Finding the optimal set of parking locations for maintenance trains in the Dutch railway network

An optimisation approach using a combination of discrete-event simulation and simulated annealing

by

O.L. Olberts

Master Thesis

To obtain the degree of Master of Science

Delft University of Technology,

Department of Transport, Engineering and Logistics

Student number:	4633822	
Report Number:	2020.MME.8476	
Project submitted:	January 5, 2021	
Project supervisors:	Dr. B. Atasoy	TU Delft
	Prof. dr. R.R. Negenborn	TU Delft
	W. Boiten	ProRail
	K.S. Hofstra	ProRail

TU Delft

Contents

List of Figures

List of abbreviations

CFLP	Capacitated Facility Location Problem
DES	Discrete-Event Simulation
DRP	Dienstregelpunt (Service Control Point)
FLP	Facility Location Problem
KPI	Key Performance Indicator
LP	Linear Programming
LAP	Location-Allocation Problem
MFLP	Multiple Facility Location Problem
MOFLP	Multi-Objective Facility Location Problem
MR	Maintenance and Renewal
MRP	Maintenance Routing Problem
MVSL	Maintenance Vehicle Stationing Locations
NSF	Neighbourhood Search Function
RSA	Random Search Algorithm
SA	Simulated Annealing
SFLP	Single Facility Location Problem
Sim-Opt	Simulation Optimisation
UFLP	Uncapacitated Facility Location Problem
ULSA	Utility Level Search Algorithm

List of Tables

Summary

The Dutch railway system is subject to maintenance, which is carried out by a group of rail contractors commissioned by ProRail. These rail contractors need parking space to park and prepare maintenance trains in between projects. Currently, ProRail reserves around 76km of track, distributed over 49 parking locations throughout the network, for the contractors to make these activities possible. This collection of parking tracks is subject to maintenance itself and thus is an expenditure that ProRail aims to decrease. Contractors prefer a large amount of parking locations distributed across the network as they can park and prepare trains close to their upcoming maintenance projects. This decreases travel time to and from maintenance projects. ProRail wants to optimise the combined costs for the maintenance of parking locations and journeys made by contractors. This leads to the main question of this research: *"What is the optimal set of parking locations for railway maintenance vehicles in the Dutch railway network?"*

A literature study has been done in order to find a feasible solution method for the case study. The problem is defined as a Facility Location problem (FLP) and this has been studied extensively in literature. Due to the dynamic nature of the problem in this research, a Discrete-Event Simulation (DES) is chosen to model the processes of the maintenance trains in the Dutch railway system. Due to the size of the solution space a Simulation-Optimisation (Sim-Opt) method using a Simulated Annealing approach is proposed, wherein the DES acts as an evaluation function for the optimisation agent. The work of Qin (2012) and Frausto-Solis et al. (2007) is used to formalise the simulated annealing process. This process is expanded by implementing relevant usage level data of the simulation output into the neighbourhood search function.

The current system and the accompanying processes have been analysed on the basis of the railway network, parking locations, train units and expenditures. The yearly parking capacity is requested by the contractors active in the dutch railway network. The requested parking capacity in 2019 amounted to 76km (whereof 59km usable), while the total amount of maintenance trains have a combined length of 22km. Contractors are not bound to certain areas of the railway network and it is therefore possible that their maintenance trains travel throughout the entire network and cover relatively large distances. Contractors allocate the space on parking locations interchangeably among themselves. Due to this dynamic process the available capacity on parking locations is only known shortly in advance, making it practically impossible to plan all journeys and stays in advance at the start of the year. The travel costs of maintenance trains are difficult to specify, since the expected expenditures are calculated into a yearly contract between the contractor and ProRail, but can be subdivided into four aspects: staff, energy, vehicle maintenance and vehicle depreciation. The costs to maintain the infrastructure of parking locations consist of renewal costs (every 60 years on average), regular maintenance costs, management costs, facilities expenditures and safe guard system of switches and tracks. Tracks reserved for maintenance trains are usually a part of a bigger yard. Therefore, the maintenance costs can best be specified through the renewal and regular maintenance cost of the infrastructure directly linked to the tracks used by maintenance contractors plus the connecting switches and crossings. Further costs can be the cost of failure, both the direct financial (repair) and operational (delays for other railway users) aspect. The cost of failure is not taken into account in this research.

A Discrete Event Simulation (DES) has been designed to simulate movements of trains trough the network and the related dynamic processes. The main components of the simulation are the projects and trains. The entire simulation is modelled around the situation of the year 2019. The projects were imported beforehand and simulated according to the yearly project schedule. During the simulation projects are assigned to an available train in the system. In the period before the project is due, the train travels to a parking location with free space nearest to the project location. Parking locations can only be occupied up to ~50% of the usable track length in order to facilitate shunting. The DES was implemented in the Sim-Opt design of Qin (2012) using a SA process. This process was extended by implementing an evaluation of the relative usage level of the parking locations during the previous iteration. This evaluation guides the search through the solution space by generating promising neighbouring solutions during the Sim-Opt process. The new neighbourhood search algorithm (ULSA) is tested against the random neighbourhood algorithm (RSA) used by Qin (2012) in Chapter 5. The evaluation is performed for both search algorithms using multiple instances of the Sim-Opt. The results of this evaluation show that the ULSA converges sooner than the RSA. Furthermore, the spread in the best solutions across all instances is tighter when applying the ULSA. When looking at the details of the solutions, it can be seen that the ULSA finds solutions across all instances which are more alike. These findings indicate that the ULSA gives a more robust solutions when compared to the RSA and makes the SA process more efficient.

According to the simulation used in this research, ProRail can potentially reduce the annual cost related to the travelling and parking of maintenance trains by ~20%. A large part of this improvement can be achieved by rethinking the parking location strategy. This research found that the best solutions in terms of overall cost includes on average 22 parking locations, which can reduce the maintenance cost by ~30%. This is in stark contrast with the current situation, where 49 parking locations are used for maintenance trains. The total travel costs show a relatively small increase of ~9%. The capacity for the optimal solution is close to the minimum capacity (including a capacity buffer of 100%) requested by the contractors. This implies that the increase in travel costs does not justify the use of more capacity than twice the train fleet length. The improved performance of the ULSA can encourage ProRail to use the relative usage level as a metric for determining the value of a parking location.

The included parking locations in the Simulation Optimisation (Sim-Opt)s are spread across the network, but are preferably located near the project locations. The number of project in a certain area of the network is related to the amount of infrastructure. As a result, parking locations near the outer part of the railway network are excluded more often and parking locations towards the centre of the railway network are included more often in the Sim-Opt. Special case *S1*, shows that the use of a single parking location, located in the centre of the railway network, would decrease the total cost considerably (by 24%). Such a situation is thought to be impractical, since the added travel time for contractors, would reduce the effective labour time for the engineer. Special case *S2* shows that in an extreme scenario in which no extra space is needed for shunting, the change in travel costs still does not justify the use of more capacity than the absolute minimum. The parking locations selected by the model are in this case, again, located across the network. This implies that a nationwide coverage is important. However, it is also shown that the size of a parking location can overrule the strategic positioning to some extend.

1
Introduction

This research is conducted at ProRail. ProRail is the rail infrastructure manager of the Netherlands, and is responsible for construction, maintenance, management and safety of the Dutch rail network. The introduction will start with an explanation of the project motivation, followed by the problem definition and approach.

1.1. Project Motivation

The Dutch railway network consists of around 7000km of railway track and is under constant pressure of an increase in passenger and freight transport. Together with the increase in railway traffic, ProRail strives to increase punctuality throughout the system and decrease the overall system costs. Besides public and freight transport, other railway users include contractors and maintenance trains to maintain the quality of the tracks and to resolve incidents. The infrastructure used to station these maintenance trains however, add to the complexity of the railway system. A higher system complexity by itself increases system failure rate, maintenance costs and both planned- and unplanned unavailability. Which could argue to remove a number of the existing parking locations. On the other hand, an abundance of parking locations could lower the contractors travelling time in preparation, as well as during maintenance projects. This in turn could decrease track decommissioning, hindrance of other traffic and lower the amount of planned unavailability. Therefore, these parking locations have to be positioned strategically in the network to enable optimal use of their service. ProRail seeks an integral and strategic clarification on the required number and positions of parking locations for maintenance trains in the network.

Figure 1.1: Example of a parking location (Spoorpro, 2019)

Figure 1.2: Example of a maintenance project and maintenance train (Spoorpro, 2013)

1.2. Problem Definition

The Dutch railway system is subject to maintenance, which is carried out by a group of rail contractors commissioned by ProRail. Rail contractors need parking space to park and prepare trains in between projects (Figure 1.1). Currently, ProRail reserves around 76km of track, distributed over 49 parking locations throughout the network (Figure 1.3), for the contractors to make these activities possible. This collection of parking tracks is subject to maintenance itself and thus is an expenditure ProRail aims to decrease. Contractors prefer a large amount of parking locations distributed across the network as they can park and prepare trains close to their upcoming maintenance projects (Figure 1.2). This decreases travel time to and from maintenance projects. The travel expenses will end up at ProRail and this amount is expected to increase if there are fewer parking locations reserved for the contractors. ProRail wants to optimise the combined costs for the maintenance of parking locations and journeys made by contractors.

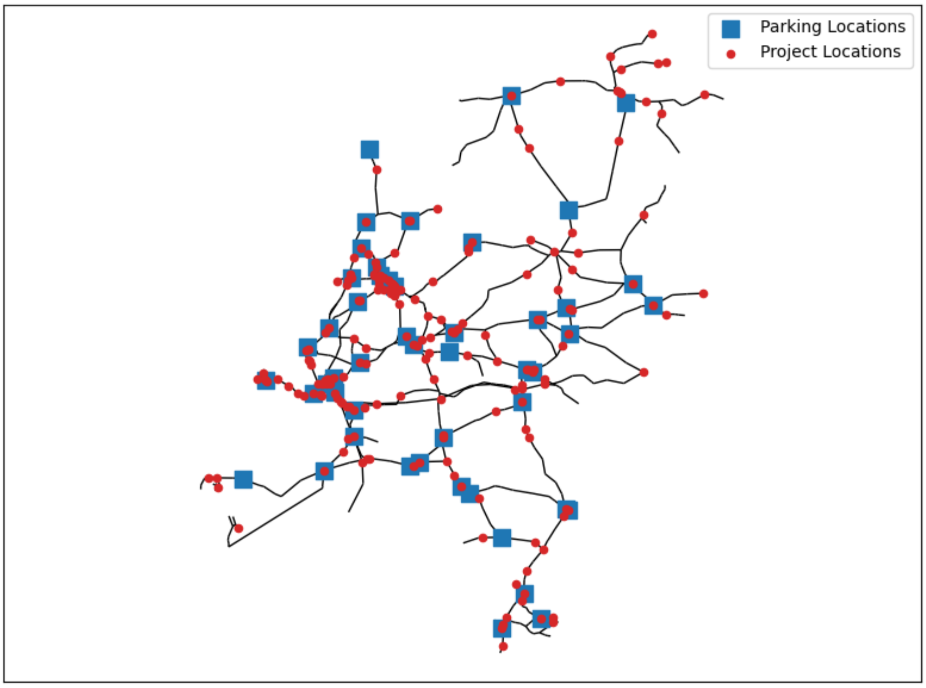


Figure 1.3: Illustration of the project locations and parking locations in the Dutch railway network.

1.3. Research Objective

The project aims to contribute to a methodology to determine positioning of parking facilities for railway maintenance vehicles. This initiates the work for the literature gap on a facility location problem applied to the parking of railway maintenance vehicles. The objective of this research in practice is to provide a model to aid decision making as to where parking capacity is needed for railway maintenance equipment. The project aim and objective lead to the main research question:

- **What is the optimal set of parking locations for railway maintenance vehicles in the Dutch railway network?**

The main research question is supported by the following sub-research questions:

- What optimisation method can be used to find the best solution.
- How are parking facilities for maintenance trains currently being used?
- How can the railway network and including maintenance related movements be modelled?
- Which recommendations towards ProRail arise from the results of the optimisation method?

These sub-research questions govern the approach taken in this project.

1.4. Scope

In order to answer the research questions, the scope of the research is narrowed by the following assumptions:

- No new parking locations are added to the system. The aim is to find an optimal solution by determining the inclusion or exclusion of existing parking locations.
- The dynamics with neighbouring countries is not taken into account. The Dutch railway system is simulated as a closed process.
- Tactical and operational decisions such as optimal routing of trains and the scheduling of maintenance projects are not considered. A basic routing model (shortest path) is used in this research.

1.5. Approach

This research question will be tackled using a Sim-Opt approach. A discrete event simulation will simulate a years worth of maintenance jobs and the accompanying journeys made by the maintenance trains. The simulation takes as input parameters a list that specifies which parking locations are included and which are omitted from the system. The performance of the Sim-Opt will be evaluated on the basis of the Key Performance Indicator (KPI)s. The KPIs for this research are: the aggregate cost of travel and maintenance of the included parking locations, the number of included parking locations and the available parking capacity. The simulation acts as an evaluation function for an optimisation algorithm. The optimisation algorithm constructs a solution to the problem as a new set of input parameters for the next iteration of the simulation. The simulation evaluates the performance of the suggested solution. The simulation and optimisation interact to find a near-optimal solution to the problem. Due to the large size of the solution space the optimisation algorithm will use the heuristic technique Simulated Annealing (SA) to search through the solution space. The search algorithm will determine a neighbouring solution. Two different search algorithms are implemented: a random search algorithm and an algorithm which takes into account information on the utility level of the parking locations during the previous simulation iteration. These search algorithms are compared in order to determine the level of importance of the utility level on the solving process.

1.6. Report Structure

A literature review and the theoretical background needed to determine the optimisation method for the project will be discussed in Chapter 2. Then, in Chapter 3, the way the current system operates will be stated, together with the needed assumptions and simplifications. Chapter 4 will cover the modelling of the railway network and the including maintenance projects. This chapter will also explain the details of the simulation-optimisation technique. The results from the experiments will be stated in Chapter 5. Conclusions from the research, including a discussion and recommendations for ProRail and future research, are given in Chapter 6.

2 Literature Review & Theoretical Background

Railway maintenance problems have been studied extensively in the literature (Lidén, 2015), but are mainly linked to routing or scheduling problems. For example, Zhang et al. (2013) try to find an optimal time schedule for the maintenance activities and combines the search for optimal routes for the necessary equipment and crew to minimise the total cost. Dao et al. (2018) focuses on determining the best maintenance schedule based on the track accessibility. The problem in this research is focused on locating parking facilities and fewer studies have linked the facility location problem to a railway network. Kho (2016) has combined the routing of trains with the location problem for maintenance facilities in the Dutch railway network. The facilities were located based on the passenger train schedules to minimise deadheading cost. Liu et al. (2018) has conducted research on the location and size of classification yards and is concerned with freight trains and the allocation of their cargo within a classification yard. Here, a strategy for the node location and yard size was linked to the yard establishment cost, improvement cost and to the train connection service plan.

The literature on railway research has a focus on routing and scheduling of maintenance jobs and rolling stock. However, the placement of stationing locations for parking of maintenance vehicles has not been studied yet for railway networks.

2.1. Facility Location Problem

The study of finding the optimal placement of facilities to minimise transportation costs, falls under the term Facility Location Problem (FLP). Solving the FLP is a valuable practice in strategic planning for various companies (Reza Zanjirani Farahani, 2009). Besides transportation costs, a typical FLP can also take into account other factors like, facility opening costs, market share, maximum coverage etc.

There is a wide variety of different types of the FLP, which are applied in various application fields

(Celik Turkoglu and Erol Genevois, 2020). Celik Turkoglu and Erol Genevois (2020) also classify the different types of FLP based on the following features:

Purpose

FLP's can either be applied in order to analyse or to select the location for facilities. Analysis of locations can assist decision makers in assessing the performance of individual facilities. Location selection on the other hand, focuses on finding the optimal locations for facilities within a bounded area.

Space

FLPs can be modelled in either continuous or discrete space. In Continuous Facility Location Problems, facilities can be located anywhere on an XY-plane defined as the service area (Celik Turkoglu and Erol Genevois, 2020). On the other hand, in discrete facility location problems, facility locations are constrained to a set of discrete points. A special case of discrete problems, is a network location problem, whereby facilities can be only located on nodes, which are interconnected by (weighted) edges. Feasible paths between nodes can only be determined by the shape of the graph. Generally, demand points are located on nodes as well, but demand could also occur on graph edges (ReVelle et al., 2008). In network models the travel distance is simply the total distance, or weight, of all the edges in the shortest path. For continuous or non-network discrete models, a distance metric has to be specified. Common distance metrics include, Euclidean, Manhattan and Chebyshev distance.

Time

FLP can be split into either static or dynamic problems. Static problems are characterised the absence of a time component and by single-stage parameters that remain constant. Dynamic problems have an integrated time component and parameters are allowed to change.

Parameters

FLPs can be deterministic or probabilistic (Daskin, 2013). In deterministic models, the solution is completely dependent on the inputs. In probabilistic models, the inputs are influenced with a level of uncertainty. Usually, probability functions are implemented to alter the inputs in order to model a specific real life problem.

Capacity

A FLP can take capacities of the facilities into account. Usually, Capacitated Facility Location Problem (CFLP)'s represent real world cases more closely. As an added benefit, the CFLP constrains the solution space, since facilities cannot service more than its maximum capacity. A FLP where facilities are assumed to be un-capacitated is called an Uncapacitated Facility Location Problem (UFLP).

Facilities

The number of facilities to be located affects the model size, complexity and therefore, its difficulty. To specify if a location problem consists of locating one or more facilities, the terms Single Facility Location Problem (SFLP) and Multiple Facility Location Problem (MFLP) are used respectively.

2.2. Simulation

Solving a FLP's is generally approached in one of two ways: by Linear Programming (LP) or simulation, with LP being more common. In literature, however, these terms are used interchangeably.
A LP approach solves the FLP by using a mathematical model to assign facilities to demand points and minimise the aggregate weighted distance between all demand points and the facilities. A simulation approach on the other hand, relies less on mathematics, but instead simulates the working system and generally takes into account the system dynamics. A simulation approach is more common for dynamic and probabilistic problems, while the LP approach is associated more with static problems.

The problem of the case study is different from the normal FLP architecture whereby facilities only service demand points closest to them. Here facilities are still to be located in a network, but they themselves do not serve any demand point. Rather, the trains driving through the system are the components which service demand points and can reside at every location in the system. Furthermore, as will be explained in Chapter 3, the system considered in the case study is highly dynamic. Given these arguments, a simulation-based approach is chosen for solving this case study.

2.3. Optimisation

A simulation in itself is not an optimisation tool. In order to use a simulation to search for an optimal solution to the problem, an optimisation algorithm has to be implemented to form the Sim-Opt technique (Qin, 2012). In such an technique, the simulation and optimisation algorithm interact on a sequential basis: the simulation is run with an initial set of input parameter values, after which the optimisation procedure uses the output values of the simulation to evaluate the corresponding performance. Based on the evaluated performance, the optimisation algorithm determines a new set of input values for the simulation (Joines et al., 2001).

Optimisation algorithms can be divided into two categories: exact algorithms and heuristics. Exact algorithms are guaranteed to find the optimal solution in a finite amount of time. For large or very complex problems (e.g. NP-hard), this might become infeasible or at least impractical. Heuristics is a problem solving technique which bypasses this issue by directing a search through the solution space and limiting the computation time or number of iterations. This means that a subset of the solution space is evaluated. As a result, heuristics trades optimality and completeness for speed (Maringer, 2003).

Heuristics algorithms tend to be very specific and problem-dependent. Meta-heuristics are high level strategies to guide the search in the solution space. Meta-heuristics are problem independent and can act as a black box solution method. Meta-heuristics are used more frequently than standard heuristics, since the technique is considered flexible in its application, and thus, easier to adopt for new problems (Maringer, 2003).

2.4. Simulated Annealing

Simulated Annealing (SA) is a commonly used meta-heuristic and depending on the problem, one of the most efficient heuristic technique (Maringer, 2003). Simulated Annealing (SA) is a technique for searching through the solution space of a problem or function. A neighbourhood function constructs candidate solutions iteratively and worse solutions are accepted based on a probability determined

Paper	Space	Time	Parameters	Capacity	Objective	Facility types	Modelling	Optimisation
Tonissen	N	S	D	C	S	S	LP	E
Qin	N	S	D	C	S	S	LP	H
Liu	N	S	D	C	S	S	LP	H
Fredrik Persson	C	D	P	U	M	S	SIM	H
Nasab	C	S	D	U	S	M	LP	H
Rahmaniania	N	S	D	C	S	S	LP	E + H
Hidaka	N	S	D	U	S	S	LP	H
Jesica de Armas	N	S	P	U	S	S	LP	H
Ka Yuk Lin	N	D	P	C	S	S	LP	H
This Research	N	D	P	C	S	S	DES	H

Table 2.1: Overview of the FLP features used in the reviewed literature in relation to this research. Space: Network (N), continous (C). Time: Static (S), Dynamic (D). Parameters: Deterministic (D), Probabilistic (P). Capacity: Capcitated (C), Uncapcitated (U), Objective: Single (S), Multiple (M). Facility types: Single (S), Multiple (M) Modelling: Simulation (SIM), Linear Programming (LP), Discrete-Event Simulation (DES). Optimisation Technique: Heuristic (H), Exact (E).

through a cooling schedule. SA was first proposed by Kirkpatrick et al. (1983) as a stochastic optimisation technique. Qin (2012) incorporated three possible mutations into the SA neighbourhood function for solving the FLP. Thirty freely available benchmark problems were analysed as a proof of concept of their neighbourhood search function. Hosseini nasab and Mobasheri (2013) apply the SA technique to the placement of different machine types in order to minimise the distance travelled by the products within a factory. The SA process consisted of two possible mutations for generating new solutions and marked it as an efficient approach to the FLP.

Qin (2012) solves a FLP of similar size to the problem of this research using a Sim-Opt technique. Furthermore, the SA search method used by Qin (2012), gave promising results. Due to the similarities, the Sim-Opt technique and SA search method used by Qin (2012) are applied to solve the problem of this research.

2.5. Summary

Table 2.1 gives an overview of the reviewed literature in relation to this research. The different features of corresponding FLP's are specified together with the solving concept and optimisation technique. To summarise, the main contributions of this research are as follows:

- This research is the first to investigate parking locations for maintenance trains in a railway network.
- This research explores the use of Discrete Event Simulation for the Sim-Opt method.
- This research develops the standard SA neighbourhood function to a more directed search.
- This research performs a case study at ProRail and gives insight into the current use of parking locations and gives recommendations on obtainable system efficiency and future research.

3

Current System

To get a clear understanding of the entire railway maintenance process, this chapter will set out the related aspects. The maintenance process in the Dutch railway system can be split up in the following components: network, parking locations, maintenance trains, infrastructure costs and travelling costs.

3.1. Network

The Dutch railway system consist of about 7200km of track and about 7000 switches. Besides normal track and switches, there are also other objects like bridges, level crossings, the overhead contact system, signals, stations, etc. All these objects need regular maintenance and intermittent renewal.

