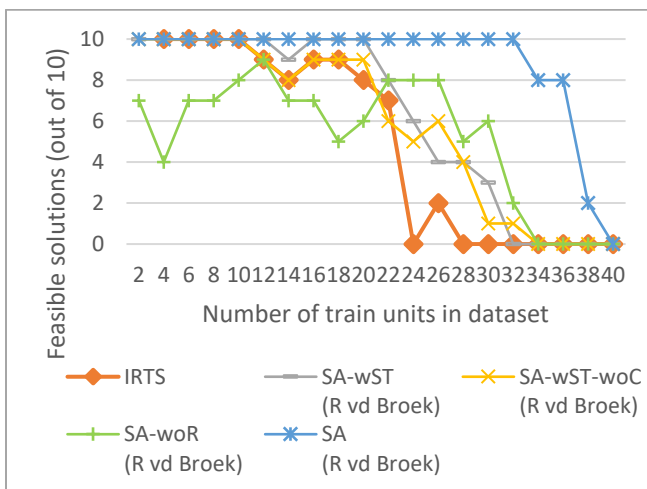


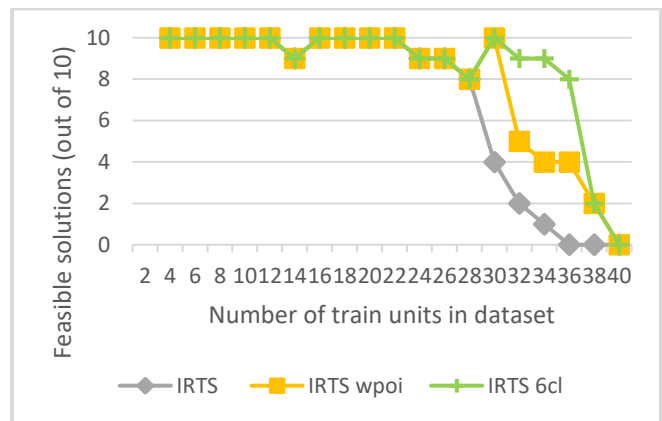
Fig. 5-Comparison between proposed method and R vd Broek (Bkh.kl)

	<20 trainsets	21-27 trainsets	≥28 trainsets
IRTS	3,9s	4,5s	5,9s
TSRG	200s	> 1000s	
OPG	40s	> 300s	
SA-wST	10s	70s	200s
SA-wST-woC	5s	50s	110s
SA-woR	4s	7s	15s
SA	5s	13s	110s

Tbl. 1-Calculation time of various methods for Bkh

4.3. Cartesiusweg

IRTS is also applied to a bigger yard with other layout. Three variations on the method are tested. A differentiation is made between the allowance of wrong order parking (not allowed in 1) and the number of cleaners (five in 1&2, six in 3).



Over 90% of all the cases that are theoretically possible are solved by the IRTS Ctw 6cl method, however the strive for better performance on one criterium leads to a decrease of the performance of at least one other criterium. Reducing the number of infeasible solutions has mainly effect on the number of trains that are parked in the wrong order.

5. CONCLUSIONS

5.1. Model

Various strategies have been proposed to solve the extended Train Unit Shunting Problem (TUSP); the assignment of trains to tracks and service activities. The proposed strategies are based on methods used in the container industry to stack containers. The methods use specific characteristics for categorization of arriving containers, and form logical piles based on this categorisation. This method is translated to the TUSP, resulting in two policies: the 'Type Based Strategy' (TBS) and the 'In Residence Time Strategy' (IRTS). The policies are tested for two rail yards with a completely different size and layout.

5.2. Methods

Both TBS and IRTS do not take future events into account, and it will therefore not be able to outperform most sophisticated mathematical methods. A downside of the proposed methods is the lack of flexibility, due to inability to further optimize a result. TBS performs less than IRTS, due to inefficient track use. Despite the mentioned weaknesses, IRTS approaches the number of successful solved instances of various other methods that include service activities, but in a fraction of the calculation time. Besides the performance and speed of the method, it gives a good representation of what is possible in a real service yard, even when uncertainty of future events is high. This might be the biggest advantage of IRTS: it generates a result that is robust, even in a chaotic system. It is able to cope with changing circumstances, which might be useful in one of the world's busiest rail networks. So above all the solution method illustrates a scenario that is easy to implement and execute in real world. Overall can be concluded that IRTS is a fairly simple policy that performs remarkably well.