ProRail defines a set of points in the network, DRP, which are relevant for locating all processes of the train service, rolling stock handling and personnel services. DRP's can be linked to the position of passenger stations, railway yards, important switches, bridges or other significant items in the railway system. The number of DRP's can change over time, due to changes in infrastructure or stations. In the case of 2019, there were 807 DRP's in the Dutch railway system.

Occasionally, maintenance trains need to change direction of travel. To do this, a train has to come to a complete stop and the engine-driver has to move to the other side of the train. In other cases, a locomotive must also connect to the other side of the train. Under normal circumstances, this cannot happen on any piece of track, since the main line is far to busy to leave a time gap for such a manoeuvre. There is a limited amount of places where it is possible to switch direction of travel without hindering other network users. This might be a single sidetrack along the main line, but could also be a yard or freight railway yard where some movement space is available.

3.2. Parking Locations

The Dutch railway system has various yards or freight railway yards throughout the county. Yards can be used to park, service or inspect freight, passenger or maintenance trains. Freight railway yards are generally used to house freight carriages and to make shunting possible for distribution of freight to and from other countries. The capacity used for stationing maintenance trains usually lies amid a yard where also other types of train are stationed. Every year, contractors request the amount of capacity they expect to need for stationing their equipment. This depends on the amount of projects that are planned for the upcoming year. ProRail evaluates these applications, combined with the demands and requests of passenger and freight transport companies, and tries to find a feasible solution to assign every track on all yards as efficiently as possible.

Three types of parking locations can be identified (ProRail, 2020):

1. Gathering locations *(Verzamel locaties)*: Location where supplies and vehicles are gathered, which can then be driven towards the project location.
2. Stationing locations *(Parkeer locaties)*: Locations where wagons and materials are prepared for a project in the immediate vicinity. These can be maintenance, new construction or replacement projects.
3. Preserving locations *(Instandhoudings locaties)*: Locations where equipment and small stock is set up and deployed for maintenance and failures.

Gathering locations are characterised by the relatively long parking periods. Machinery, equipment and material are stored and arranged here during periods when they are not needed for maintenance projects. In 2019, the Dutch railway network had 4 gathering locations with a total usable capacity of 21150 metres.

Stationing locations are being used to temporarily park equipment before and during maintenance projects. These activities can also take place on gathering locations. The parking needs for contractors are accommodated via shifts in the available parking capacity for rolling stock. The Dutch Railway network has 9 stationing locations with a total length of 7500 metres.

Conservation Locations are being used for management and conservation activities of rail infrastructures by rail contractors. These activities can also take place on gathering locations and stationing locations. These locations are characterised by the frequent use and relatively short parking periods. Examples of activities on these locations are: temporary parking of wagons and equipment in preparation of a project, loading and unloading material and the loading and unloading of railway vehicles.

Table 3.3 gives an overview of the total stationing capacity for all stationing locations in the Dutch railway network for 2019. Tracks have an actual length and a usable length. The total amount of usable capacity adds up to 59349m and exceeds the total maintenance fleet length of 22km considerably. However, contractors calculate a buffer into the requested parking capacity to be able to make shunting or executing other movements smoother. Due to this buffer, contractors request a capacity of at least two times the size of the maintenance train fleet. The minimum parking capacity thus lies near 44km.

The allocation of maintenance trains is a highly dynamic process, wherein contractors apply for parking capacity at ProRail via a representative of the contractors. Almost all contractors are part of the Infragroup (a partnership of various contractors) and apply for parking capacity as one entity. Contractors of the Infragroup have made agreements among themselves to interchange the assigned capacity throughout the year. Meaning that, throughout the year, maintenance trains leave certain parking locations and can enter other ones, which are technically reserved by other contractors. Contractors do not have insight in what parking locations are occupied by other contractors at specific moments throughout the year and have to check with the Infragroup if a certain parking spot is free in the coming days or weeks, making it practically impossible to plan all journeys and stays in advance at the start of

[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

Table 3.1: Overview of the estimated total train length per contractor.

the year.

3.3. Train Units

Obtaining an overview of the maintenance train fleet is a complicated problem. Contractors can operate in the rest of Europe, meaning that the fleet is multiple times larger than if it would be to only service the Dutch railway system. Furthermore, trains are not bound to a single country, meaning that a train that has worked on a project in the Netherlands in a certain year might end up in different country and another train will be driven to the Netherlands if a new, similar project, arises. The new train could belong to another contractor, have a different length, etc. The Dutch railway network is not an isolated system and neither are the processes that occur in the system. The dynamics with neighbouring countries is beyond the scope of this project and as a result, an isolated system is simulated.

Furthermore, trains do not always consist of the same combination of wagons and locomotives. Some maintenance trains are able to work and travel on their own, but there is a portion of trains that need a locomotive to move from location to location. During specific projects, building material will need to be supplied and disposed of. For this, ballast wagons are needed, which are passive units. A locomotive is needed to move one or more of these units, making the length of a train combination variable.

In short, the train fleet of the different contractors working on behalf of ProRail is not known to ProRail precisely. Therefore, a fictional fleet of trains has been constructed and used. This fleet is confined to the Dutch railway system only and will thus stay in the system. The total length of the fleet is approximately 22km (ProRail, 2020). Based on expert judgement, a distribution defining the occurrence of length for the trains was used to construct the fleet (Table 3.1) and will be further discussed in Chapter 4.

3.4. Infrastructure Costs

Costs for infrastructure can be divided into two categories: opening costs and operating costs. Opening costs are any expenditures related to the construction of a new parking facility. Examples of these expenditures are land costs, costs per km of new track, cost per new switch or installing facilities (e.g. security, footpaths, water hydrant system, etc.). In the current situation, ProRail has a collection of parking locations which are already opened and operational and the scope of this research does not cover the placement of new parking locations. Therefore, it makes sense to not include opening costs to the cost analysis of this research and only focus on the financial gain by lowering the operating expenditures.

Other costs include: management costs (traffic control, asset management, capacity allocation), supply/facilities expenditures and security/safe guard system of switches and tracks. Since track reserved for the parking of maintenance trains is generally located on yards, together with track reserved for other users, these other costs would not change drastically, when the amount of capacity for maintenance trains is changed.

A part of the operating costs, which can be measured directly when changing the amount of capacity, is the infrastructure maintenance costs. The infrastructure on parking locations consists of track and switches. The side track on parking locations is different from main track lines, since, due to the relatively low speeds and low usage, the loads involved are generally lower. As a result, side tracks have to be maintained according to lower standards, which in turn lowers the annual maintenance cost compared to main track. The same is true for switches. Because of the lower speeds, the loads on switches are lower and less maintenance is necessary.

Infrastructure is subject to intermittent renewal. Some parking locations have been renewed more recently than others, and as a result, some parking locations have a high write off value if they are removed from the system in the near future. Within ProRail, write off of switches or other infrastructure is generally not taken into account. Renewal or removal of infrastructure is based on what is needed for the coming 10-100 years. If a new study finds a different solution would benefit the process in the network and if that means that something that has just been renewed has to be removed then it is going to be removed. In practice this happens rarely, since studies concerning infrastructure usefulness are generally performed 5-7 years prior to the expected end of life of the relevant infrastructure. Renewal of switches is generally leading in the renewal of other infrastructure connected to it, i.e. when an expensive switch needs to be renewed, everything else connected to it (within reason) is going to be renewed as well. However, since this research is concerned with the minimisation of expenditures by taking into account the network process, the cost of renewal will be accounted for. The estimated renewal costs and estimated lifespan will be used to calculate the effective yearly expenditures for a parking location when it is kept in use.

	Regular Maintenance	Renewal
Side Track (per km)	1000	11667
Switch	1000	Depends on the type of switch Ranges from 1167-6667

Table 3.2: Estimated yearly costs in euros for maintenance and renewal for switches and side track.

The estimated yearly costs for maintenance and renewal for switches and track is specified in Table 3.2. The total infrastructure maintenance costs per parking location is given in Figure 3.1. Table 3.3 lists the amount of track reserved for maintenance trains and the number of corresponding switches per parking location.

3.5. Travelling Costs

Contractors usually work for ProRail on a contract-basis, wherein all the expected expenditures are calculated. If contractors spend more on their operations than expected, this extra cost will not be directly passed on to ProRail. However, the expenditures of previous years will influence the price during the negotiation of the next contract. Thus, ProRail still benefits from lowering any expenditures, and in the case of this research, the amount of extra kilometres driven by contractors. Travelling costs can be subdivided into four aspects: Staff, energy, vehicle maintenance and vehicle depreciation.

Staff

Generally, only one engine driver is needed to drive a train. The rest of the personnel needed for a project arrives by their own means of transport. As a result, the contribution to the hourly rate of a maintenance train during travelling can be limited by the hourly cost for the employer of a single engine driver. In consultation with experts, this hourly cost has been set to be 75 euros.

Energy

Trains are either designed to drive on fuel (Diesel) or electricity. While diesel trains are uncommon for passenger train in the Netherlands, practically all maintenance trains are powered by diesel engines or pulled by diesel locomotives. Maintenance trains use considerably less energy when travelling than in use at a project. Nevertheless, contractors do bill ProRail for any additionally driven kilometres. The diesel consumption for a typical diesel hydraulic locomotive is set, in consultation with experts, to 3 litre per kilometre at normal speed under normal load. The diesel price has been taken to be around 1 euro per litre, resulting in a rough estimate of 3 euros per driven kilometre.

Vehicle maintenance and depreciation

Similar to the fuel consumption, maintenance trains are under a very small load during travelling compared to the actual workload during a project. Contractors recon that the wear on parts of the train as a result of normal travelling, is negligible compared to the wear on parts during a project. The same is true for vehicle depreciation. Vehicle maintenance and vehicle depreciation are therefore not taken into account for calculating the cost per travelled kilometre or hour.

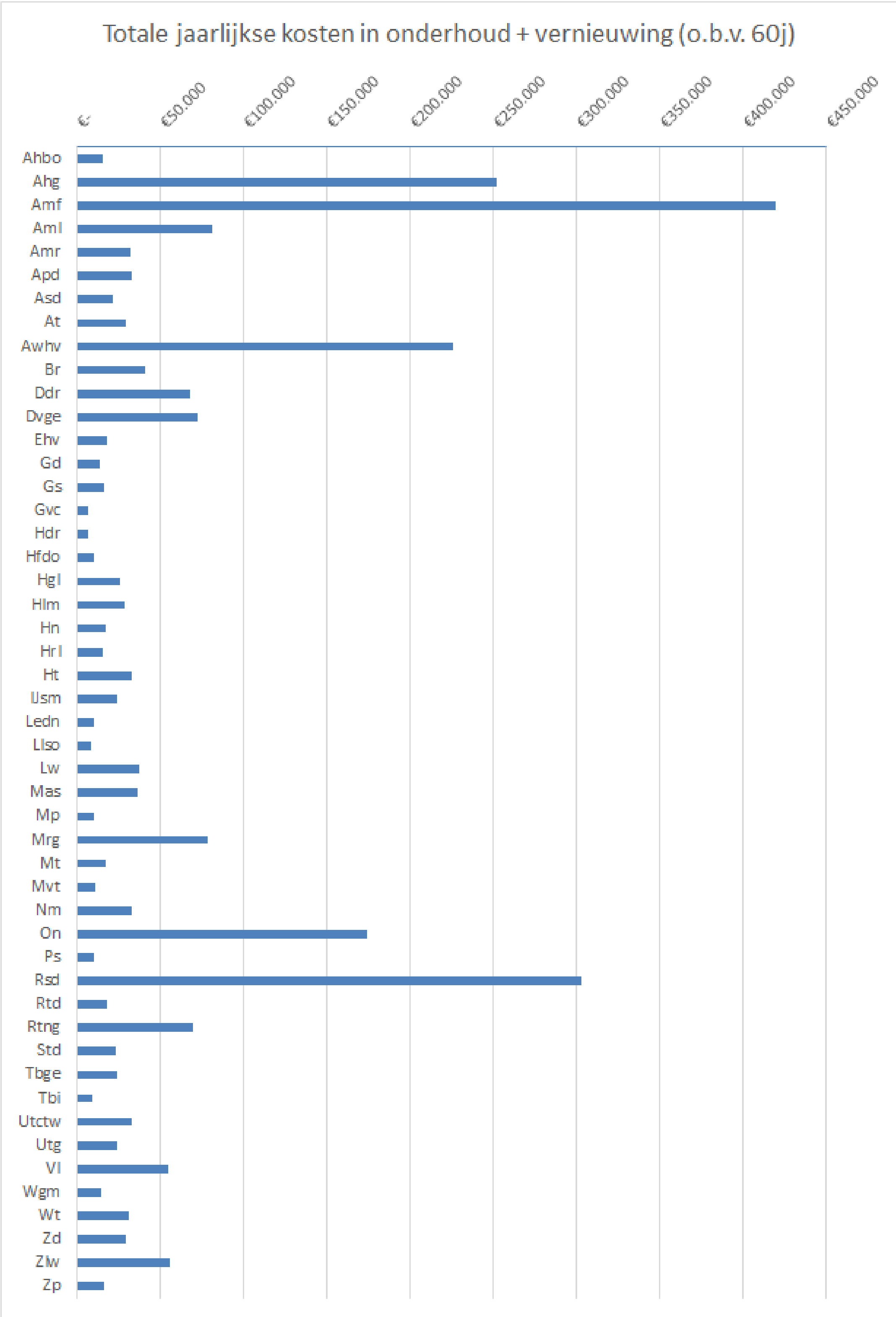


Figure 3.1: Total yearly maintenance cost per parking location

Parking Location	Usable Capacity	Actual Track Length	Number of switches
Ahbo	352	413	2
Ahg	4801	6911	33
Amf	9867	11710	50
Aml	1801	2080	10
Amr	312	434	3
Apd	857	997	4
Asd	157	1330	1
At	472	740	4
Awhv	3859	5918	22
Br	852	1278	5
Ddr	1224	1365	10
Dvge	1569	1881	8
Ehv	129	314	2
Gd	91	270	2
Gs	209	896	1
Gvc	148	147	1
Hdr	112	163	1
Hfdo	189	80	1
Hgl	800	547	3
Hlm	461	648	4
Hn	474	564	2
Hrl	324	434	2
Ht	333	1462	3
IJsm	237	409	3
Ledn	142	441	1
Llso	266	307	1
Lw	870	1041	4
Mas	1191	1331	4
Mp	272	402	1
Mrg	2297	3129	7
Mt	350	563	2
Mvt	109	129	2
Nm	915	710	4
On	5534	6233	18
Ps	71	96	1
Rsd	7332	10050	34
Rtd	325	221	3
Rtng	2186	2393	7
Std	410	699	3
Tbge	506	520	2
Tbi	306	320	1
Utctw	743	607	5
Utg	402	1144	2
Vl	1429	2337	5
Wgm	314	388	2
Wt	904	964	3
Zd	928	705	4
Zlw	1441	1634	7
Zp	476	530	2
Total	59349	77885	302

Table 3.3: Overview of the usable and actual track length and number of switches per DRP.

4

Simulation-Optimisation Methodology

The problem of this research will be tackled using a simulation-optimisation approach. The current set of parking locations will be altered by including and excluding parking locations from every simulation iteration. A Discrete-Event Simulation (DES) and an optimisation stage interact to find a near-optimal solution to the problem. The simulation-optimisation interaction is illustrated in Figure 4.1. The entire framework has been coded using Python.

A DES will simulate a year's worth of maintenance projects and the accompanying journeys made by the maintenance trains. The simulation takes as input a list that specifies which parking locations are included and which are omitted from the system. The KPIs for this research are: the aggregate cost of travel and maintenance of the included parking locations, the number of included parking locations and the available parking capacity. The DES acts as an evaluation function for the optimisation algorithm.

The optimisation algorithm constructs a new solution to the problem, which is used as a new input for the next iteration of the DES. The performance of the suggested solution is evaluated in the optimisation stage. Due to the large size of the solution space the optimisation algorithm will use the heuristic technique SA to search through the solution space. The search algorithm will determine a neighbouring solution, i.e. a list which hold information on which parking location to omit in the next simulation iteration. The neighbouring solution is constructed by applying a small permutation, making it a 'neighbour' of the existing solution.

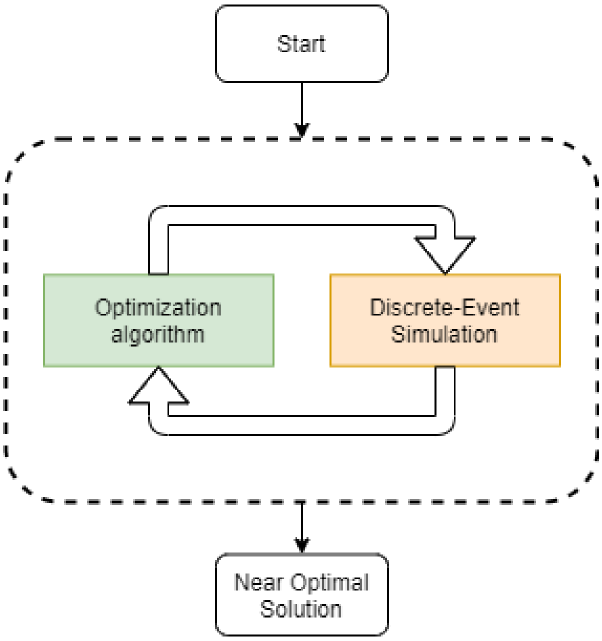


Figure 4.1: Flow diagram of simulation-optimisation interaction

Two different search algorithms are implemented: a random search algorithm (RSA) and an algorithm which takes information from the utility level of the parking locations during the previous simulation iteration into account (ULSA). These search algorithms are compared in order to determine the level of importance of the utility level on the solving process.

4.1. Discrete-Event Simulation

The simulation will be done in chronological order to facilitate the dynamic pseudo random allocation of maintenance trains to parking locations. For this, two python libraries will be used: Salabim and Networkx. Networkx is a network graphing tool with various built in algorithms, e.g. for finding the shortest path, etc. Salabim is an event based programming tool with a basic visualisation option. These will be connected through custom python scripts. The DES consists of two passive components: the railway network and parking locations, and two active components: projects and trains. These components and their relation will be explained below.

4.1.1. Network

The railway network is represented as a graph of nodes and edges. Every DRP in the dutch railway system is represented by a node in the graph. If there is a railway connection between any two DRP's, this connection is represented by an edge in the graph. The length of the edges is taken to be the straight distance in kilometres between the respective nodes. The railway network is illustrated in Figure 4.2.

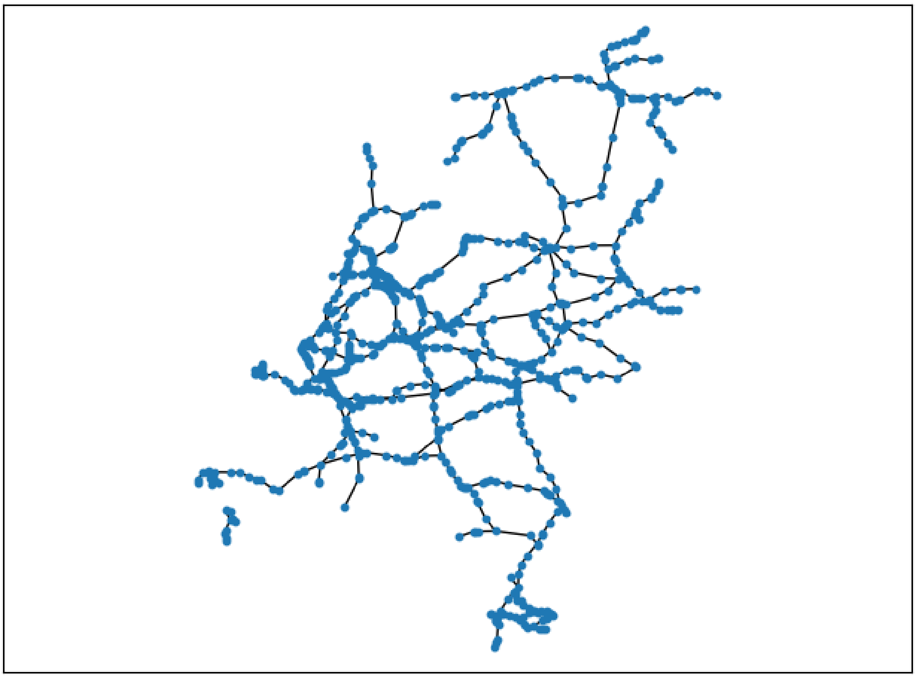


Figure 4.2: Railway network overview.

The capacity of the connections is not taken into consideration, i.e., multiple railway tracks parallel to each other are consolidated into a single edge. The presence of other traffic on tracks is also not taken into account. The amount of train rides (excluding inspection rides) contractors undertake for around 1200 projects a year is negligible compared to the 2.2 million rides undertaken by other trains (Spoorpro, 2020). During journeys for ad hoc maintenance situations, passenger trains that run on a tight schedule generally have priority access to a piece of track over maintenance trains. In practice, this means that contractors have to weave through other traffic. The complicated dynamics involved in this process are better suited to be analysed by follow-up research.

Some projects cause an obstruction on certain edges in the vicinity of the project location. In early experimentation with the simulation, these obstructions were implemented and as a result certain edges would become inaccessible once the corresponding projects were carried out. Other maintenance trains should not be able to use there edges for their shortest paths, and thus, the shortest path search was executed for every journey. The total calculation time (due to the high number of iterations needed for the SA) accumulated to such an extent that it made running a single simulation impractical. Therefore, another solution was implemented: The shortest paths from every DRP to every other DRP was calculated beforehand to drastically decrease the simulation run time. The shortest distance between every DRP was stored in a 807x807 matrix. This meant that the implementation of dynamic obstruction of edges was no longer implemented. The impact of the omission of this dynamic was evaluated by running the simulation without the physical obstruction, but keeping track of which edges should have been obstructed at any time. From this, the probability for detour as result of an obstruction was found to be 4%. The increase of total distance travelled for a simulation run with obstructions compared to a simulation run without obstructions was found to be less than 1% on average. This, together with an hundred fold decrease in simulation run time justified the omission of obstructions in the DES.

4.1.2. Parking Locations

As discussed in Chapter 3, the available parking locations assigned to contractors varies by year. As most information is available for the year 2019, the entire simulation is modelled around the situation of that year. If data for other years is available, it is possible to import this into the simulation. The parking locations used in the simulation are illustrated in Figure 4.3

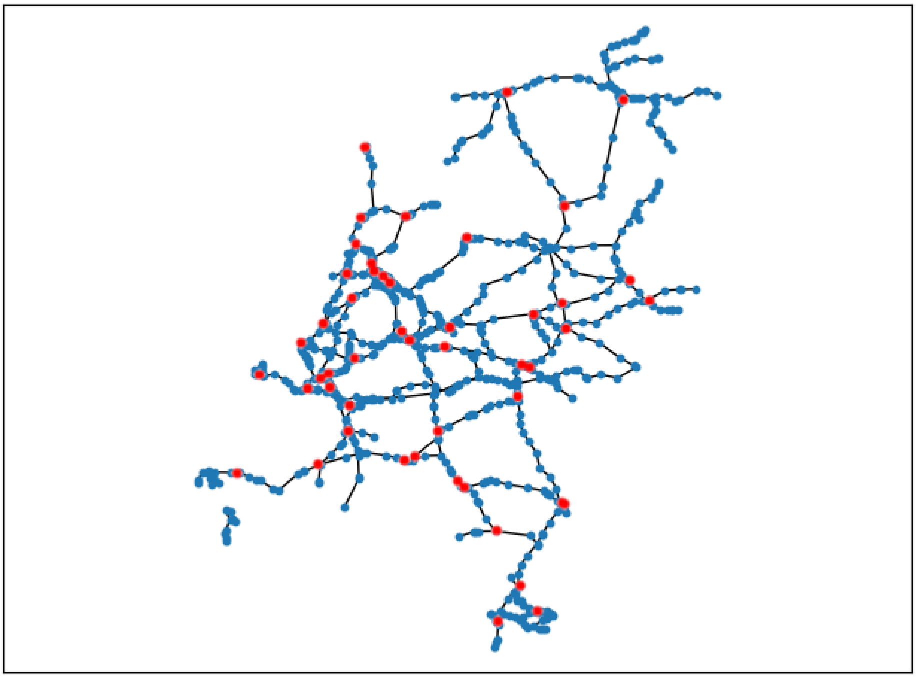


Figure 4.3: Overview of the railway network and parking locations (red).

All parking locations have information on where they are located in the network via the corresponding DRP name. The locations of the parking locations are fixed, but the aspect that changes from iteration to iteration is the inclusion of the parking location in the system.

Per parking parking location, the capacity is distributed over one or more separate tracks. Thus every parking location has a certain amount of parking queues equal to the amount of different tracks that are assigned to the contractors. Each track has its own maximum usable capacity in metres. Based on this usable amount of capacity and the total length of trains parked per location, the free capacity can be calculated at every moment in the simulation. As stated in Chapter 3, contractors need a buffer of at least 100% of the train fleet lengths as free space on parking locations. This results in parking locations being ~50% occupied in order to make shunting possible. In practice, this occupation rate is a preferred upper limit, but amenable to any situation if needed. Trains are of different lengths and any combination of trains therefore rarely fills exactly 50% of a parking location. The summations of train lengths on any particular track may exceed the 50% limit slightly. In the simulation, it is therefore specified that no parking location can be occupied more than 55%, while the total amount of available capacity cannot be lower than 50% of the total train fleet length. This makes it possible to simulate solutions whereby the total available capacity is equal to double the train fleet length, while allowing trains to fit.

For the evaluation of the maintenance costs of the included parking locations in every solution, the

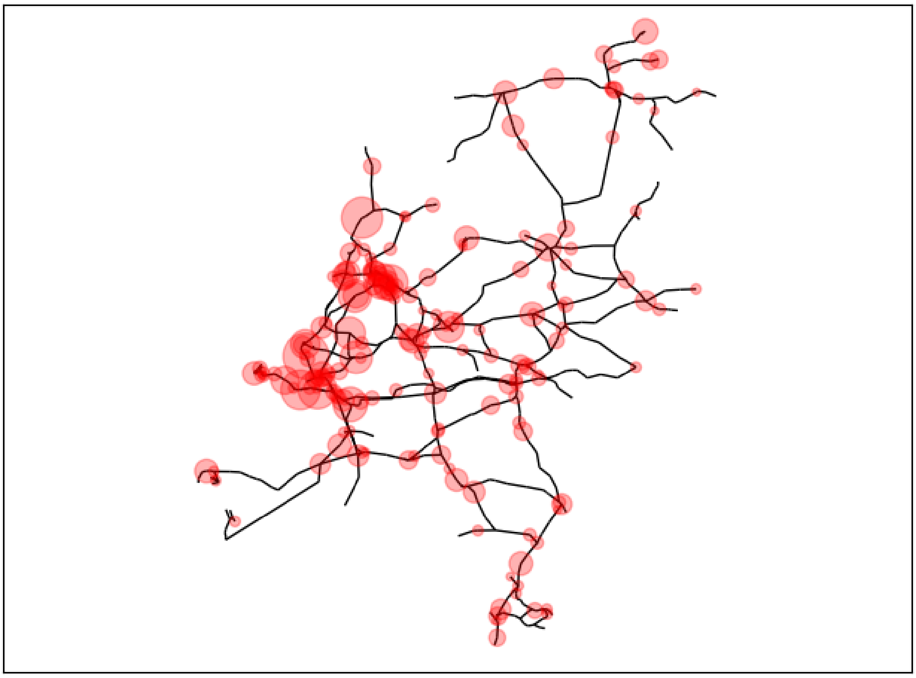


Figure 4.4: Heat-map of project locations.

actual length of track is used. This length differs from the usable capacity. Every parking location has a set amount of actual track length and a set collection of switches, from which the total yearly expenditures for conservation of all the included parking locations can be calculated.

During implementation of the ULSA, the simulation outputs statistics of the parking locations. For every parking location the number of movements (arrivals and departures at a parking location) are stored. From this, the relative usage per km of actual track is calculated and used by the ULSA to construct a neighbouring solution for the next iteration.

4.1.3. Projects

The simulation will use information on the planned maintenance projects in the railway system for one year. There are 1216 projects to be simulated for the year 2019. Projects are an active component in the Salabim simulation, which means that each projects acts via a specific process. The main aspect of the process is activating a train to service the project.

To manage the projects, a project generator is created. The project generator reads the project list and puts the projects in the relevant project-type queues. The projects will enter the simulation, and thus enter the project type queue, a few weeks prior to the start of the project. This emulates the practical allocation of trains to projects. Once a train has been activated, the train-process will guide the rest of the operations needed to service the project. Once a project is finished, the corresponding train is made available again, the project process terminates itself and is thus deleted from the simulation. The information needed to simulate a project consists of a timestamp of when a project starts, the location, the required contractor and the project duration.

4.1.4. Trains

Trains are the second active components in the Salabim simulation. A set of trains to be used in the simulation is created using crude assumptions on the length per train, the number of trains per contractor and the average speed.

A small portion of the maintenance trains in use is known. The average lengths of five different train types in this dataset are: Flat Wagon (10m), locomotive (15m) small milling train (30m), large milling train (62m) and video inspection train (28m). The quantity of trains per type was not known. Based on contractor experience, it is assumed that flat wagons are connected in threesome and are always driven by a locomotive resulting in an total unit-length of 45m. From this small dataset, the average train length is found to be 42m. The total length of the maintenance train fleet is set to be 22km. From this, it is calculated that there are 524 train units to be generated. To assign the length to every train unit, a uniform distribution between 22 and 62 metres is applied.
Because the total amount of train types that are used or parked by the contractors is unknown, it is decided to link trains to projects based on the train's contractor. While the individual train fleets per contractors is also unknown, the estimated total length of trains per contractor is documented and is shown in Table 3.1. The share of train lengths is used to randomly assign contractor labels to the individual trains in the fictional train fleet. Maintenance trains generally reach speeds of up to 80km/h. However, due to intermittent stops, direction changes and run-up time, it is decided to use an average travelling speed of 40 km/h for every train in the system.

The train data are stored in an external file and imported at the start of the simulation run. During the initialisation, the trains are placed in the relevant contractor queue. This queue is no physical queue, but act as a way to determine which trains are available to service a project. Afterwards, trains are also distributed over the included parking locations at random.

A flowchart of the process of the train component is given in Figure 4.5. Once a train has entered the relevant contractor queue, it will wait for an available project. The project component activates an available train when it enters the system. Once the train is activated, it claims the project for itself, leaves the contractor queue and reads the project location *C* and the starting date. The train checks if it is parked at a parking station or not. If the train is at a parking station, it waits until three to one days prior to the project starting date. When this moment has arrived, the train searches for a parking location *B* with available space nearest to the project location. If the distance from its current location *A* to *C* is smaller than the total distance for the journey from *A* to *B* to *C*, then the train does not travel to *C*, to minimise the distance travelled. Once at *B*, the train combines its average speed and the distance to travel plus a buffer time period and compares it to the starting moment of the project to calculate its departure moment from the parking location. The train is idle for the duration of the project. When the project is finished, the train enters the back of the contractor queue, checks if there is room at the last parking location *B* it has visited and travels to it if there is. Otherwise, the train will check if there is an unfulfilled project available. If a project is available, it will claim that project and continues the process from there. Otherwise, it will look for a parking location with available space nearest to its current location.

Whenever a train moves, the travelled distance is stored for that particular train. From this, after one iteration, the total distance travelled and the resulting travel costs can be determined.

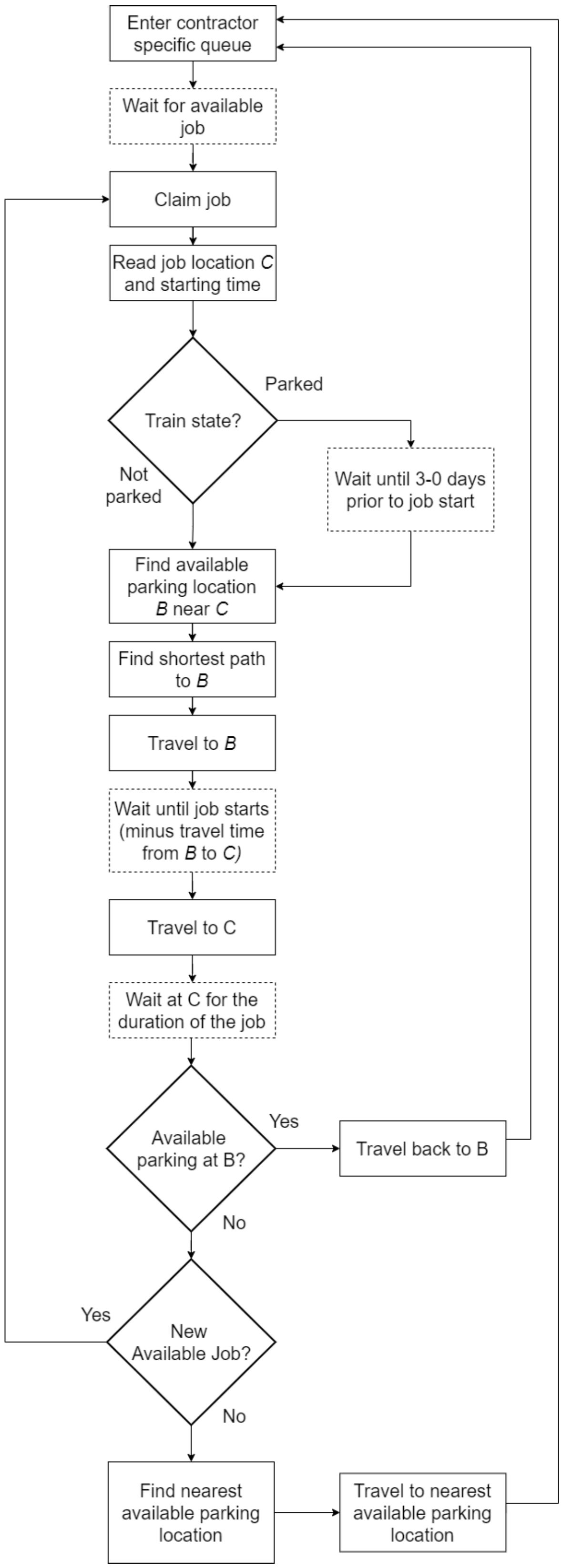


Figure 4.5: Flow diagram of the train process during the simulation.

4.2. Optimisation Algorithm

As stated before, the simulation acts as an evaluation function for the optimisation algorithm. This optimisation algorithm will search for a near optimal solution within the solution space. The solution space consists of all possible combinations of parking locations in the system, whereby a parking location is either included or not included. Currently, 49 different parking locations have a total usable capacity of 59km, bringing the average parking capacity per location to 1.2 km. As stated in Chapter 3, the minimum capacity is set to be around 44km. Based on these numbers, one could estimate that roughly 36 out of 49 parking locations should be included. To choose 36 locations out of a set of 49, brings the resulting solution space to $2.6 * 10^{11}$ possibilities. Thus, the optimisation algorithm should be designed in such a way that it can find a near optimal solution without having to evaluate every single solution.

The optimisation algorithm has two tasks: evaluating the output from the simulation and constructing a new solution to the problem for the next iteration of the simulation. The simulation and optimisation interact to find a near-optimal solution to the problem. Due to the large size of the solution space the optimisation algorithm will use the heuristic technique SA to search through the solution space. The search algorithm will determine a neighbouring solution, i.e. a list which holds information on which parking location to include or exclude during the next simulation iteration through the use of a neighbourhood function.

4.2.1. Neighbourhood Function

The neighbourhood function determines the way the optimisation algorithm walks through the solution space. For this research, two different search algorithms are implemented: a Random Search Algorithm (RSA) and a Utility Level Search Algorithm (ULSA), which takes information from the utility level of the parking locations during the previous simulation iteration. These search algorithms are compared in order to determine the level of importance of the utility level on the solving process.

Random Search Algorithm (RSA)

The neighbouring function of the RSA can execute one of the following operations.

1. If the number of included parking locations does not exceed the maximum number of included parking locations, then, from the previously evaluated solution, an excluded parking location is randomly selected and included in the candidate solution.

2. If the total parking capacity of all parking locations is not lower than the minimum parking capacity, then, from the previously evaluated solution, an included parking location is randomly selected and excluded in the candidate solution.

3. From the previously evaluated solution, one excluded parking location is included in the candidate solution and one included parking location is excluded in the candidate solution. The conditions of operation 1 and 2 both apply to this operation.

Only one of these operations can be executed per iteration and every operation has an equal chance of being executed. Figure 4.6 summarises the concept of the RSA.

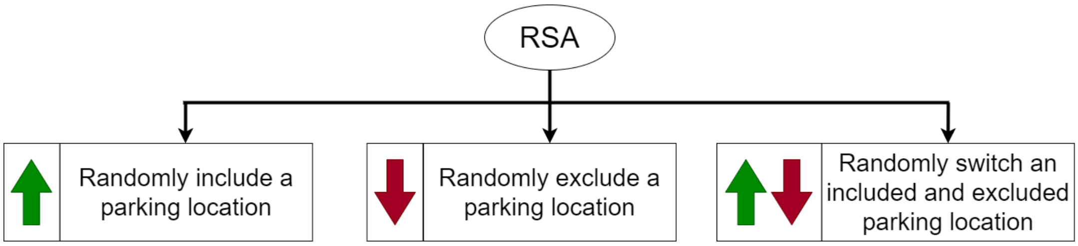


Figure 4.6: Decision tree for the RSA.

Utility Level Search Algorithm (ULSA)
The neighbouring function of the ULSA can execute one of the following operations.

1. If the number of included parking locations does not exceed the maximum number of included parking locations, then, from the previously evaluated solution, the most heavily used parking location is found. An excluded parking location, which lies closest to the most used parking location, is selected and included in the candidate solution.

2. If the total parking capacity of all parking locations is not lower than the minimum parking capacity, then, from the previously evaluated solution, the least used parking location is selected and excluded from the candidate solution.

3. From the previously evaluated solution, an excluded parking location, closest to the most used parking location is included in the candidate solution and the least used parking location is excluded in the candidate solution. The conditions of operation 1 and 2 both apply to this operation.

Similar to the RSA, in the ULSA only one operation is randomly executed at every iteration. Figure 4.7 summarises the concept of the ULSA.

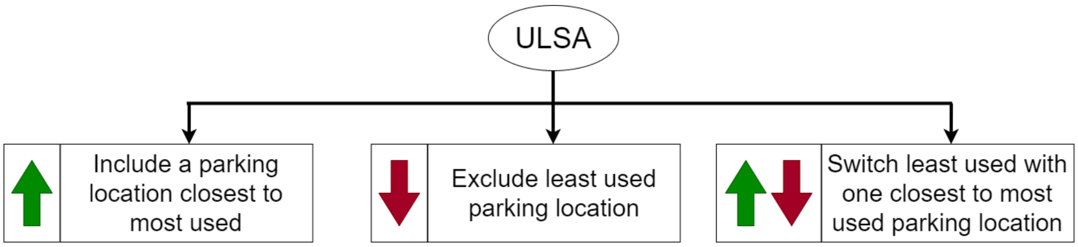


Figure 4.7: Decision tree for the ULSA.

The usage level is evaluated based on the amount of departures and arrivals of a parking location, as well as its capacity. A parking location with a relatively high number of departures and arrivals compared to its capacity is therefore assumed to lay in a strategically beneficial location within the network.

4.2.2. Simulated Annealing

After a neighbouring solution is generated by the neighbourhood function, the DES evaluates the performance of that solution. The performance indicator of that solution is used in the SA process. The SA process, accepts better and worse solutions to the problem with a certain probability. This probability is a function of the cost difference between the candidate solution and the intermediate accepted solution and the temperature:

$$E = exp\left(\frac{-\Delta_{cost}}{T(n)}\right). \tag{4.1}$$

The temperature lowers according to a cooling schedule. The most common cooling schedule follows an exponential decay curve. The shape of this curve is determined by the starting temperature T_0, the cooling factor α and the cooling step n.

$$T(n) = T_0\alpha^n. \tag{4.2}$$

k_{max} is the maximum number of iterations at each temperature value. Usually, k_{max} is constant throughout the cooling process. However, Frausto-Solis et al. (2007) applies a variable number of iterations per temperature cycle with improved performance. In this case, k is equal to k_0 at the start of the cooling schedule and increases to k_{max} at $n = N$, according to:

$$k(n) = k_0\beta^n, \tag{4.3}$$

where β is the increment coefficient and is greater than one. β needs to be tuned in order to allow k to be equal to k_{max} at $n = N$, according to::

$$\beta = exp\left(\frac{lnk_{max} - lnk_0}{N}\right). \tag{4.4}$$

N is the maximum number of cooling steps and is a result of the initial and final temperatures and the cooling factor. N can be calculated through:

$$N = \frac{lnT_f - lnT_0}{ln\alpha}. \tag{4.5}$$

In literature, attempts have been made to define values for the SA parameters, based on the problem characteristics. In practice, one would assume some values and tweak them to find parameter values that are feasible to work with. Frausto-Solis et al. (2007) published an approach for tuning the SA parameters. Qin (2012) applies a simulated annealing method for the FLP to various instances of different sizes, i.e., the number of facilities to be located. The range of sizes of these instances match the order of magnitude of the number of parking locations used in this research. In this research, the approached proposed by Frausto-Solis et al. (2007) is used in combination with the parameter values used by Qin (2012). These values are tweaked in order to achieve the following demands:

1. Calculation time of a single Sim-Opt run is no longer than 1.5 hours
2. The energy curve starts off at a value near 1 and ends near 0 (Frausto-Solis et al., 2007)
3. The cooling factor lies between 0.7 and 0.99 (Kirkpatrick et al., 1983)
4. The number of iterations lies in the order of 2000 (Qin, 2012)

Through experimentation, it is found that a single simulation (i.e. one iteration of the Sim-Opt) takes on average about 2.5 seconds to run. Qin (2012) found robust results for a similar size problem with

2000 iterations of the SA optimisation. For this research, the same amount of iterations is desired in order to satisfy the first demand.

The initial and final temperatures can be expressed as a function of the minimum Z_{min} and maximum deterioration Z_{max} and the probability of accepting them (see Eq. 4.6 and 4.7). The minimum or maximum possible deterioration has no exact value, since the inclusion and exclusion of parking locations is expected to have an unknown negative effect on the travel costs. Therefore, the possible deterioration is based on the cost increase of including the least- and most expensive parking location. Making $Z_{min} \approx 10.000$ and $Z_{max} \approx 420.000$. For the SA process to function correctly, the acceptance probability should be near 1 at the start of the SA process and near 0 at the end of the SA process. The acceptance probability at the start is taken to be 0.999 and the acceptance probability at the last iteration is taken to be 0.001 (Frausto-Solis et al., 2007). By applying Equation 4.6 and 4.7, T_0 was found to be $4.2e8$ and T_f is set to $2e3$. The resulting T_0 of is lowered to $3e8$, since experimenting shows that the energy function remained high for a disproportionately amount of iterations. Through trial and error, T_0 is lowered to $3e8$.

$$T_0 = \frac{-\Delta Z_{max}}{ln(P(\Delta Z_{max}))} \tag{4.6}$$

$$T_f = \frac{-\Delta Z_{min}}{ln(P(\Delta Z_{min}))} \tag{4.7}$$

Qin (2012) sets the maximum number of iterations per temperature cycle k_{max} equal to the total number of facilities in the underlying problem: $k_{max} = 49$. According to Frausto-Solis et al. (2007), the number of iterations in the first temperature cycle k_0 is a ten fold smaller than k_{max}: $k_0 = 5$.

Equations 4.8 - 4.10 are used to calculate the number of temperature cycles and the resulting number of iterations. Frausto-Solis et al. (2007) gives no analytical approach to determine the cooling factor. The cooling factor α and the increment coefficient β are tuned in order to keep the the total number of iterations around 2000. Table 4.1 gives an overview of the SA parameter values used in this research. Equations 4.8 - 4.10 show how these values work together in order to satisfy the demands stated above.

$$n = \frac{ln(2e3) - ln(3e8)}{ln0.90} = 113 \; cooling \; steps \tag{4.8}$$

$$\beta = exp\Big(\frac{ln49 - ln5}{113}\Big) = 1.02 \tag{4.9}$$

The number of iterations can be calculated by integrating k_{max} as a function of the cooling schedule:

$$\int_0^{113} k_0 * \mathrm{B}^n \, dn = \int_0^{113} 5 * 1.02^n \, dn = 2114 \; Iterations \tag{4.10}$$

The number of iterations can deviate from the calculated answer from Equation 4.10, due to the rounding errors of β and k in the program. Nevertheless, this deviation will be small and is still a good indication of the total amount of iterations.

Parameter	Value
T_1	3e8
T_f	2e3
α	0.90
β	1.02
k_0	5
k_{max}	49

Table 4.1: Overview of the Simulated annealing parameter values used for theRSA and ULSA.

4.3. Simulation Verification

The following sections will present the verification and validation of the simulation. Afterwards, the results of the experiments will be presented. The simulations for the verification and validation are run using the scenario described in Chapter 3. This scenario is based on historical data of the year 2019 with 1216 projects and all possible parking locations are included. This setting is also used in the final experiments. The simulations for the verification and validation parts are based on these input parameters, and altered where specified.

To verify the simulation a dynamic testing approach (Sargent, 2011) will be used; the output will be examined for reasonableness under a variety of settings of the input parameters. The aspects that are tested include: a check that the maximum parking capacity is not violated, a check to see if the maximum number of expected movements is not exceeded and an evaluation of the total distances travelled by all the trains in the system.

4.3.1. Parking Capacity

When investigating the needed parking capacity in the system, is it necessary to check if the maximum parking capacity is not violated. The total usable parking capacity, as stated in Chapter 4, is 59349m. For the average train length (42m) calculated in Chapter 3, the total amount of trains that could be stationed if 55% of the usable track capacity could be filled, would be 777. However, the train lengths in the simulation are randomly chosen between 22m and 62m and on top of that, randomly allocated over the parking locations, making an occupancy rate of exactly 55% highly infeasible. In order to check if the maximum parking capacity is not violated, the number of trains in the system is varied. Table 4.2 shows five test wherein the maximum possible number of trains in the simulation is found, together with the calculated maximum number of trains based on an average train length of 42m. The maximum number of trains in the simulations never exceed the theoretical maximum.