5.3. Yard layout

By analysing 2 different locations of different sizes and setups, an insight is given in the effect of different factors. In general, the more possible combinations of trains and tracks, the better the performance. Dead end tracks with build in cleaning platforms makes the system significantly less complex. Dead end tracks are however not beneficial for maintaining the prescribed sequence when the yard nears its maximum capacity, since early leaving trains and late arriving trains tend to block each other.

6. RECOMMENDATIONS

As mentioned before, each solution method for itself has little flexibility. At the same time, each variation creates new possibilities, but regularly with a negative effect on other aspects of the solution. It might be beneficial to apply the artificial instances to multiple solution methods, and select the best result. In this way one would be able to create more optimised results, without a reduction in the number of feasible solutions.

Yards are not by definition empty at the start of the day, or even at any moment of a day, so the assumption of an empty yard is not always realistic. It might therefore be interesting to create a simulation that is able to cope with initial filling of the yard. The policy is perfectly able to solve problems with trains already present in the yard, but the influence is not tested yet. The addition of initial filling also makes it possible to rerun the problem when the input changes due to disturbances in the network, which is earlier mentioned to be a potential benefit of this policy.

Finally some remarks could be made about the datasets that were used. Only a few train types are represented in the artificial instances, which makes the solution a lot less complex, but also a little less realistic. It might be interesting

to construct datasets with other types of trains or more types. The higher variety of types might change the performance of the various methods, since the matching of trains leaves less room for optimization. At the same time, the higher variety of train lengths, especially the addition of smaller train units, will probably increase the possibilities of filling a track to full capacity.

The last suggestion for further research also deals with the content of the dataset. In this research only a limited amount of service activities is tested. Especially the addition of external cleaning might be interesting, since it is a real time consuming activity using a scarce resource. The addition of external cleaning and other activities such as safety checks could be implemented using the same assignment rules as proposed in this research. It might reveal new bottlenecks, in the same way the cleaners at the cleaning platform proved to be a bottleneck during this research.

7. REREFERENCES

- [1] R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 186(2):261-272, 2005.
- [2] R. van den Broek, Train Shunting and Service Scheduling: an integrated local search approach, Master's Thesis, Department of information and computing science, Utrecht University, 2016.
- [3] R. M. Lentink, P.-J. Fioole, L. G. Kroon, and C. van't Woudt, Applying operations research techniques to planning of train shunting. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*, pages 415-436, 2008.
- [4] F. Wolfhagen, The Train Unit Shunting Problem with Reallocation, Master's Thesis, Department of Econometrics & Management Science, Erasmus University Rotterdam, 2017.

Appendix B

Treatment time of train types

	DDZ4	DDZ6	VIRM4	VIRM6	SLT4	SLT6
lengte (m)	101,08	154	108,56	162,06	69,36	100,54
normtijd A-controle (min)	76	90	48	60	45	51
periode A-controle (dagen)	12	12	12	12	12	12
normtijd B-controle (min)	15	18	11	14	23	27
periode B-controle (dagen)	2	2	2	2	1	1
normtijd kopmaken zonder lopen (minuten)	4,21	4,21	4,4	4,4	2	2
Looptijd per bak (minuten)	0,5	0,5	0,5	0,5	0,33	0,33
normtijd combineren (minuten)	3	3	3	3	3	3
normtijd splitsen (minuten)	2	2	2	2	2	2
normtijd inwendig reinigen (minuten voor 1 persoon)	186	278	186	278	78	101
periode reinigen intern (dagen)	1	1	1	1	1	1
periode reinigen zijanten zeep (dagen)	7	7	7	7	7	7
periode reinigen koppen zeep (dagen)	7	7	7	7	7	7
periode reinigen extern oxaal (dagen)	63	63	63	63	63	63
Normtijd reinigen zijanten zeep (minuten)	4	6	4	6	3	4
Normtijd reinigen koppen (minuten per kant)	10	10	10	10	7	7
Normtijd reinigen zijanten oxaal (minuten per bak)	16	24	16	24	12	16