Available capacity	Max number of trains (calculated)	Max number of trains (simulated)
59437	778	750
44681	585	565
51131	669	656
53611	702	691
57629	754	732

Table 4.2: Verification of the maximum number of trains allowed in the system.

4.3.2. Number of Movements

The fictional train fleet that is created for this research consists of 524 trains and there are 1216 projects to be serviced. During the simulation, there are at maximum three movements possible per project: one journey to a parking location close to the project location, one journey from that parking location to the project location and one journey from the project location to a free parking location. The maximum number of movements should therefore not be able to exceed a threefold of the number of projects. In Table 4.3, an overview of a few simulations are shown, whereby the number of projects is altered. Table 4.3 shows that the maximum number of movements is not exceeded. The number of movements in the simulation is not equal to the calculated number of movements. When a project is assigned to a train in the simulation, there is a small chance for that train to already be parked at a location closest to a project location. This means that the train will not undertake a journey to another parking location before going to the project location.

Number of projects	Max number of movements (calculated)	Max number of movements (simulated)
0	0	0
341	1023	1016
655	1965	1951
902	2706	2688
1216	3648	3623

Table 4.3: Verification of the number of train movements.

4.3.3. Distance Travelled

In the simulation, trains enter the back of the waiting queue after a project is finished. As a result, projects will be serviced by as many trains as possible and not by a small portion of trains only, while another portion of the trains in the system would stay idle for the duration of the simulation. The distribution of total distance travelled by every train gives insight to this effect. A few scenarios wherein the amount of trains in the system is changed are simulated. The resulting distributions of travelled distances for the trains in the system are compared and evaluated against logical predictions.

If there is a rotation in the assignment of trains to the projects, one would expect a distribution for the amount of kilometres driven by the trains in the system. Figure 4.8 shows the distributions of the total travelled distances for various numbers of trains in the system. Histograms *A*, shows that a relatively large part of the trains travel in total between 50 and 400 km. A very low number of trains (2) travel long distances of over 600km, with a maximum of 675km. Only a few trains (5) are never or rarely used and travel between 0 and 25km. Incidentally, this suggests that the estimated number of trains of 524 in the real world system (see Chapter 3) is a reasonable amount. Contractors would aim to have the least amount of trains in their portfolio to service all the projects and thus the amount of trains that would travel rarely should be low. Such a desired effect can be seen in histogram *A*.

Histogram *B* shows a simulation with 800 trains in the system. When more trains are added, the amount of projects per train should drop. With less projects to service for any one train, the average total distance travelled per train should also drop, as can be seen in Figure 4.8. Furthermore, trains that would be under high workload (>400km) in scenario *A*, would experience a drop in workload and as a result there will be less trains with a high yearly mileage. Histogram *B*, shows that the distribution is skewed more to the left in comparison to histogram *A* and that no train drives more than 550km.

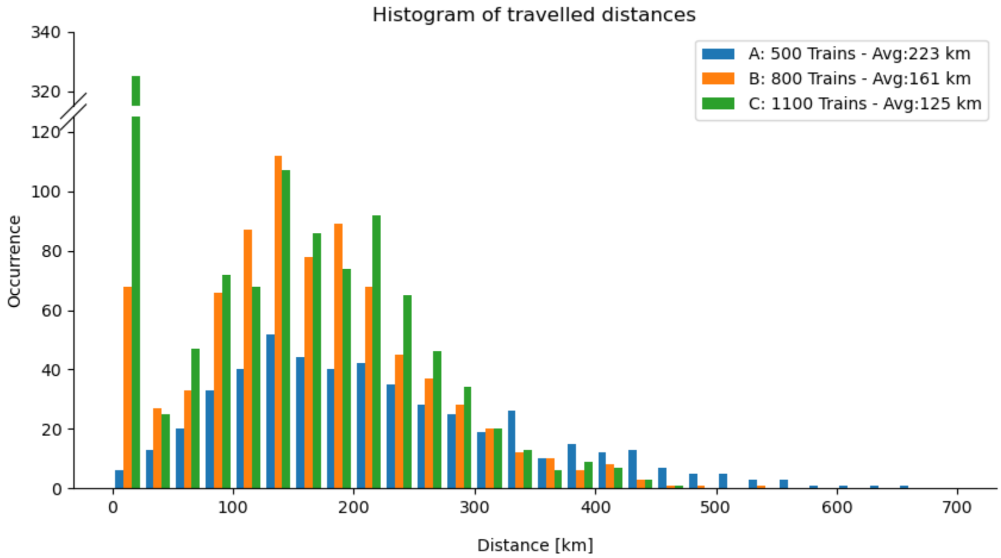


Figure 4.8: Histograms of travelled distances for three simulation scenarios.

Lastly, the amount of trains that are not summoned for project should increase, because there were already enough trains for all the projects in scenario *A* and the number of projects has not changed. This effect can be clearly seen in histogram *B*, as the number of trains that drive between 0 and 25km increased to 69.

When the amount of trains increases even more (1100) in scenario *C*, it can be seen that the number of trains under high workload (>400km) has decreased again. Also, the maximum distance travelled has decreased, as no train has driven more than 475km. This confirms the expected dynamic explained in the previous paragraph. The bulk of the distribution has not changed significantly with respect to scenario *B*. A logical explanation of this would be that the projects are assigned to trains that have been randomly placed in a waiting line and randomly allocated over the included parking locations in the system. With this randomness and such a large number of trains, a normal-shaped distribution of travelled distances is not remarkable. The bulk of histogram *B* and *C* both show a similar distribution with a comparable amplitude, indicating that a similar amount of trains is used to service the projects. The amount of trains that drive between 0 and 25km has increased by 267, while the amount of trains in the system has increased by 300. This shows that there is no need for the extra trains in scenario *C*.

The evaluations described above, together with a detailed examination of the simulation trace, verifies the correct implementation of the conceptual model into the simulation.

4.4. Simulation Validation

A validation of the simulation will be performed to compare the representation of a conceptual model to the real system. This will be done through the evaluation of three aspects of the simulation:

- Total expenditures (Travel and maintenance)
- Distance travelled by trains (individually and total)
- Paths that have been driven (simulation and in real life)

4.4.1. Total Expenditures

The lack of real world data available, and consequently, the artificial generation of the data, makes it difficult to validate the simulation. As a result, the validation will be performed using expert judgement. The numbers used as inputs in this study are already based on expert judgement. Nevertheless, the total expenditures is evaluated by experts. For the simulation, the yearly maintenance costs is determined by combining the maintenance cost per kilometre of track and switch with the data on the infrastructure elements present in the system. This total amounts to 2.65 million euros when all parking locations are included.

This amount is assessed by experts in the field and acknowledged as a lower limit. The expected yearly infrastructure maintenance cost is estimated to be between 3.4 and 3.8 million euros. The facilities on parking locations are also taken into account in this estimation. This research does not take into account other costs and the total costs thus lie within the same order of magnitude as the real world system.

4.4.2. Distance Travelled

ProRail has data available on train movements per train type for the year 2019. This data holds no information on the individual trains that perform the movements. The dataset is filtered to include the relevant contractors for this research and exclude the measurement trains. The resulting data gives the start- and endpoints of 2780 individual train movements and the date on which a movement occurred. The amount of train movements has the same order of magnitude as the number of train movements in the benchmark simulation scenario. The benchmark simulation scenario shows around 3000 train movements. The train start- and endpoint of the movements in the dataset is used to calculate the total distance travelled. In the real world system the 2870 movements resulted in a total travelled distance of 113500 km. In total, 112000 km is travelled during the benchmark simulation.

In the real world system, trains travel on average more kilometres per movement compared to the simulation. This can be explained by the fact that simulation does not deal with dedicated parking locations for certain train types and contractors. During the simulation, all trains search for the closest available parking location after a project and thus tend to drive shorter distances. Nevertheless, the number of movements and the therefrom calculated total distances travelled are similar to each other and confirms this part of the validation.

4.4.3. Travelled Paths

In addition to the check on the total distance travelled, a check on the reasonableness of the travelled paths in the network is done. Figure 4.9 shows a heat-map of how many times tracks have been used by maintenance trains during a simulation. This heat-map matches the projects locations shown in Figure 4.4. Figure 4.10 shows a heat-map of how many times tracks have been used by maintenance trains in the real world according to the filtered dataset discussed in the previous section. Figure 4.9 and 4.10 are not equal to each other, because the locations where all the individual trains park is different. Both figures do show similarities. There are tracks that are used very often and tracks that are used rarely. Furthermore, the tracks that are used more often lie in the centre of the network and tracks that are used less often lie on the outside (dead-end tracks are used the least).

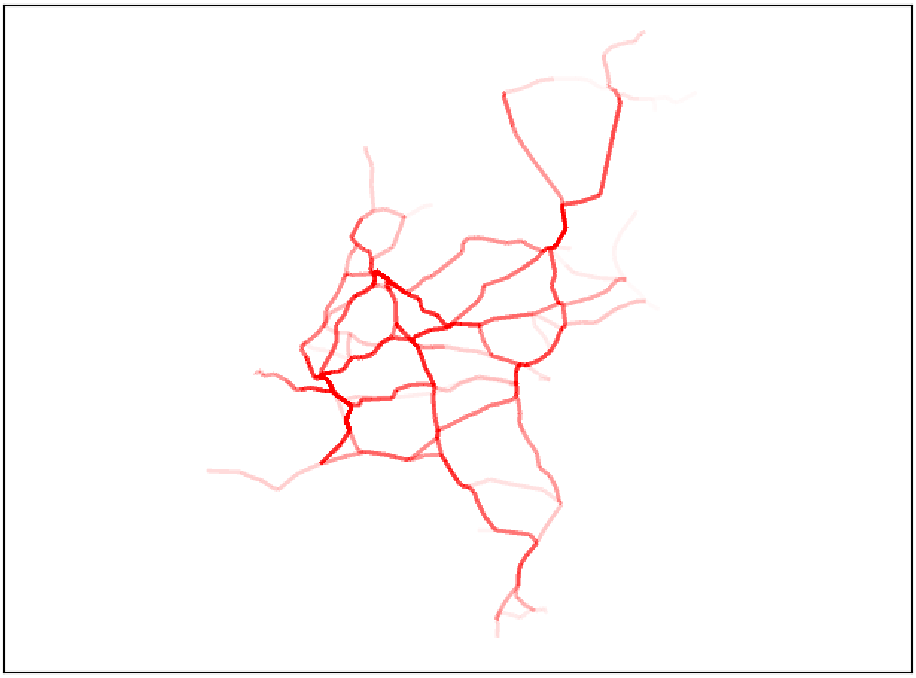


Figure 4.9: Normalised heat-map of travelled edges in the simulation.

Figure 4.10: Normalised heat-map of travelled edges in the real world.

5

Experimental Results

This chapter will give an explanation of the experiments that are performed, followed by an overview of the results. The results are divided into several experiments and will initially be evaluated separately. This chapter ends with an evaluation of the results for all experiments combined.

5.1. Experiments

Figure 5.1 shows an overview of the experiments in this research. The experiments are divided into two categories: Sim-Opt's using the RSA and Sim-Opt's using the ULSA. Per search algorithm, there are two kinds of initial solutions from which the optimisation will start the iteration process. Since this research is based on a case study, it is interesting to see the optimisation behaviour using the current real world situation as a starting point (Experiment A). This is especially relevant to the ULSA, since the utilisation level of the current real world situation provides a practical starting point. On the other hand, starting the Sim-Opt from the same point in the solution space may restrict the reachable solution area. Because there are a limited number of iterations, a limited amount of permutations is possible, restricting the possibility to reach certain solutions. To increase the reachable solution area, the Sim-Opt will be started from multiple initial solutions, wherein a random number of parking locations (while respecting the minimum capacity needed) will be excluded (Experiment B).

Preferably, one would not rely on the results of one simulation run. Therefore, multiple instances of the Sim-Opt will be run per experiment, all with different random seeds. This ensures that the different starting solutions exclude special cases which might hinder the search. Per search algorithm and experiment, 10 simulation runs with different random seeds will be performed. More runs are preferable to obtain more robust results, but considering the computation time per simulation, this is set to be a workable, yet insightful amount.

Both the RSA and ULSA will be tested. The aim is to obtain results to see if the algorithms come to similar results. Due to the large solution space, exact similar results, in terms of the individual parking locations that are included, are welcome, but not expected. Instead, the total expenditures, capacity and the number of included parking locations per solution will be of interest.

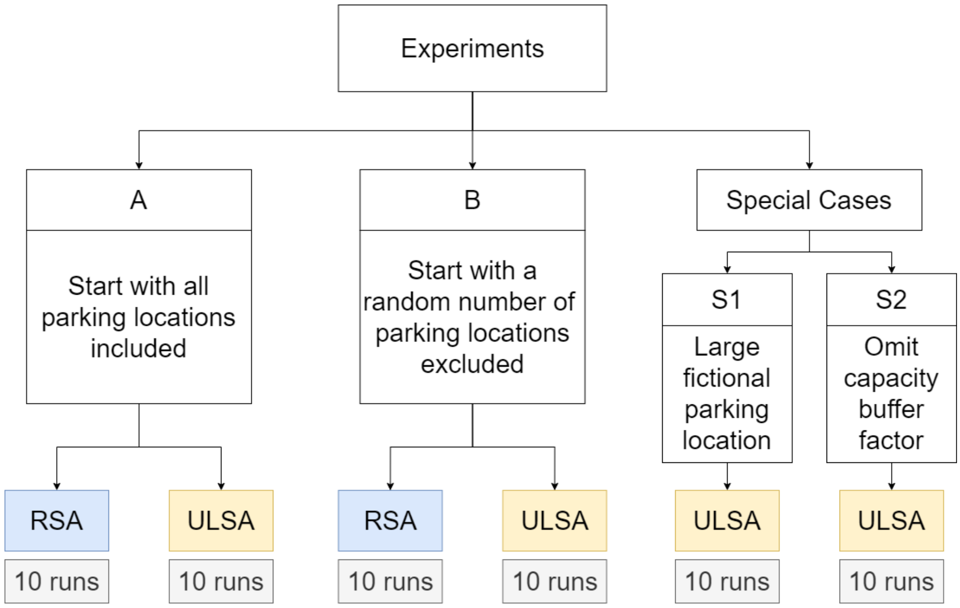


Figure 5.1: Graphical representation of the experiments in this research.

Besides the evaluation of the solution, the different search algorithms will be compared. It is interesting to see if one search algorithm converges faster than the other. These search algorithms are further compared in order to determine the level of importance of the usage level search metric.

There are two special cases that will be simulated in addition to Experiment A and B. These experiments are run using the ULSA, since this neighbourhood search algorithm performed better than the RSA during Experiment A and B.

In one special case (S1), a fictional parking location is added in the approximated centre of the network. This fictional parking location has a capacity of 44620m (the length of the train fleet of 22310m, plus the buffer of 100%). The cost of this parking location is determined by scaling the cost of the largest real parking location (Amf), based on the capacity. The yearly maintenance costs of this fictional parking location amounts to 1.83 mln euros. This fictional parking location is located close to what can be considered the centre of the railway network: Utrecht (*Ut*). The closest DRP to *Ut* is *Utvr* and thus the fictional parking location will be located (and named) *Utvr*. This scenario is simulated in order to assess the performance of a single parking location layout.

In the second special case (S2), the capacity-buffer is removed. The system has the same amount of trains (524) with a combined length of 22310m, but in this scenario there is no need to have a parking capacity buffer of 100%. The maximum capacity at the individual parking locations of 55% of the usable length is also removed. Simulating this scenario will hopefully give insight into the impact of the buffer factor on the equilibrium between travel and maintenance costs of the parking locations in the system. Again, 10 instances of this case are simulated and optimised.

5.2. Results

Before the experimental results are presented, the current system is simulated. The current system consists of all 49 parking locations being available for the parking of maintenance trains. The movements of 524 trains to and from 1216 projects are simulated. The mean values for the KPI's of 10 instances of this simulation are shows in Table 5.1.

Table 5.1: Mean system performance of the current real life scenario, based on 10 simulation runs.

KPI	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
Mean	59349	49	0.983	2.65	3.64

The results of the experiments are visualised in various figures. The numerical data of the best solutions that are found per experiment are given in Appendix A. The mean values of all the Sim-Opt instances per experiment are consolidated in Table 5.3.

5.2.1. Experiment A

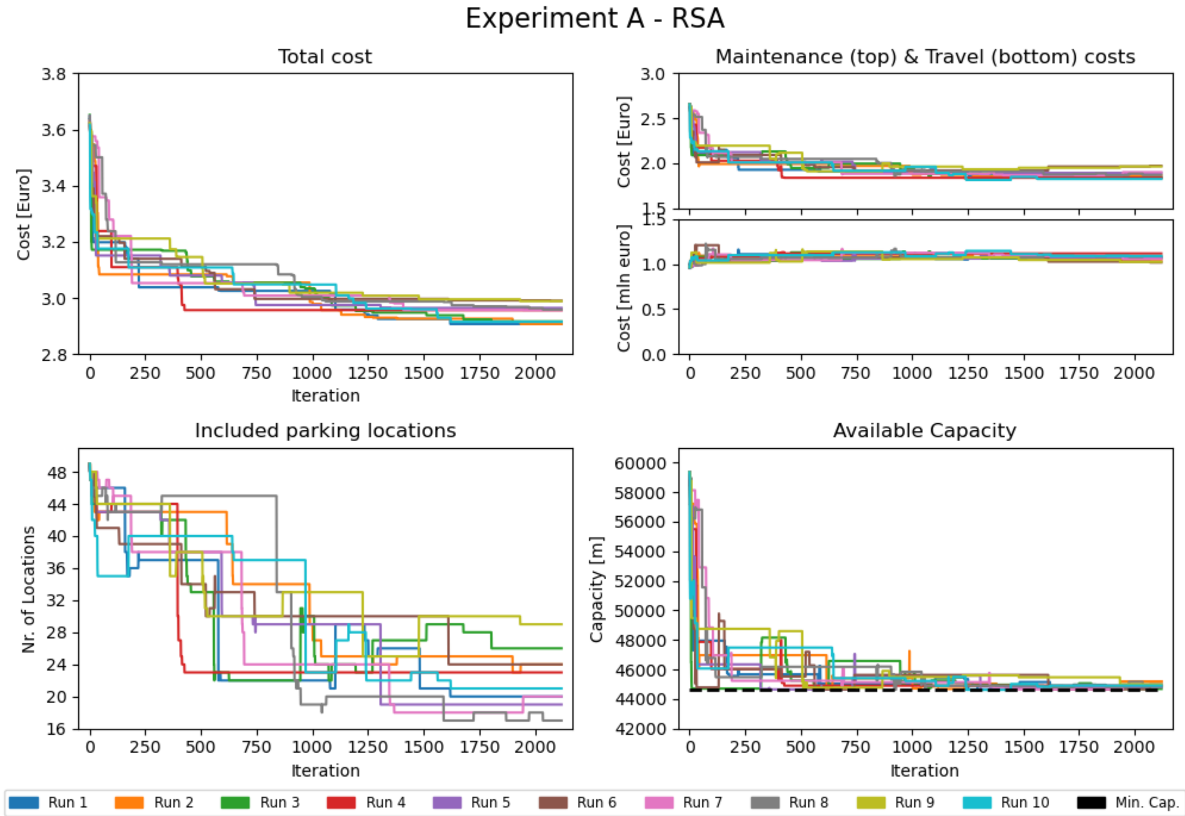
Figure 5.2a shows a set of graphs for the Sim-Opts using the RSA. These graphs show the progress of every instance for experiments A. The total cost for the rolling best solution is plotted in the top left graph. The total cost is a result of the sum of maintenance and travel costs, which are plotted together in the top right graph. The total costs decrease rapidly at the start and this decrease slows as the number of iterations increases. The spread between all the instances is relatively tight over the full length of the Sim-Opts and all final best solutions range between 2.9 and 3.0 million euros. The maintenance costs for all instances follow a similar curve as the total cost. This is in contrast with the travel costs, which show a slight increase at the start of the Sim-Opts, but remain largely constant for the rest of the iterations. The increase in travel costs is low compared to the reduction in maintenance cost and as a result the reduction in maintenance cost has the largest impact on the improvement of the total cost.

In experiment A all parking locations are included at the start of all instances of the Sim-Opt, which corresponds to ~59km of available capacity. The number of included parking locations decreases as the Sim-Opts progress, but the spread of included parking locations between the different instances remains relatively large. The available capacity shows a steep decrease at the start of the Sim-Opts and best solutions are accepted near the minimum capacity boundary of 44620m for the rest of the iterations. Solutions with a higher capacity are accepted occasionally (a result of the SA process), but better solutions (where the total costs is lower) tend to correspond with an available capacity very close to the minimum capacity required.

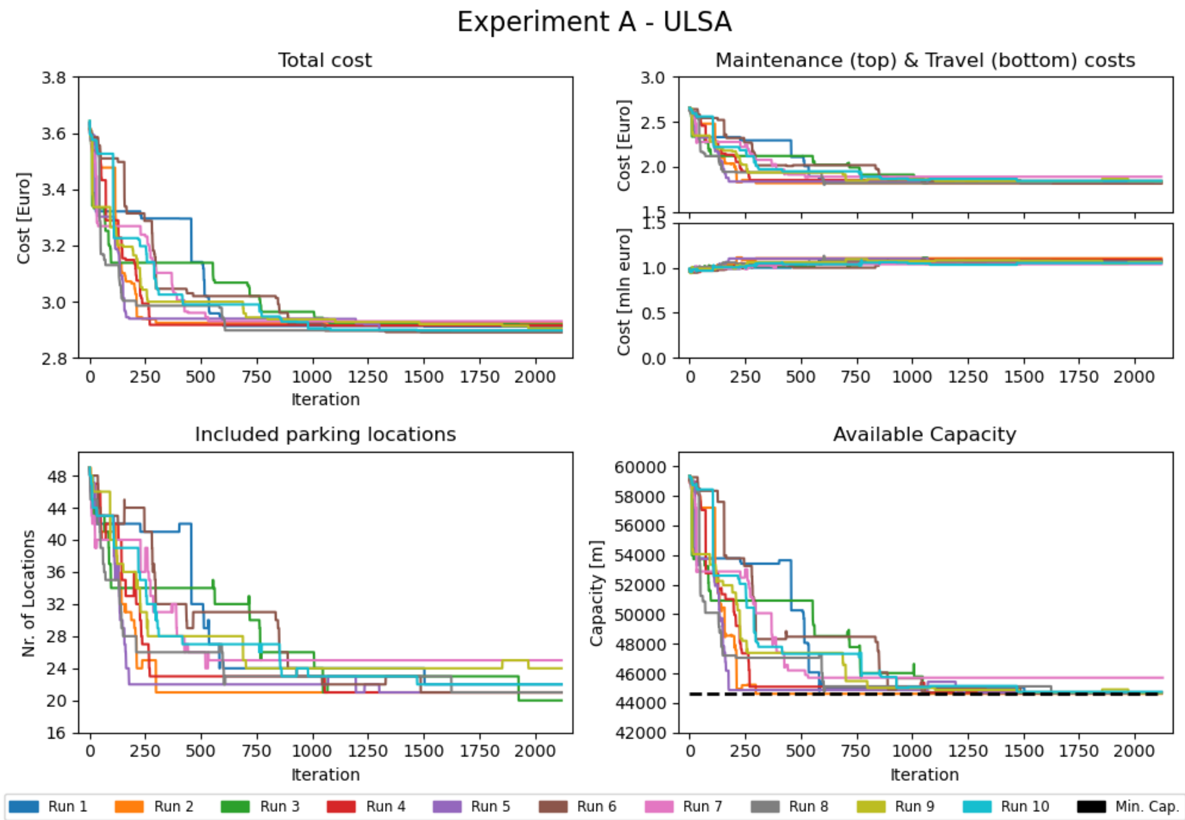
Figure 5.2a shows a set of plots for the Sim-Opts using the ULSA. The trends in every graph are similar to the trends of the RSA; the maintenance costs decrease more than the travel costs increase, the number of included parking locations decreases and the capacity of the best solutions end up near the minimum capacity needed to house all trains plus the buffer size of 100%. However, there are some notable differences.

In contrast to the RSA, the ULSA total cost graphs show a less vigorous decrease in the first 100 iterations. Then, from 100-1000 iterations the spread in the best solutions for all 10 instances is greater. After 1000 iterations, the spread is noticeably tighter during ULSA compared to the graphs of the total cost during the RSA. These characteristics are also observed in the graphs for the maintenance costs. The number of included parking locations shows a steeper decline during ULSA than during RSA. Furthermore, the spread is noticeably tighter in the second half of the Sim-Opts. The available capacity for all ULSA instances show a slower decline when compared to the RSA Sim-Opts. The spread in available capacity between RSA and ULSA is comparable When the SA process nears its end.

The best solution of every Sim-Opt instance hold information on which parking locations are included or excluded. The best solutions of all 10 instances of the Sim-Opts for experiment A are combined for



(a) Progress of all KPI's for all 10 Sim-Opt runs in Experiment A using RSA



(b) Progress of all KPI's for all 10 Sim-Opt runs in Experiment A using ULSA

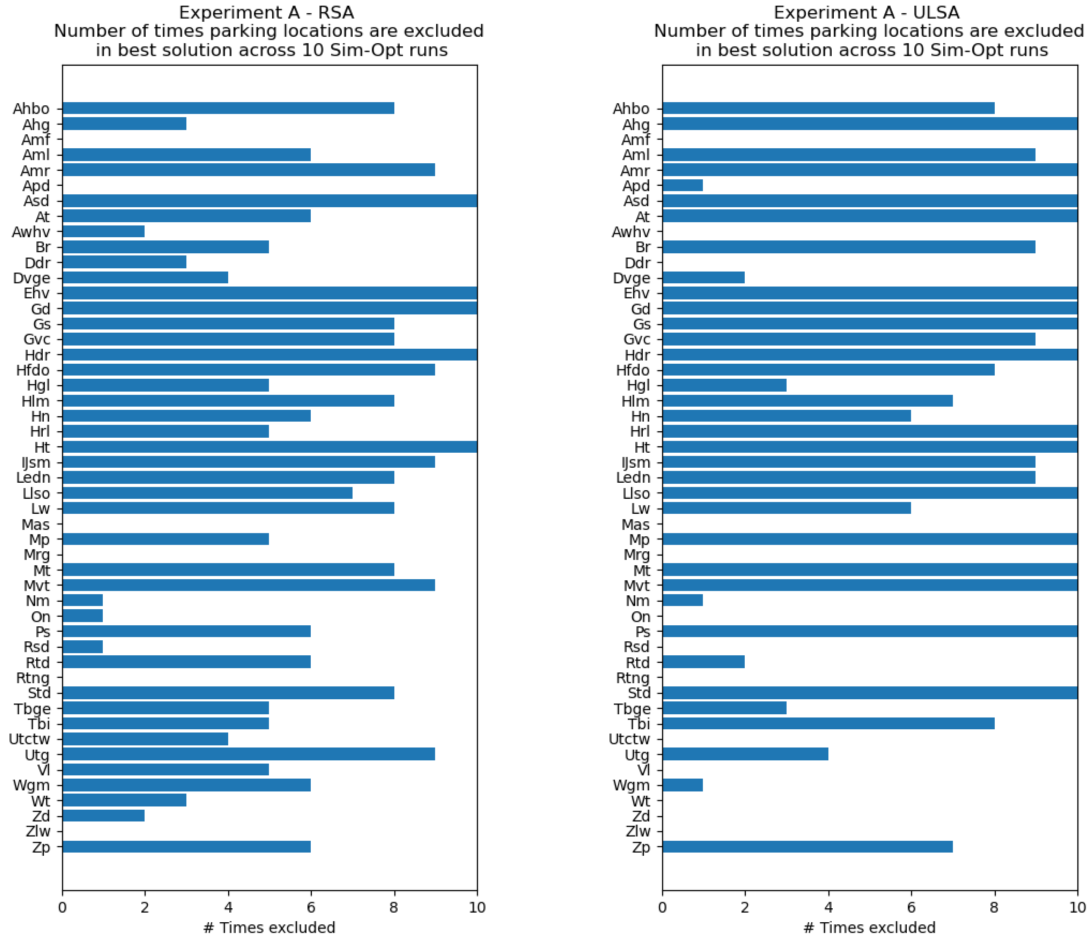
Figure 5.2: Progress of all KPI's for all Sim-Opts in Experiment A for both the RSA and ULSA.

both the RSA and the ULSA to give insight into the inclusion- and exclusion frequency for the individual parking locations (Figure 5.3). Figure 5.3a shows the number of times the individual parking locations have been excluded in all 10 instances during the RSA and Figure 5.3b shows the number of times the individual parking locations have been excluded in all 10 instances during the ULSA. There are similarities between Figure 5.3a and 5.3b. In Figure 5.3a it can be seen that some parking locations are included in all 10 best solutions and some are excluded in all 10 best solutions. In Figure5.3b it can be seen that more parking locations are excluded in all 10 ULSA runs compared to the RSA runs, but this selection incorporates at least the parking locations that are excluded during the RSA runs. The ULSA finds that the parking locations which are included in all instances, are also included in all, or close to all, instances of the RSA. The above mentioned observations indicate several things. First, some specific parking locations are favourable to either include or exclude from the solution. The consistency of which the two different search algorithms find this set of parking locations indicate that these parking locations can be classified as respectively more, or less valuable within the system. Secondly, the ULSA finds solutions which are more alike, which also suggests the ULSA finds solutions with more confidence than the RSA.

The distribution of excluded parking locations can be visualised in a histogram which displays the frequency of the exclusion frequency. Figure 5.4 shows such a histogram of the data from Figure 5.3. The two histograms of Figure 5.4 show the distribution of how many times any parking location was excluded during the experiments for both the RSA and ULSA. Figure 5.4a shows that the distribution is more uniform than the distribution in 5.4b. A evenly distributed histogram indicates that the search algorithm is not converging and is still in the process of trying out neighbouring solutions without much progress. An outwardly skewed histogram, as can be seen in 5.4b, indicates that the optimisation process is close to an optimum.[1]

Figures 5.5 and 5.6 show the data of Figure 5.3 in a map of the Dutch railway network. Figures 5.5a and 5.5b show the occurrences and geographical locations of the included parking locations. Figures 5.6a and 5.6b show the occurrences and geographical locations of the excluded parking locations. These figures can be compared to Figure 4.4, which shows the project locations and frequency. Generally, the included parking locations are positioned near project locations. For example, there is a cluster of projects in the north-eastern part of the Netherlands. Both search algorithms find that it is valuable to include *On*, the parking location central to that area, and exclude *Lw* more often. A general trend is that the included parking locations are clustered more towards the centre of the rail network, where more projects are concentrated. Parking locations near the outer part of the network are excluded more often. Excluding parking locations on network branches would decrease the travelled distance. For example: if a parking location is located on a network branch, there is more infrastructure to be reached towards and beyond a neighbouring network node, than towards the end of the remaining network branch. As a result there is a higher probability of journeys going one specific way and these journeys being relatively longer. This increases the travel distance and thus the travel costs. It would be more efficient to position parking locations in a central position in relation to the amount of projects or infrastructure in the vicinity.

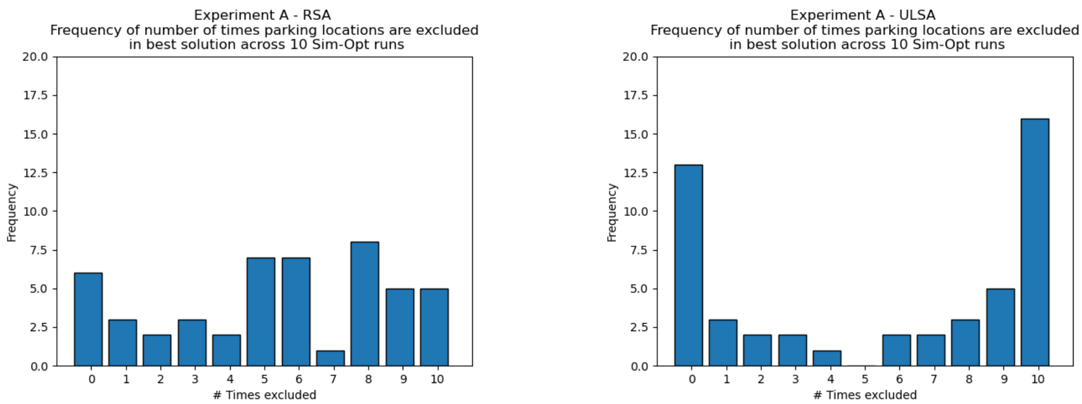
[1] It has to be noted that this optimum does not need to be the global optimum, since using the SA heuristic, this global optimum cannot be defined by searching through a subset of the solution space.



(a) Experiment A - RSA

(b) Experiment A - ULSA

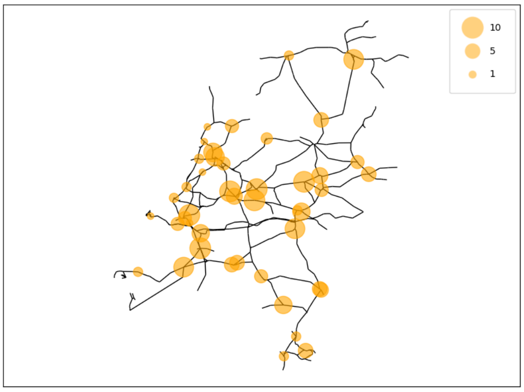
Figure 5.3: Occurrence of exclusion for individual parking locations in Experiment A.



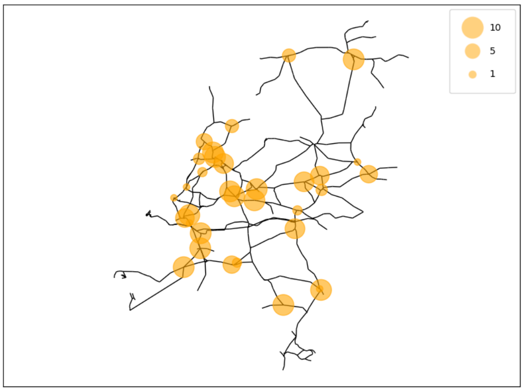
(a) Experiment A - RSA

(b) Experiment A - ULSA

Figure 5.4: Histogram of occurrence of exclusion for individual parking locations in Experiment A.

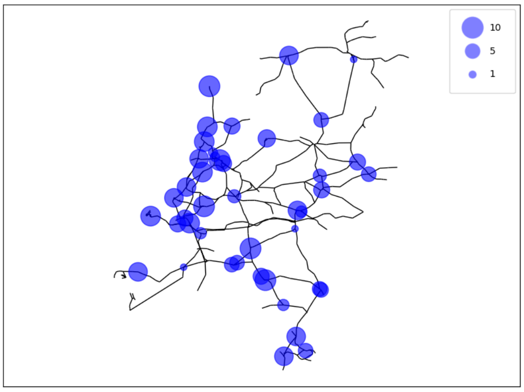


(a) Experiment A - RSA

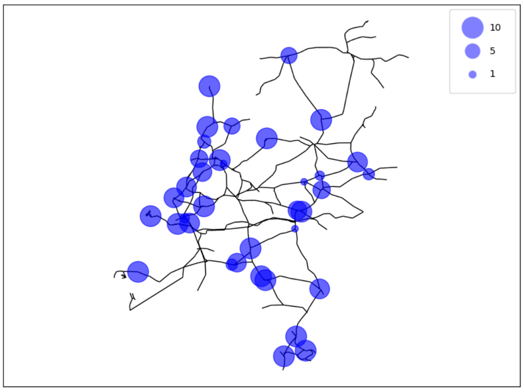


(b) Experiment A - ULSA

Figure 5.5: Occurrence of included individual parking locations in Experiment A.



(a) Experiment A - RSA



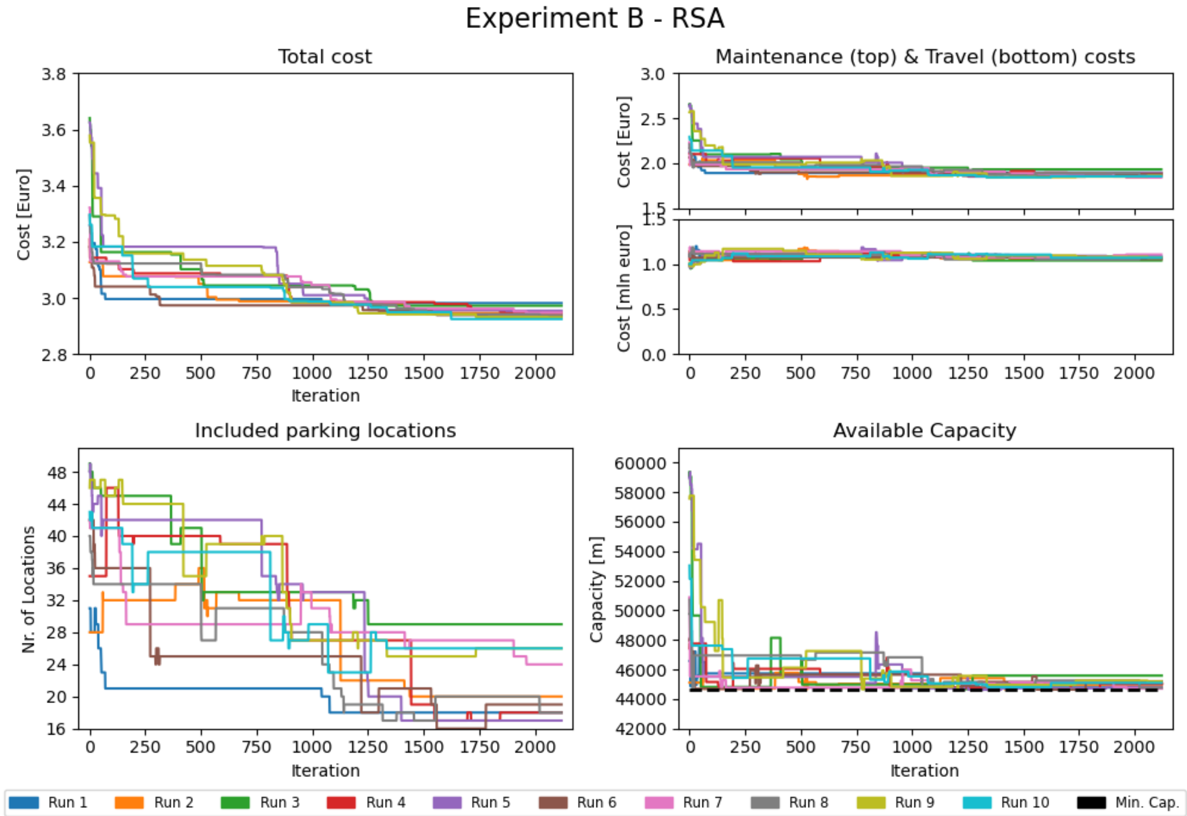
(b) Experiment A - ULSA

Figure 5.6: Occurrence of excluded individual parking locations in Experiment A.

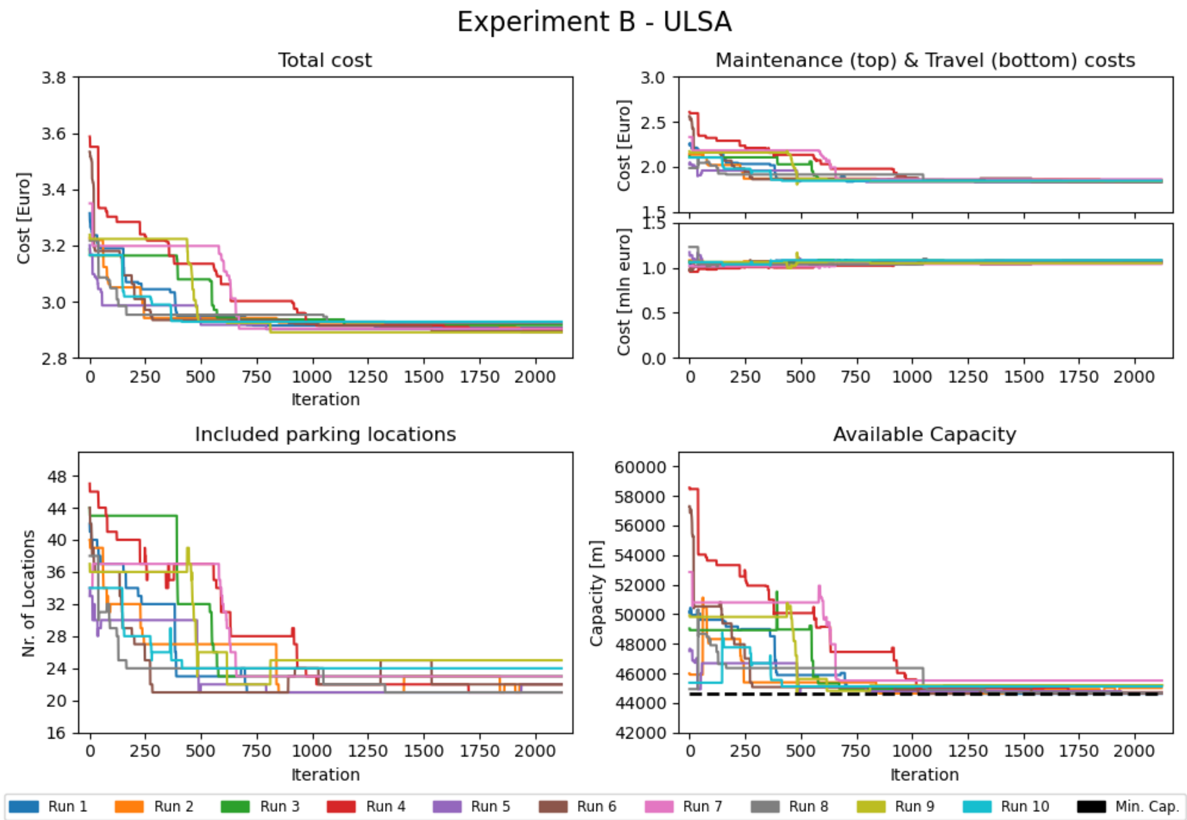
5.2.2. Experiment B

To describe the results of Experiment B, similar figures are used. Figure 5.7a and 5.7a show sets of graphs for the Sim-Opts using the RSA and ULSA respectively.

In Experiment B, a random number of parking locations was excluded for the initial iteration. The behaviour of all the graphs for both RSA and ULSA is comparable to the behaviour of the graphs for Experiment A. The total costs have a steep decline at the start of the iteration process and reach a tighter spread near the second half. It is notable that the spread for the RSA seems tighter in Experiment B than in Experiment A. The maintenance costs show a similar steep reduction and also during Experiment B, the travel costs increase only slightly. The spread and downtrend of the number of included parking locations and the available capacity is analogous to the behaviour during Experiment A. The final best solutions also range between 2.9 and 3.0 million euros. The minimum available capacity for the best found solutions all lie close to the minimum set capacity.

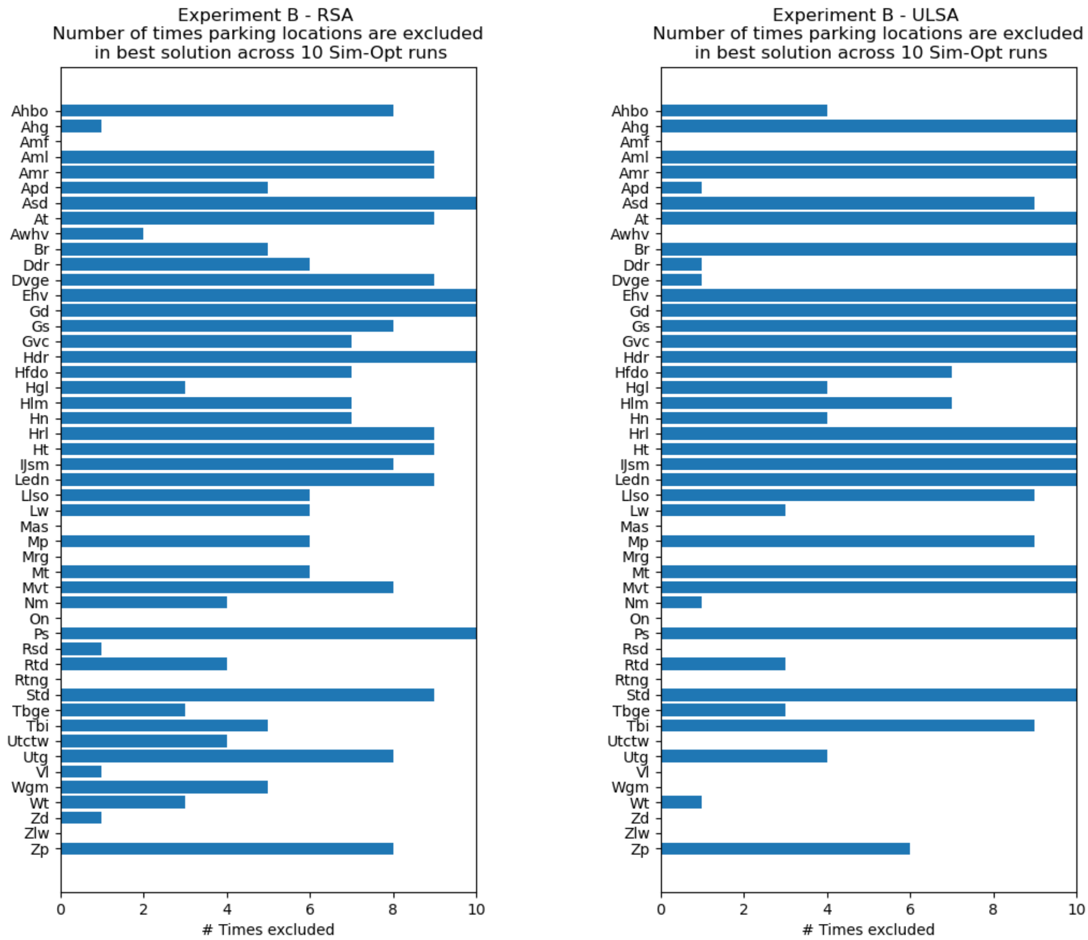


(a) Progress of all KPI's for all 10 Sim-Opt runs in Experiment B using RSA



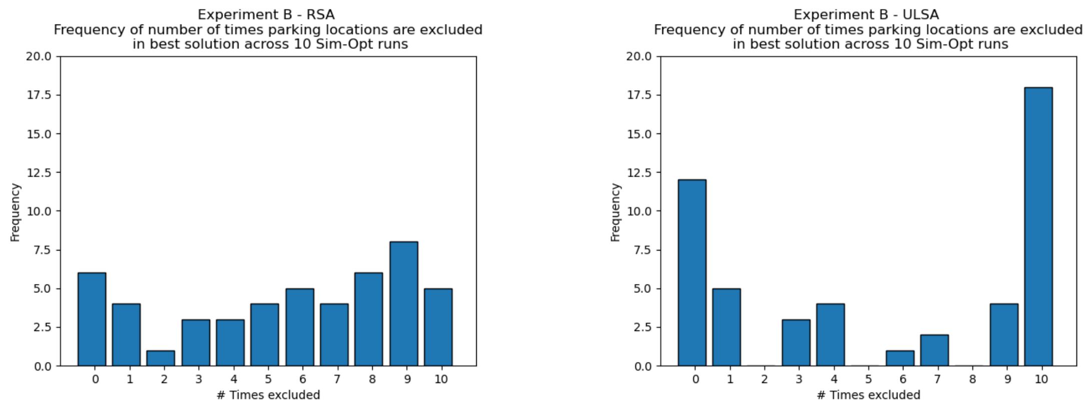
(b) Progress of all KPI's for all 10 Sim-Opt runs in Experiment B using ULSA

Figure 5.7: Progress of all KPI's for all Sim-Opts in Experiment B for both the RSA and ULSA.



(a) Experiment B - RSA (b) Experiment B - ULSA

Figure 5.8: Occurrence of exclusion for individual parking locations in Experiment B.



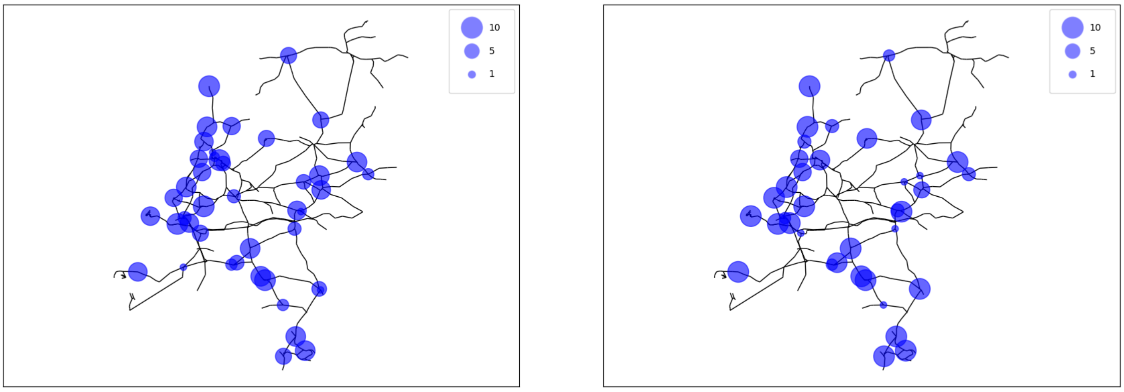
(a) Experiment B - RSA (b) Experiment B - ULSA

Figure 5.9: Histogram of occurrence of exclusion for individual parking locations in Experiment B.

(a) Experiment B - RSA

(b) Experiment B - ULSA

Figure 5.10: Occurrence of included individual parking locations in Experiment B.



(a) Experiment B - RSA

(b) Experiment B - ULSA

Figure 5.11: Occurrence of excluded individual parking locations in Experiment B.

Figure 5.9a shows the number of times the individual parking locations have been excluded in all 10 instances during the RSA and Figure 5.9b shows the number of times the individual parking locations have been excluded in all 10 instances during the ULSA. These graphs show similarities with Figures 5.3a and 5.3b of Experiment A. Most parking locations that are included during all instances in experiment A, are also included during all instances in experiment B. The number of times individual parking locations are excluded in experiment A is similar to the number of times they are excluded in experiment B. Again, in Figure 5.3b it can be seen that more parking locations are excluded in all 10 ULSA runs compared to the RSA runs, but this selection incorporates at least the parking locations that are excluded during the RSA runs. Similar to Experiment A, the parking locations which are included in all instances of the ULSA, are also included in all, or close to all, instances of the RSA. Figures 5.8a and 5.8b show that the ULSA finds solutions with more confidence.

Figures 5.5 and 5.6 show the occurrences and geographical locations of the included parking locations for Experiment B. Similar to Experiment A, the included parking locations are generally positioned near project locations. The parking locations included in the best solutions tend to be located near the centre of the network. Parking locations near the outer part of the network are excluded more often.

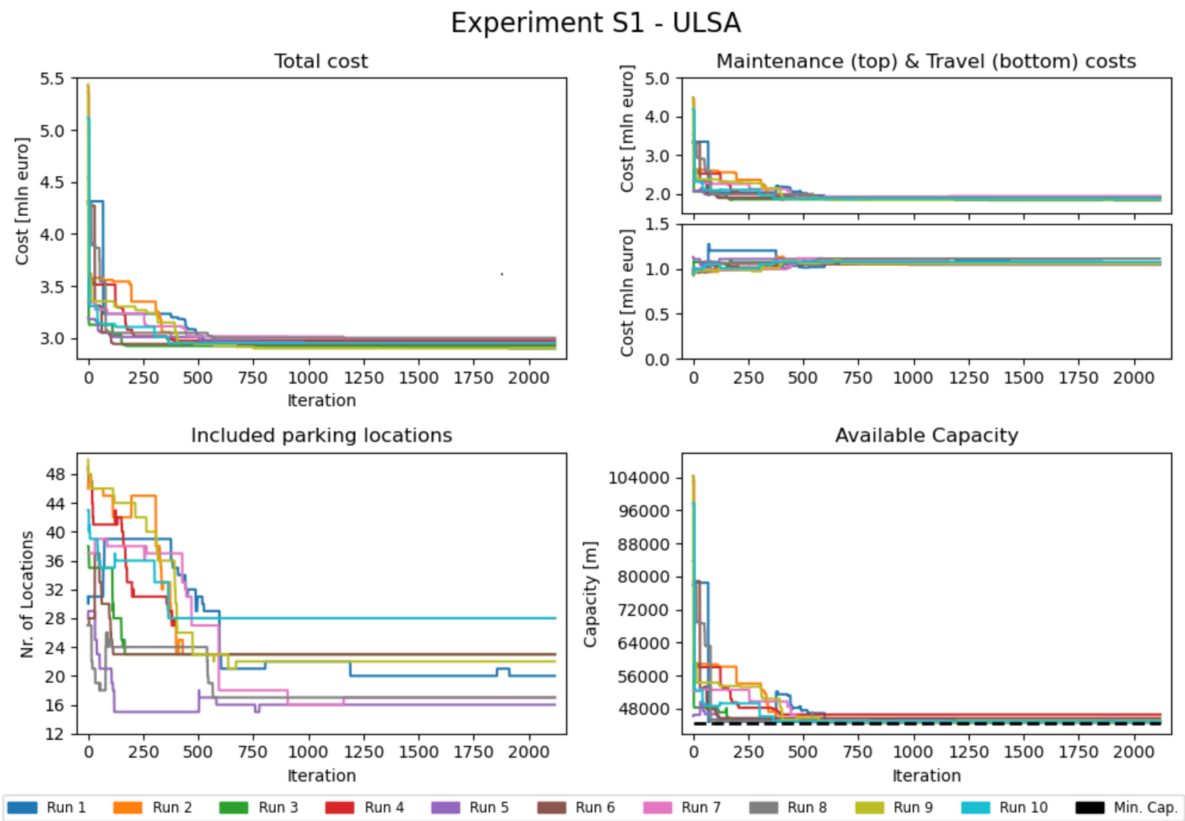
5.2.3. Experiment S1

In this experiment, a fictional parking location is added to the system. First, a single simulation is run wherein the new fictional parking location *Utvr* is the only included parking location in the system. All trains reside, dispatch and return to *Utvr*. Table 5.2 gives an overview of the KPI's for this scenario.

Table 5.2: Experiment S1 - System performance with one fictional parking location.

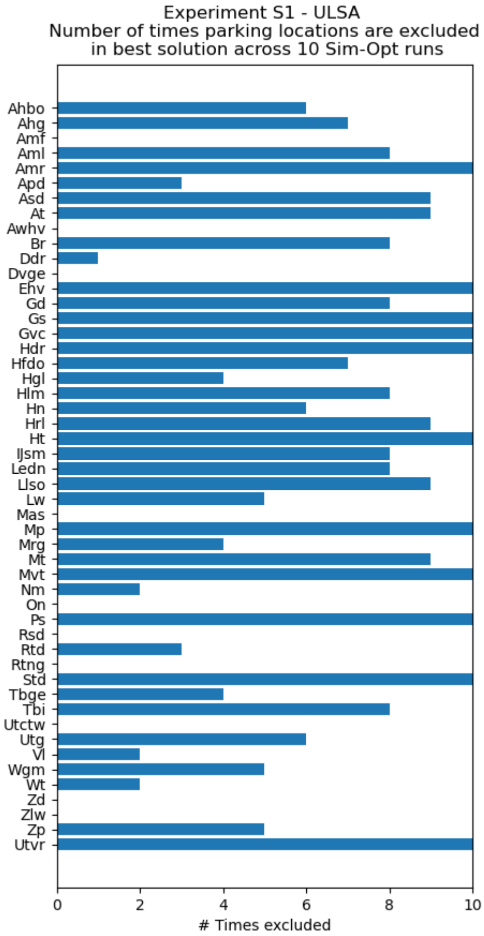
KPI	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
Mean	45000	1	0.904	1.83	2.74

It is interesting to see the outcome of running the Sim-Opt using the ULSA with the possibility of including *Utvr* into the system. Figure 5.12a shows the progress of 10 Sim-Opt instances. In some instances, the total cost a the start of the Sim-Opt is considerably higher than in the other experiments in this research. A similar increase can be seen for the maintenance costs and the available capacity. It is evident that the new expensive parking location *Utvr* was included in those initial solutions. The development in the number of included parking locations shows a wide spread in the first half if the Sim-Opt. Although the lines stabilise in the second half of the Sim-Opt, the spread does not become as tight as in the other experiments in this research. The cooling process has progressed to a point where the SA process can no longer escape from local minima.

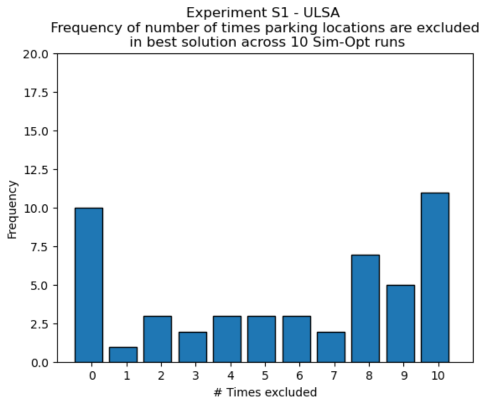


(a) Progress of all KPI's for all 10 Sim-Opt runs for Experiment S1 using ULSA

From Figure 5.13a it becomes clear that the ULSA has excluded *Utvr* from all 10 instances. Even though the maintenance cost and travel cost are lower when *Utvr* is the only included parking location,



(a) Experiment S1 - ULSA



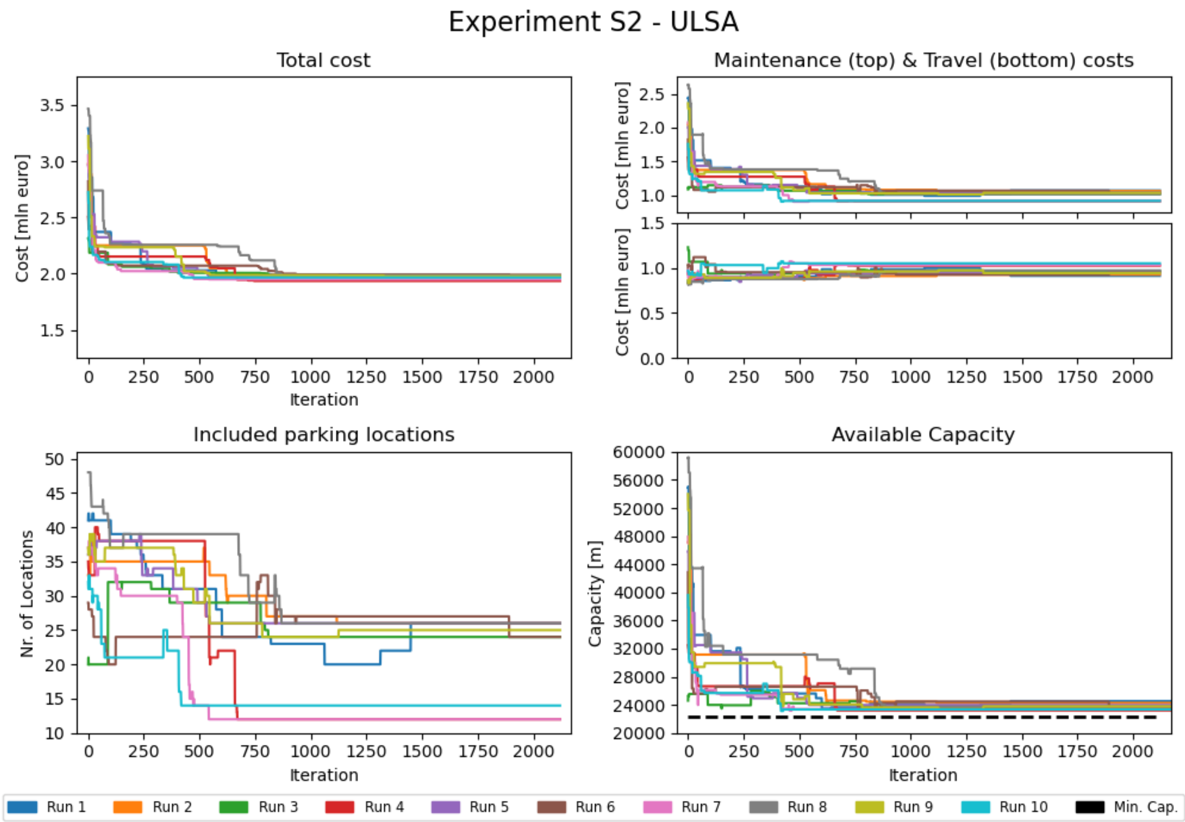
(b) Experiment S1 - ULSA

the search algorithm does not find this solution. In Experiments A and B, the available capacity for the best solutions bottomed out at ~46km and ~22 parking locations. *Utvr* mimics the capacity of these parking locations. So when this parking location is included during a certain iteration in the Sim-Opt, it would take a substantial number of iterations to close enough other, smaller, parking locations to compensate for the giant sudden maintenance cost increase. Using the architecture of the SA process in this research, there is a 1 in 3 chance that a closing-permutation happens. In short, this heuristics method is not suited for handling this scenario. The possible cost jumps are too high in comparison to the permutation size.

5.2.4. Experiment S2

It can be observed that all best solutions for all instances of Experiment A, B and S1 have an available capacity near to the minimum set capacity. Experiment S2 investigates the system behaviour when relaxing this boundary.

Figure 5.14a shows the progress of 10 Sim-Opt instances for Experiment S2. The total cost decreases rapidly and the spread of solution cost is tight in the second half of the iteration processes. The improvement in maintenance costs is (again) the major contributor to the decrease in total costs. The travel costs seem to increase a minimal amount. The spread in included parking locations is relatively large, with some best solutions having more included parking locations than the minimal number of included parking locations found in Experiments A, B and S1. Two of the best solutions include only 11 parking locations, which is considerably lower than in Experiments A, B and S1. The available capacities in the best solutions are all found near the new, lowered minimum capacity. It is interesting to see that the travel costs are not considerably higher when more parking locations are allowed to be excluded when compared to the other experiments in this research.

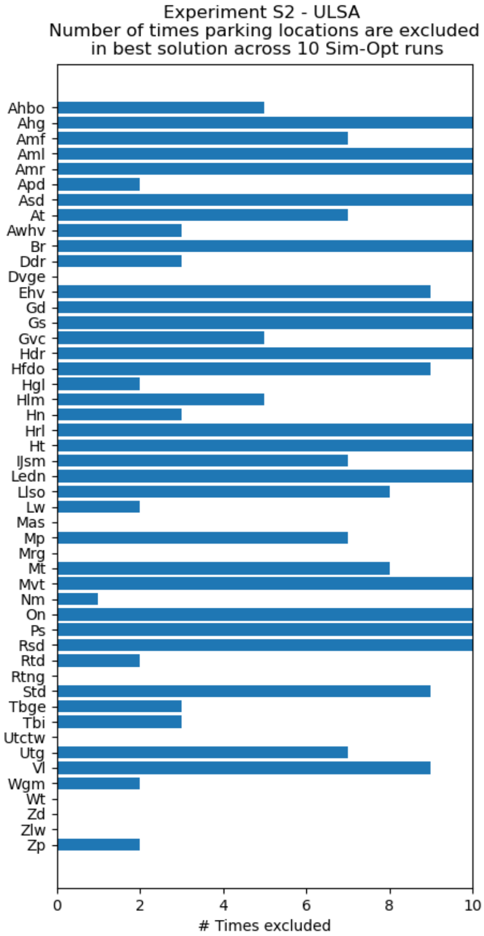


(a) Progress of all KPI's for all 10 Sim-Opt runs for Experiment S2 using ULSA

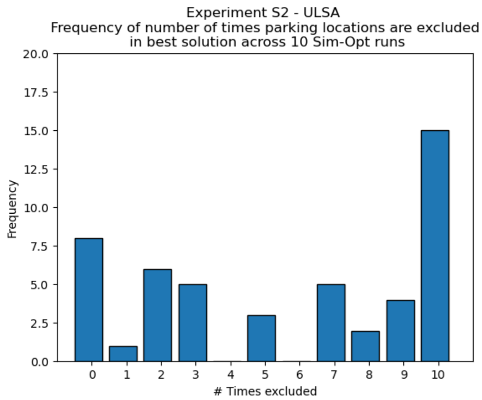
Figure 5.15b shows that the histogram is skewed outward, indicating more confident findings compared to the more uniformly distributed histogram of Figure 5.15a. The number of times parking locations are excluded in all instances is higher compared to the number of times parking locations are included in all instances. On average, more parking locations are closed in the best solutions when the capacity buffer is removed. This is in accordance with the fact that there is less capacity needed. Lowering the amount of available capacity to the bare minimum seems to have little effect on the travelling costs.

When inspecting Figures 5.16a and 5.16b it can be seen that the search algorithm still tends to include the central laying parking locations and exclude the parking locations near the outer parts of the network. Parking locations are concentrated near the bulk of the project locations.

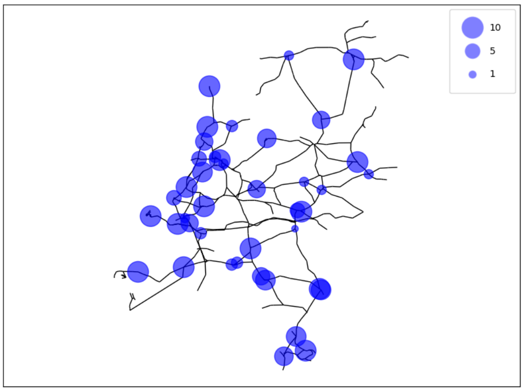
Interestingly in this experiment, *On* is closed considerably more often and *Lw* is opened more often when compared to the other experiments. With the removal of the capacity buffer, the ULSA finds that *On* is used relatively less than *Lw*. The amount of trains needed in that area can suffice with the capacity offered by *Lw* alone. In other experiments the number of trains needed in that area require more capacity than the 55% of *Lw's* total track length and thus it is cheaper to open *On*, which is several times larger. This example implies that the size of a parking location can overrule the strategic positioning to some extend.



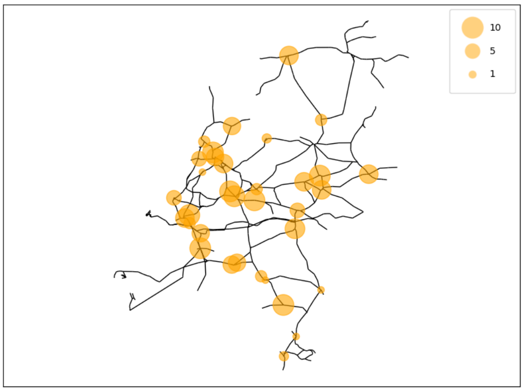
(a) Experiment S2 - ULSA



(b) Experiment S2 - ULSA



(a) Included parking locations



(b) Excluded parking locations

Figure 5.16: Occurrence of included and excluded individual parking locations in Experiment S2.

5.2.5. Summary

There is no notable difference between the outputs of Experiment A and Experiment B. It can therefore be concluded that the SA process in both experiments have had enough time to find similar solutions. The results of both Experiment A and B show that the ULSA converges sooner than the RSA. Furthermore, the spread in the best solutions across all instances is tighter when applying the ULSA and the ULSA finds solutions which are more alike across all instances. These findings indicate that the ULSA gives more robust solutions when compared to the RSA and makes the SA process more efficient.

Table 5.3: Mean results of all best found solutions per experiment.

Experiment	Capacity [m] (SD)	Locations [#] (SD)	Travel [mln €] (SD)	Maintenance [mln €] (SD)	Total cost [mln €] (SD)
Current System	59349 (0)	49 (0)	0.983 (0.0146)	2.65 (0)	3.64 (0.0146)
A (RSA)	44920 (130)	22 (3.41)	1.07 (0.0294)	1.88 (0.0496)	2.95 (0.0305)
B (ULSA)	44789 (308)	22 (1.47)	1.07 (0.0183)	1.84 (0.0200)	2.91 (0.0127)
B (RSA)	45013 (255)	22 (4.10)	1.07 (0.0171)	1.88 (0.0247)	2.95 (0.0169)
B (ULSA)	44936 (294)	22 (1.28)	1.06 (0.0135)	1.84 (0.0089)	2.91 (0.0124)
S1 (ULSA)	45092 (579)	22 (4.05)	1.07 (0.0182)	1.88 (0.0334)	2.95 (0.0336)
S2 (ULSA)	23757 (394)	22 (5.85)	0.973 (0.0455)	1.00 (0.0589)	1.97 (0.0167)

Table 5.3 shows a summary of the results by stating the mean values of all Sim-Opts for the KPI's. It can be concluded that ProRail can potentially reduce the annual cost related to the travelling and parking of maintenance trains by ~20%. This research found that the best solutions in terms of overall cost includes on average 22 parking locations, which can reduce the maintenance cost by ~30%. The total travel costs show a relatively small increase of ~9%. The capacity for the optimal solution is close to the minimum capacity requested by the contractors. This implies that the increase in travel costs does not justify the use of more capacity than twice the train fleet length.

Special case S1, shows that the use of a single parking location, located in the centre of the railway network, would decrease the total cost considerably (by 24%), but is thought to be impractical. Special case S2 shows that in an extreme scenario wherein no extra space is needed for shunting, any change in travel costs during the optimisation process still does not justify the use of more capacity than the absolute minimum.

The included parking locations in the resulting solutions of the Sim-Opts are spread across the network, but are preferably located near the project locations. As a result, parking locations near the outer part of the railway network are excluded more often and parking locations towards the centre of the railway network are included more often in the Sim-Opts.

6

Conclusion & Recommendations

6.1. Conclusion

The main research question is answered in this chapter by discussing the sub-research questions. This chapter will end with an evaluation and the resulting recommendations for future research.

The first sub-research question *"What optimisation method can be used to find the best solution"* is answered in the second chapter. A literature study has been done in order to find a feasible solution method for the case study. The FLP has been studied extensively in literature. Due to the dynamic nature of the problem in this research, a DES has been chosen to model the processes of the maintenance trains in the Dutch Railway system. Due to the size of the solution space a Sim-Opt using a SA approach was proposed, wherein the DES acts as an evaluation function for the optimisation agent. The work of Qin (2012) and Frausto-Solis et al. (2007) was used to formalise the simulated annealing process. This process is expanded by implementing relevant usage level data of the simulation output into the neighbourhood search function. This improved neighbourhood search function makes a more directed search through the solution space possible.

The second sub-research question *"How are parking facilities for maintenance trains currently being used?"* is answered in Chapter 3. The current system and the accompanying processes have been analysed on the basis of the railway network, parking locations, train units and expenditures. The yearly parking capacity is requested by the contractors active in the Dutch railway network. The requested parking capacity in 2019 amounted to 59km, while the total amount of maintenance trains have a combined length of 22km. Contractors are not bounded to certain areas of the railway network and it is therefore possible that their maintenance trains travel throughout the entire network, covering relatively large distances. Contractors use the space on parking locations interchangeably among themselves. Due to this dynamic process the available capacity on parking locations is only known shortly in advance, making it practically impossible to plan all journeys and stays in advance at the start of the year. The number of trains and the individual train types of the maintenance train fleet that is active in the Dutch Railway network is not known to ProRail. Furthermore, this train fleet is dynamic, since the contractors also operate in other countries within Europe. The travel cost of maintenance trains are difficult to specify, since the expected expenditures are calculated into a yearly contract between the contractor and ProRail, but can be subdivided into four aspects: staff, energy, vehicle maintenance and vehicle depreciation. The cost to maintain the infrastructure of parking locations consist of renewal costs (every 60 years on average), regular maintenance costs, management costs, facilities expenditures and safe guard system of switches and tracks. Tracks reserved for maintenance trains are usually a part of a bigger yard. Therefore, the maintenance costs can best be specified through the renewal and regular maintenance cost of the infrastructure directly linked to the tracks used by maintenance contractors.

The third sub-research question *"How can the railway network and including maintenance related movements be modelled?"* is answered in Chapter 4. A Discrete Event Simulation (DES) has been designed to simulate movements of trains trough the network and the related dynamic processes. The main components of the simulation are the projects and trains. The entire simulation is modelled around the situation of the year 2019. The projects were imported beforehand and simulated according to the yearly project schedule. During the simulation projects are assigned to an available train in the system. In the period before the project is due, the train travels to a parking location with free space nearest to the project location. Parking locations can only be occupied up to ~55% of the usable track length in order to facilitate practical movements. The DES was implemented in the Sim-Opt design of Qin (2012) using a SA process. This process was extended by implementing an evaluation of the relative usage level of the parking locations during the previous iteration. This evaluation guides the search through the solution space by generating promising neighbouring solutions during the Sim-Opt process.

The new neighbourhood search algorithm (ULSA) is tested against the random neighbourhood algorithm (RSA) used by Qin (2012) in Chapter 5. The evaluation is performed for both search algorithms using multiple instances of the Sim-Opt. The results of this evaluation show that the ULSA converges sooner than the RSA. Furthermore, the spread in the best solutions across all instances is tighter when applying the ULSA. When looking at the details of the solutions, it can be seen that the ULSA finds solutions across all instances which are more alike. These findings indicate that the ULSA gives more robust solutions when compared to the RSA and makes the SA process more efficient.

The final sub-research question *"Which recommendations towards ProRail arise from the results of the optimisation method?"* is answered in Chapter 5. According to the simulation used in this research, ProRail can potentially reduce the annual cost related to the travelling and parking of maintenance trains by ~20%. A large part of this improvement can be achieved by rethinking the parking location strategy. This research found that the best solutions in terms of overall cost includes on average 22 parking locations, which can reduce the maintenance cost by ~30%. This is in stark contrast with the current situation, where 49 parking locations are used for maintenance trains.[1] The total travel costs show a relatively small increase of ~9%. The capacity for the optimal solution is close to the minimum capacity (including a capacity buffer of 100%) requested by the contractors. This implies that the increase in travel costs does not justify the use of more capacity than twice the train fleet length. The improved performance of the ULSA can encourage ProRail to use the relative usage level as a metric for determining the value of a parking location.

The included parking locations in the Sim-Opts are spread across the network, but are preferably located near the project locations. The number of projects in a certain area of the network is related to the amount of infrastructure.[2] As a result, parking locations near the outer part of the railway network

[1] It is too premature to make policy changes on the basis of these conclusions, since the simulation used in this research is highly simplified. A reflection on these simplifications and recommendations are discussed in the following chapter.

[2] The type of infrastructure is another factor which might impact the amount of projects in a certain area.

are excluded more often and parking locations towards the centre of the railway network are included more often in the Sim-Opts. Special case *S1*, shows that the use of a single parking location, located in the centre of the railway network, would decrease the total cost considerably (by 24%). Such a situation is thought to be impractical, since the added travel time for contractors, would reduce the effective labour time for the engineer. Special case *S2* shows that in an extreme scenario wherein no extra space is needed for shunting, the change in travel costs still does not justify the use of more capacity than the absolute minimum. The parking locations selected by the model are, again, located across the network. This implies that a nationwide coverage is important. However, S2 also shows that the size of a parking location can overrule the strategic positioning to some extend.

6.2. Reflection and Recommendations

This research shows that DES is a feasible evaluation method for finding possible solutions to the FLP in a Sim-Opt method. However, a lot of simplifications are needed to bring the calculation time down. Furthermore, the solution space of such a problem is relatively large. The optimisation process would benefit from a selection process, on top of the improved neighbourhood function applied in this research, in order to reduce the solution space.

A permutation size of one has been used in the neighbourhood search function, but a larger permutation size can reduce the number of iterations needed to find an optimum. Furthermore, a larger permutation size can counteract the limitations found in special case S2. A larger fixed permutation size can aid the search in an early stage, when coarse steps occur relatively frequently, but might not be desired in a later stage when the temperature is lower and generally fine steps are needed. Using a fixed large permutation size in a late stage can cause the algorithm to over- or underestimate the optimal solution. The algorithm may include or exclude more parking locations than the optimal solution requires. Therefore, a dynamic permutation size is preferred in future research. For this dynamic permutation size a usage level threshold can be determined. This usage level threshold should reflect a practically viable parking location, e.g. exclude a parking location if it has less than *n* movements in a year.

Although the conclusions made in the previous section are very promising, it should be kept in mind that this research uses a highly simplified model, which can be considered to stand far from the real world system and its dynamics. The railway system is a complex system, which needed a great amount of simplifications in order to accommodate it into a graduation thesis.

The results show that the capacity buffer is a prominent limiting factor. This capacity buffer needs to be investigated further. The buffer is meant to ease the shunting process on parking locations. Research should be done to get insight into the shunting process: how many movements are necessary, how long do different shunt-processes take, what impact does the track layout have on the shunting process and can this be optimised? A cost-benefit analysis should be performed in order to determine the capacity buffer size.

The unknown maintenance train fleet made the use of a fictional train fleet necessary. It is necessary for future research to find a way to specify the train fleet that operates and resides within the Netherlands. In order to do this, the dynamics with the rest of Europe should be investigated. It should be determined how often maintenance trains cross the border and if there are any trains stationed in the Netherlands even though they do not need to be here. It could also be possible that there is unused parking capacity in neighbouring countries.

A big limitation in this research was the amount of data available. As a result, only one year of maintenance projects was simulated. Railway infrastructure lifespan exceeds this period greatly. Future research should thus be based on multiple years of data, which at least covers a sizeable period of the lifespan of infrastructure parts.

Measurement trains are not taken into consideration in this research, although they use parking locations as well and travel more often than maintenance trains. These type of trains should be taken into consideration in future research.

The simulation in this research allowed trains to reside at any available parking location. In real life however, some contractors are assigned to specific parking locations due to any specific facilities needed or adjacent buildings. Any additional measurement trains also reside on a few specific parking locations in real life. This extra constraint should be added in future models.

The costs that contractors make during journeys to and from project locations is incorporated into the value of a yearly contract. However, the exact numbers are not known and thus this research used a simple estimation. The portion of the contract value which covers the related travel expenditures should be specified more clearly. Furthermore, the practical inconvenience of longer train journeys on a working day for contractors should be translated into a monetary metric in order to make a solid cost-benefit analysis in the future.

The results show that the parking locations are preferably placed near the project locations. The assumption that the amount of infrastructure in a certain vicinity is a measure for the amount of projects, leads to a recommendation for the methodology in future research: a mathematical model whereby the amount and type of infrastructure is evaluated on the network nodes and edges can give an exact solution to the problem. Such a solving method can be further explored in future research.

Bibliography

Derya Celik Turkoglu and Müjde Erol Genevois. A comparative survey of service facility location problems. *Annals of Operations Research*, 292, 09 2020. doi: 10.1007/s10479-019-03385-x.

Leon Cooper. *Location-Allocation Problems. Operations Research.*, volume 11. 1963. ISBN 331-343.

C.D. Dao, Andreas Hartmann, and Rob Basten. Maintenance scheduling for railway tracks under limited possession time. *Transportation engineering journal of ASCE*, 144, 04 2018. doi: 10.1061/JTEPBS.0000163.

Mark S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. 2013. ISBN 978-1-118-53696-4.

Juan Frausto-Solis, E. F. Román, David Romero, Xavier Soberon, and Ernesto Liñán-García. Analytically tuned simulated annealing applied to the protein folding problem. pages 370–377, 2007.

Daniel Erlandsson Fredrik Persson. Solving location problems with simulation modelling - a case fom the construction industry. *2013 Winter Simulation Conference*, 2013. doi: 978-1-4799-2076-1. URL https://informs-sim.org/wsc13papers/includes/files/297.pdf.

Hasan Hosseini nasab and Fatemeh Mobasheri. A simulated annealing heuristic for the facility location problem. *Int. J. of Mathematical Modelling and Numerical Optimisation*, 4:210 – 224, 01 2013. doi: 10.1504/IJMMNO.2013.056532.

Jeffrey Joines, R. Barton, Keebom Kang, Paul Fishwick, James Swisher, Sheldon Jacobson, Paul Hyden, and Lee Schruben. A survey of simulation optimization techniques and procedures. 01 2001.

M.L. Kho. Facility location optimization and maintenance routing in a railway network, 2016.

Scott Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science (New York, N.Y.)*, 220:671–80, 06 1983. doi: 10.1126/science.220.4598.671.

Tomas Lidén. Railway infrastructure maintenance - a survey of planning problems and conducted research. *Transportation Research Procedia*, 10, 07 2015. doi: 10.1016/j.trpro.2015.09.011.

Siqi Liu, Boliang Lin, Jiaxi Wang, and Jianping Wu. Modeling the multi-period and multi-classification-yard location problem in a railway network. *Symmetry*, 10:135, 04 2018. doi: 10.3390/sym10050135.

D. Maringer. Portfolio management with heuristic optimization. 8, 01 2003. doi: 10.1007/b136219.

Jeroen van Rutten ProRail, Marcel Drenth. Referentiekader: Opstellen van materieel ten behoeve van werkzaamheden voor beheer., 2020.

Jin et al. Qin. Combined simulated annealing algorithm for the discrete facility location problem. 01 2012. doi: 10.1100/2012/576392.

Ragheb Rahmaniani, Abdolsalam Ghaderi, and Mohammad Saidi-Mehrabad. A simulated annealing algorithm within the variable neighbourhood search framework to solve the capacitated facility location-allocation problem. *Journal of Optimization in Industrial Engineering*, 5:45–54, 03 2012.

C.S. ReVelle, H.A. Eiselt, and M.S. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3):817 – 848, 2008. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2006.12.044. URL http://www.sciencedirect.com/science/article/pii/S037722170700080X.

Masoud Hekmatfar Reza Zanjirani Farahani. *Facility Location - Concepts, Models, Algorithms and Case Studies*. Springer-Verlag, 2009. ISBN 978-3-7908-2150-5. URL https://www.researchgate.net/publication/227149037_Dynamic_Facility_Location_Problem.

Robert G. Sargent. Verification and validation of simulation models. *Engineering Management Review, IEEE*, 37:166 – 183, 01 2011. doi: 10.1109/WSC.2010.5679166.

Spoorpro. Grootste vernieuwingstrein van europa vervangt 20.000 dwarsliggers, 07 2013. URL https://www.spoorpro.nl/spoorbouw/2013/07/05/grootste-vernieuwingstrein-van-europa-vervangt-20-000-dwarsliggers/.

Spoorpro. Acm: sporen kijfhoek hoeven niet opnieuw te worden verdeeld, 02 2019. URL https://www.spoorpro.nl/spoorbouw/2019/02/05/acm-sporen-kijfhoek-hoeven-niet-opnieuw-te-worden-verdeeld/.

Spoorpro. Prorail: ondanks crisis volgend jaar meer aanvragen treinritten, 08 2020. URL https://www.spoorpro.nl/spoorbouw/2020/08/24/prorail-ondanks-crisis-volgend-jaar-meer-aanvragen-treinritten/.

Jan C.W. van Ommeren, A.F. Bumb, and A.V. Sleptchenko. Locating repair shops in a stochastic environment. *Computers & operations research*, 33(6):1575–1594, 2006. ISSN 0305-0548. doi: 10.1016/j.cor.2004.11.008.

G. Wainer and Pieter Mosterman. *Discrete-Event Modeling and Simulation: Theory and Applications*. 04 2016. ISBN 9781315218731. doi: 10.1201/9781315218731.

Tao Zhang, John Andrews, and Rui Wang. Optimal scheduling of track maintenance on a railway network. *Quality and Reliability Engineering International*, 29, 03 2013. doi: 10.1002/qre.1381.

A
Numerical Results

The results of the different experiments in this research are presented here. Per experiment 10 instances of the Sim-Opt were run. Per instance the best solution is given in terms of the KPIs: the available capacity, number of included locations, the travel cost, maintenance cost for all included parking locations and the total cost (travel + maintenance). The mean value and the corresponding standard deviation of the 10 instances are also given.

Table A.1: Experiment A - RSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	44841	20	1.07	1.84	2.91
2	45192	24	1.04	1.86	2.91
3	44686	26	1.08	1.84	2.92
4	44905	23	1.12	1.84	2.96
5	44820	19	1.07	1.89	2.96
6	44998	24	1.02	1.97	2.99
7	44997	20	1.06	1.90	2.96
8	44845	17	1.09	1.87	2.96
9	45002	29	1.02	1.96	2.99
10	44912	21	1.09	1.83	2.91
Mean	44919.8	22.3	1.07	1.88	2.95
Std Dev	130	3.41	0.0294	0.0496	0.0305

Table A.2: Experiment A - ULSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	44718	22	1.07	1.84	2.91
2	44620	21	1.10	1.82	2.92
3	44716	20	1.08	1.84	2.92
4	44697	21	1.09	1.82	2.92
5	44693	21	1.05	1.84	2.90
6	44665	21	1.08	1.82	2.89
7	45707	25	1.04	1.89	2.93
8	44650	21	1.07	1.83	2.90
9	44675	24	1.06	1.84	2.90
10	44750	22	1.05	1.85	2.90
Mean	44789.1	21.8	1.07	1.84	2.91
Std Dev	308	1.47	0.0183	0.0200	0.0127

Table A.3: Experiment B - RSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	44778	18	1.08	1.90	2.98
2	44965	20	1.06	1.89	2.95
3	45585	29	1.04	1.93	2.97
4	45029	18	1.06	1.88	2.95
5	44820	17	1.06	1.89	2.95
6	45205	19	1.06	1.88	2.94
7	44707	24	1.11	1.84	2.95
8	44765	18	1.08	1.86	2.94
9	45190	26	1.07	1.86	2.93
10	45088	26	1.07	1.85	2.92
Mean	45013.2	21.5	1.07	1.88	2.95
Std Dev	255	4.1	0.0171	0.0247	0.0169

Table A.4: Experiment B - ULSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	44623	21	1.08	1.83	2.91
2	45015	22	1.05	1.85	2.90
3	45191	23	1.06	1.85	2.92
4	44722	21	1.07	1.84	2.90
5	44651	22	1.05	1.84	2.89
6	44649	22	1.06	1.84	2.89
7	45513	23	1.04	1.86	2.90
8	44693	21	1.08	1.84	2.92
9	45186	25	1.05	1.84	2.89
10	45118	24	1.08	1.85	2.93
Mean	44936.1	22.4	1.06	1.84	2.91
Std Dev	294	1.28	0.0135	0.0089	0.0124

Table A.5: Experiment S1 - ULSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	44804	20	1.07	1.83	2.90
2	44979	23	1.07	1.86	2.93
3	44880	23	1.07	1.85	2.92
4	46600	28	1.05	1.92	2.98
5	44817	16	1.04	1.91	2.95
6	45740	23	1.06	1.89	2.94
7	44781	17	1.06	1.94	3.00
8	44638	17	1.11	1.89	3.00
9	44752	22	1.06	1.85	2.90
10	44932	28	1.08	1.88	2.95
Mean	45092.3	21.7	1.07	1.88	2.95
Std Dev	579	4.05	0.0182	0.0334	0.0336

Table A.6: Experiment S2 - ULSA

Run	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
1	24535	26	0.912	1.07	1.98
2	24273	26	0.926	1.06	1.99
3	23618	24	0.966	1.02	1.98
4	23212	12	1.03	0.910	1.94
5	23979	26	0.941	1.04	1.98
6	23684	24	0.957	1.03	1.99
7	23387	12	1.03	0.917	1.95
8	23749	26	0.973	1.01	1.99
9	23785	25	0.942	1.03	1.98
10	23345	14	1.05	0.915	1.97
Mean	23756.7	21.5	0.973	1.00	1.97
Std Dev	394	5.85	0.0455	0.0589	0.0167

B

Scientific Paper

Finding the optimal set of parking locations for maintenance trains in the Dutch railway network

Olof Olberts, Bilge Atasoy, Wubbo Boiten, Klaas Hofstra & Rudy Negenborn

Abstract— **The Dutch railway system is subject to maintenance, which is carried out by a group of rail contractors who need parking space to park and prepare trains in between projects. The objective of this study is to minimise the total costs as a result of the distance travelled by maintenance trains and the preservation of the included parking locations in the Dutch railway network. According to the dynamic nature and the NP-hardness of the Facility Location Problem (FLP), a Simulation-Optimisation (Sim-Opt) approach is proposed. This Sim-Opt consists of a Discrete Event Simulation (DES) and Simulated Annealing (SA) optimisation. Two neighbourhood functions are evaluated: a Random Search Algorithm (RSA) and a Utility Level Search Algorithm (ULSA), which takes the utility level of parking locations into consideration. This research shows that DES is a feasible evaluation method for finding possible solutions to the FLP in a Sim-Opt approach. The results of this evaluation show that the ULSA converges sooner than the RSA. Furthermore, the spread in best solutions across all instances is tighter when applying the ULSA. The findings indicate that the ULSA gives more robust solutions when compared to the RSA and makes the SA process more efficient. This research found that the best solutions in terms of overall cost includes on average 22 parking locations, which can reduce the maintenance cost by ˜30%. The average increase in travel costs of ˜9% does not justify the use of more capacity than the absolute minimum to house all trains.**

I. INTRODUCTION

The Dutch railway system is subject to maintenance, which is carried out by a group of rail contractors commissioned by ProRail. These rail contractors need parking space to park and prepare trains in between projects. Currently, ProRail reserves around 76km of track, distributed over 49 parking locations throughout the network for the contractors to make these activities possible. This collection of parking tracks is subject to maintenance itself and thus is an expenditure ProRail aims to decrease. Contractors prefer a large amount of parking locations distributed across the network as they can park and prepare trains close to their upcoming maintenance projects. This decreases travel time and costs to and from maintenance projects. ProRail wants to optimise the combined costs for the maintenance of parking locations and journeys made by contractors.

II. LITERATURE REVIEW

A. Facility Location Problem

Railway maintenance problems have been studied extensively in the literature *(Lidén, 2015), (Turkoglu, 2020)*, but are mainly linked to routing or scheduling problems. The problem in this research is focused on locating parking facilities and fewer studies have linked the Facility Location Problem (FLP) to a railway network. *Kho (2016)* has combined the routing of trains with the location problem for maintenance facilities in the Dutch railway network. The facilities were located based on the passenger train schedules to minimise deadheading cost. *Liu et al. (2018)* has conducted research on the location and size of classification yards and is concerned with freight trains and the allocation of their cargo within a classification yard.

The literature on railway research has a focus on routing and scheduling of maintenance jobs and rolling stock. However, the placement of stationing locations for parking of maintenance vehicles has not been studied yet for railway networks.

B. Simulation

Solving a FLP's is generally approached in one of two ways: by Linear Programming (LP) or simulation, with LP being more common. A simulation approach is more common for dynamic and probabilistic problems, while the LP approach is associated more with static problems. DES is generally used for analysis rather than optimisation. The problem in this research is highly dynamic and can benefit from an analysis through Discrete-Event Simulation (DES). Therefore, the use of the DES for optimisation is investigated by applying a Simulation-Optimisation (Sim-Opt) technique.

C. Optimisation

In such an technique, the simulation and optimisation algorithm interact on a sequential basis: the simulation is run with an initial set of input parameter values, after which the optimisation procedure uses the output values of the simulation to evaluate the corresponding performance. Based on the evaluated performance, the optimisation algorithm determines a new set of input values for the simulation *(Joines et al., 2001)*.

Optimisation algorithms can be divided into two categories: exact algorithms and heuristics. Exact algorithms are guaranteed to find the optimal solution in a finite amount of time. For large or very complex problems (e.g. NP-hard), this might become infeasible or at least impractical. Heuristics is a problem solving technique which bypasses this issue by directing a search through the solution space and limiting the computation time or number of iterations. This means that a subset of the solution space is evaluated. As a result, heuristics trades optimality and completeness for speed *(Maringer, 2003)*. A heuristic solving technique is selected for the optimisation in this research.

D. Simulated Annealing

Simulated annealing is a commonly used meta-heuristic and depending on the problem, one of the most efficient heuristic technique *(Maringer, 2003)*. A neighbourhood function constructs candidate solutions iteratively and worse solutions are accepted based on a probability determined through a cooling schedule. Annealing (SA) was first proposed by *Kirkpatrick et al. (1983)* as a stochastic optimisation technique. *Qin (2012)* incorporated three possible mutations into the SA neighbourhood function for solving the FLP. *Qin (2012)* solves a FLP of similar size to the problem of this research. Due to the similarities, the Sim-Opt technique and SA search method used by *Qin (2012)* are applied to solve the problem of this research.

III. CURRENT SYSTEM

A. Network

The Dutch railway system consist of about 7200km of track and 7000 switches. Besides normal track and switches, there are also other objects like bridges, level crossings, the overhead contact system, signals, stations, etc. All these objects need regular maintenance and intermittent renewal. ProRail defines a set of points in the network, servicecontrol points (Dienstregelpunten (DRP)), which are relevant for locating all processes of the train service, rolling stock handling and personnel services

B. Parking Locations

The Dutch railway system has various yards or freight railway yards throughout the county. Yards can be used to park, service or inspect freight, passenger or maintenance trains. The capacity used for stationing maintenance trains usually lies amid a yard where also other types of train are stationed. Every year, contractors request the amount of capacity they expect to need for stationing their equipment. Three types of parking locations can be identified:

- Gathering locations (Verzamel locaties): Location where supplies and vehicles are gathered, which can then be driven towards the project location.
- Stationing locations (Parkeer locaties): Locations where wagons and materials are prepared for a project in the immediate vicinity. These can be maintenance, new construction or replacement projects. (These activities can also take place on gathering locations)
- Preserving locations (Instandhoudings locaties): Locations where equipment and small stock is set up and deployed for maintenance and failures. (These activities can also take place on gathering locations and stationing locations)

The total amount of usable capacity adds up to 59349m and exceeds the total maintenance fleet length of 22km considerably. However, contractors calculate a buffer into the requested parking capacity to be able to make shunting or executing other movements smoother. Due to this buffer, contractors request a capacity of at least two times the size of the maintenance train fleet. The minimum parking capacity thus lies near 44km. Contractors have made agreements among themselves to interchange the assigned capacity throughout the year. Meaning that, throughout the year, maintenance trains leave certain parking locations and can enter other ones, which are technically reserved for another contractor.

TABLE I

OVERVIEW OF THE ESTIMATED TOTAL TRAIN LENGTH PER CONTRACTOR.

[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

C. Trains

Obtaining an overview of the maintenance train fleet is a complicated problem. Contractors can operate in the rest of Europe, meaning that the fleet is multiple times larger than if it would be to only service the Dutch railway system. Furthermore, trains are not bound to a single country, meaning that a train that has worked on a project in the Netherlands in a certain year might end up in different country and another train will be driven to the Netherlands if a new, similar project, arises. Furthermore, trains do not always consist of the same combination of wagons and locomotives. As a result the train fleet of the different contractors working on behalf of ProRail is not known to ProRail precisely. Therefore, a fictional fleet of trains has been constructed and used. This fleet is confined to the Dutch railway system only and will thus stay in the system. Based on expert judgement, a distribution defining the occurrence of length for the trains was used to construct the fleet (Table I).

D. Infrastructure Costs

Costs for infrastructure can be divided into two categories: operating costs and renewal costs. A part of the operating costs, which can be measured directly when changing the amount of capacity, is the infrastructure maintenance costs. The infrastructure on parking locations consists of track and switches and is subject to regular maintenance. Infrastructure is also subject to intermittent renewal. Some parking locations have been renewed more recently than others, and as a result, some parking locations have a high write off value if they are removed from the system in the near future. The estimated renewal costs and estimated lifespan will be used to calculate the effective yearly expenditures for a parking location when it is kept in use. The estimated yearly costs for maintenance and renewal for switches and track is specified in Table II.

E. Travelling Costs

Travelling costs can be subdivided into four aspects: staff, energy, vehicle maintenance and vehicle depreciation.

TABLE II

ESTIMATED YEARLY COSTS FOR MAINTENANCE AND RENEWAL FOR SWITCHES AND SIDE TRACK.

Infrastructure type	Maintenance	Renewal
Side Track (per km)	1000	11667
Switch	1000	Ranges from 1167-6667

Generally, only one engine driver is needed to drive a train. The rest of the personnel needed for a project arrives by their own means of transport. As a result, the contribution to the hourly rate of a maintenance train during travelling can be limited by the hourly cost for the employer of a single engine driver. In consultation with experts, this hourly cost has been set to be 75 euros. Maintenance trains generally reach speeds of up to 80km/h. However, due to intermittent stops, direction changes and run-up time, it is decided to use an average travelling speed of 40 km/h for every train in the system. The diesel consumption for a typical diesel hydraulic locomotive is set, in consultation with experts, to 3 litre per kilometre at normal speed under normal load. The diesel price has been taken to be around 1 euro per litre, resulting in a rough estimate of 3 euros per driven kilometre. Contractors recon that the wear on parts of the train as a result of normal travelling, is negligible compared to the wear on parts during a project. The same is true for vehicle depreciation. Vehicle maintenance and vehicle depreciation are therefore not taken into account in this research.

IV. SIMULATION-OPTIMISATION METHODOLOGY

The problem of this research will be tackled using a simulation-optimisation approach. The current set of parking locations will be altered by including and excluding parking locations from every simulation iteration. A DES and an optimisation stage interact to find a near-optimal solution to the problem. The simulation-optimisation interaction is illustrated in Figure 1. The entire framework has been coded using Python.

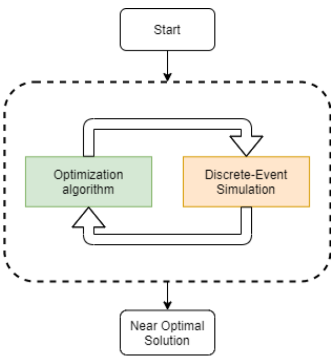


Fig. 1. Flow diagram of simulation-optimisation interaction.

The simulation will be done in chronological order to facilitate the dynamic pseudo random allocation of maintenance trains to parking locations. The DES consists of two passive components: the railway network and parking locations, and two active components: projects and trains. The railway network is represented as a graph of nodes and edges. Every DRP in the Dutch railway system is represented by a node in the graph. If there is a railway connection between any two DRP's, this connection is represented by an edge in the graph. All parking locations have information on where they are located in the network via the corresponding DRP name. The locations of the parking locations are fixed, but the aspect that changes from iteration to iteration is the inclusion of the parking location in the system. The simulation will use information on the planned maintenance projects in the railway system for one year. There are 1216 projects to be simulated for the year 2019. A set of trains to be used in the simulation is created using crude assumptions on the length per train, the number of trains per contractor and the average speed. From this small dataset, the average train length is found to be 42m. The total length of the maintenance train fleet is set to be 22km. From this, it is calculated that there are 524 train units to be generated. To assign the length to every train unit, a uniform distribution between 22 and 62 metres is applied. The share of total train fleet length per contractor is used to randomly assign contractors specific projects to the individual trains in the fictional train fleet.

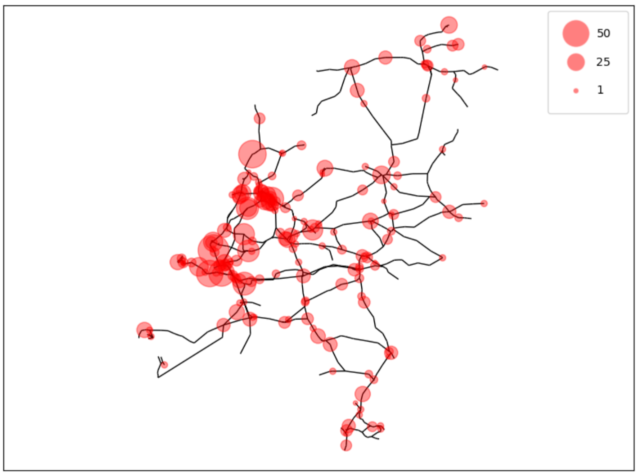


Fig. 2. Heat-map of project locations.

A. Optimisation algorithm

The optimisation algorithm has two tasks: evaluating the the output from the simulation and realising a new solution to the problem for the next iteration of the simulation. The simulation and optimisation interact to find a near-optimal solution to the problem. Due to the large size of the solution space the optimisation algorithm will use the heuristic technique Simulated Annealing to search through the solution space. The search algorithm will determine a neighbouring solution through the use of a neighbourhood function. For this research, two different search algorithms are implemented: A Random Search Algorithm (RSA) and a Utility Level Search Algorithm (ULSA), which takes information from the utility level of the parking locations during the previous simulation iteration

The neighbouring function of the RSA can execute one of the following operations.

- If the number of included parking locations does not exceed the maximum number of included parking locations, then, from the previously evaluated solution, an excluded parking location is randomly selected and included in the candidate solution.
- If the total parking capacity of all parking locations is not lower than the minimum parking capacity, then, from the previously evaluated solution, an included parking location is randomly selected and excluded in the candidate solution.
- From the previously evaluated solution, one excluded parking location is included in the candidate solution and one included parking location is excluded in the candidate solution.

Only one of these operations can be executed per iteration and every operation has an equal chance of being executed.

The neighbouring function of the ULSA can execute one of the following operations.

- If the number of included parking locations does not exceed the maximum number of included parking locations, then, from the previously evaluated solution, the most heavily used parking location is found. An excluded parking location, which lies closest to the most used parking location, is selected and included in the candidate solution.
- If the total parking capacity of all parking locations is not lower than the minimum parking capacity, then, from the previously evaluated solution, the least used parking location is selected and excluded from the candidate solution.
- From the previously evaluated solution, an excluded parking location, closest to the most used parking location is included in the candidate solution and the least used parking location is excluded in the candidate solution.

The usage level is evaluated based on the amount of departures and arrivals of a parking location, as well as its capacity. A parking location with a relatively high number of departures and arrivals compared to its capacity is therefore assumed to lay in a strategically beneficial location within the network.

B. Simulated Annealing

After a neighbouring solution is generated by the neighbourhood function, the DES evaluates the performance of that solution. The SA process accepts better and worse solutions to the problem with a certain probability. This probability is a function of the cost difference between the candidate solution and the intermediate accepted solution and the temperature. The temperature lowers according to a cooling schedule. The most common cooling schedule follows an exponential decay curve. The shape of this curve is determined by the starting temperature T_0 and the cooling factor α.

Parameter	Value
T_1	3e8
T_f	2e3
α	0.90
β	1.02
k_0	5
k_{max}	49

TABLE III
SIMULATED ANNEALING PARAMETER VALUES USED FOR THE RSA AND ULSA.

k_{max} is the maximum number of iterations at each temperature value. Usually, k_{max} is constant throughout the cooling process. However, *Frausto-Solis et al. (2007)* applies a variable number of iterations per temperature cycle with improved performance. In this case, k is equal to k_0 at the start of the cooling schedule and increases to k_{max} at the final temperature step according to:

$$k(n) = k_0 \beta^n, \tag{1}$$

where β is the increment coefficient and is greater than one and n is the cooling step. This technique is implemented in the SA process of this research.

Table III gives an overview of the SA parameter values used in this research. These values are partially based on the work of *Qin (2012)*, found through experimentation and through the tuning technique proposed by *Frausto-Solis et al. (2007)*. The values are tuned to keep the calculation time of a single Sim-Opt below 1.5 hours.

V. EXPERIMENTAL RESULTS

A. Experiments

The experiments are divided into two categories: Sim-Opt's using the RSA and Sim-Opt's using the ULSA (Figure 4). Per search algorithm, there are two kinds of initial

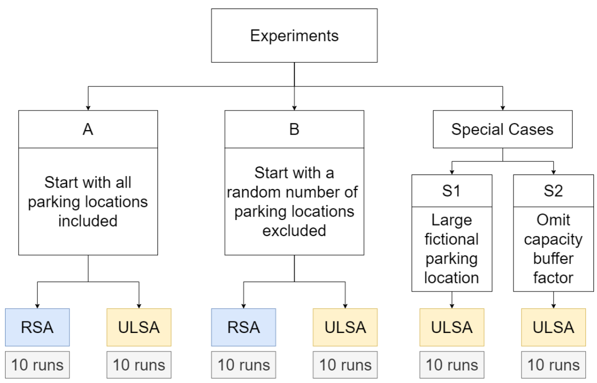


Fig. 3. Graphical representation of the experiments in this research.

solutions from which the optimisation will start the iteration process. Since this research is based on a case study, it is interesting to see the optimisation behaviour using the current real world situation as a starting point (Experiment A). This is especially relevant to the ULSA, since the utilisation level of the current real world situation provides a practical starting point. To increase the reachable solution area, the Sim-Opt will be started from multiple initial solutions, wherein a random number of parking locations (while respecting the minimum capacity needed) will be excluded (Experiment B). Per search algorithm and experiment, 10 simulation runs with different random seeds will be performed. More runs are preferable to obtain more robust results, but considering the computation time per simulation, this is set to be a workable, yet insightful amount.

In one special case (S1), a fictional parking location is added in the approximated centre of the network. This fictional parking location has a capacity of 44620m (the length of the train fleet plus the buffer of 100%). This scenario is simulated in order to assess the performance of a single parking location layout.

In the second special case (S2), the capacity-buffer is removed. The system has the same amount of trains (524) with a combined length of 22310m, but in this scenario there is no need to have a parking capacity buffer of 100%. Simulating this scenario will hopefully give insight into the impact of the buffer factor on the equilibrium between travel and maintenance costs of the parking locations in the system.

B. Results

As a baseline, the current system is simulated. The current system consists of all 49 parking locations being available for the parking of maintenance trains. The movements of 524 trains to and from 1216 projects are simulated for the year 2019. The mean values for the KPI's of 10 instances of this simulation are shows in Table IV.

C. Experiment A and B

Figure 4 shows a collection of graphs for the Sim-Opts of Experiments A and B. These graphs show the mean progress of every instance for all experiments. The total cost for the rolling best solution is plotted in the top left graph. The total cost is a result of the sum of maintenance and travel costs, which are plotted together in the top right graph. For all graphs the total costs decrease rapidly at the start and this decrease slows as the Sim-Opts progress. In experiment A all parking locations are included at the start of all instances of the Sim-Opt, which corresponds to ˜59km of available capacity. In Experiment B, a random number of parking locations was excluded for the initial iteration. The available capacity shows a steep decrease at the start of the Sim-Opts and best solutions are accepted near the minimum capacity boundary of 44310m for the rest of the iterations.

The trends for the ULSA graphs are similar to the trends of the RSA; the maintenance costs decrease more than the travel costs increase, the number of included parking locations decreases and the capacity of the best solutions end up near the minimum capacity needed to house all trains plus the buffer size of 100%.

The best solution of every Sim-Opt instance holds information on which parking locations are included or excluded. The best solutions of all 10 instances of every Sim-Opt are combined and represented in Figure 5 for Experiments A and B to give insight into the inclusion- and exclusion frequency for the individual parking locations. Figure 5 shows the number of times the individual parking locations have been excluded in all 10 instances per experiment. It can be seen that some parking locations are included in all 10 best solutions and some are excluded in all 10 best solutions.

It can be seen that more parking locations are excluded in all ULSA runs compared to the RSA runs, but this selection incorporates at least the parking locations that are excluded during the corresponding RSA runs. The ULSA finds that the parking locations which are included in all instances, are also included in all (or close to all) instances of the RSA.

The above mentioned observations indicate several things. First, some specific parking locations are favourable to either include or exclude from the solution. The consistency of which two different search algorithms find this set of parking locations indicate that these parking locations can be classified as respectively more, or less valuable within the system. Secondly, the ULSA finds solutions which are more alike, which also suggests the ULSA finds solutions with more confidence than the RSA.

There is no notable difference between the final outputs of Experiment A and Experiment B. It can therefore be concluded that the SA process in both experiments have had enough time to find similar solutions. The results of both Experiment A and B show that the ULSA converges sooner than the RSA. Furthermore, the spread in the best solutions across all instances is tighter when applying the ULSA and the ULSA finds solutions which are more alike across all instances. These findings indicate that the ULSA gives more robust solutions when compared to the RSA and makes the SA process more efficient.

D. Experiment S1

In this experiment, a fictional parking location is added to the system. First, a single simulation is run wherein the new fictional parking location *Utvr* is the only included parking location in the system. All trains reside, dispatch and return to *Utvr*. Table V gives an overview of the KPI's for this scenario, based on 10 simulations.

It is interesting to see the outcome of running the Sim-Opt using the ULSA with the possibility of including *Utvr* into the system. From Figure 6 it becomes clear that the ULSA has excluded *Utvr* from all 10 instances. Even though the maintenance cost and travel cost are lower when *Utvr* is the only included parking location, the search algorithm does not find this solution. In Experiments A and B, the available capacity for the best solutions bottomed out at ˜46km and ˜22 parking locations. *Utvr* mimics the capacity of these parking locations. So when this parking location is included during a certain iteration in the Sim-Opt, it would take a substantial number of iterations to close enough other, smaller, parking locations to compensate for the giant sudden maintenance

TABLE IV

MEAN RESULTS OF ALL BEST FOUND SOLUTIONS PER EXPERIMENT.

Experiment	Capacity [m] (SD)	Locations [#] (SD)	Travel [mln €] (SD)	Maintenance [mln €] (SD)	Total cost [mln €] (SD)
Current System	59349 (0)	49 (0)	0.983 (0.0146)	2.65 (0)	3.64 (0.0146)
A (RSA)	44920 (130)	22 (3.41)	1.07 (0.0294)	1.88 (0.0496)	2.95 (0.0305)
B (ULSA)	44789 (308)	22 (1.47)	1.07 (0.0183)	1.84 (0.0200)	2.91 (0.0127)
B (RSA)	45013 (255)	22 (4.10)	1.07 (0.0171)	1.88 (0.0247)	2.95 (0.0169)
B (ULSA)	44936 (294)	22 (1.28)	1.06 (0.0135)	1.84 (0.0089)	2.91 (0.0124)
S1 (ULSA)	45092 (579)	22 (4.05)	1.07 (0.0182)	1.88 (0.0334)	2.95 (0.0336)
S2 (ULSA)	23757 (394)	22 (5.85)	0.973 (0.0455)	1.00 (0.0589)	1.97 (0.0167)

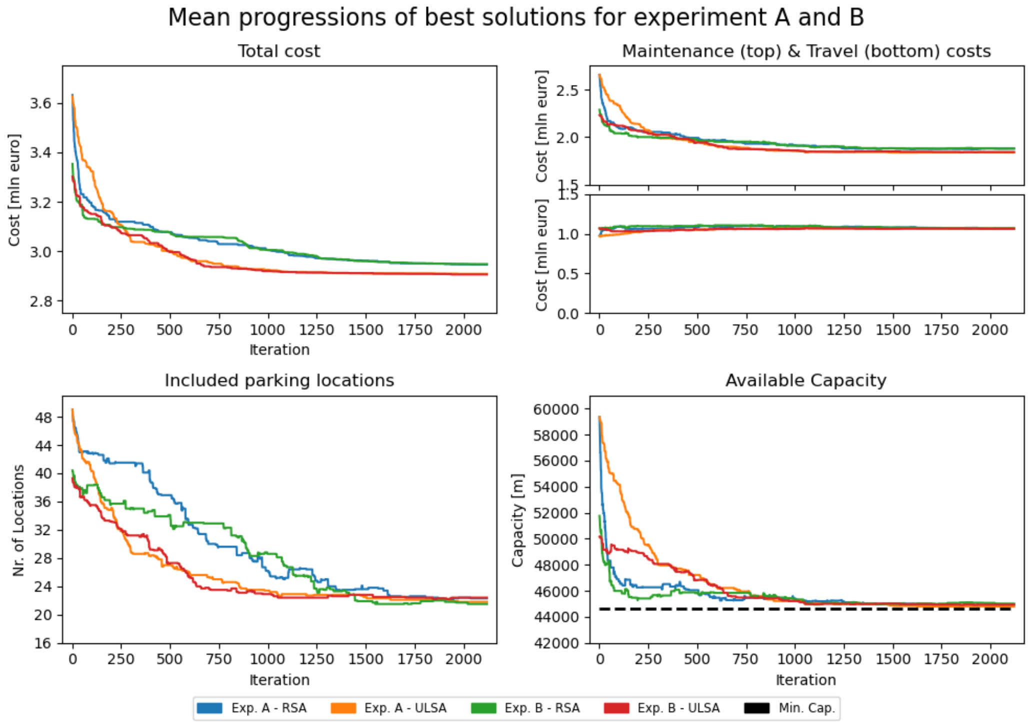


Fig. 4. Mean progressions of best solutions for experiment A and B.

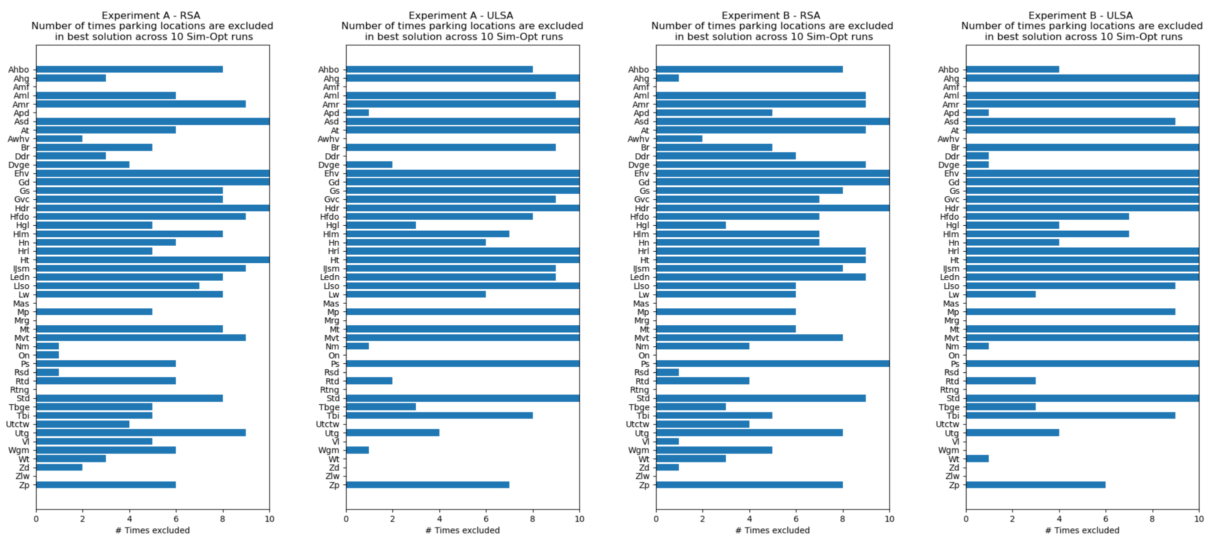


Fig. 5. Number of times parking locations are excluded in best solution across 10 Sim-Opt runs for Experiments A and B.

TABLE V

EXPERIMENT S1 - SYSTEM PERFORMANCE WITH ONE FICTIONAL PARKING LOCATION (MEAN VALUES BASED ON 10 SIMULATIONS).

KPI	Capacity [m]	Locations [#]	Travel [mln €]	Maintenance [mln €]	Total cost [mln €]
Mean	45000	1	0.904	1.83	2.74

cost increase. In short, this heuristics method is not suited for handling this scenario. The possible cost jumps are too high in comparison to the permutation size.

E. Experiment S2

It can be observed that all best solutions for all instances of Experiment A, B and S1 have an available capacity near to the minimum set capacity. Experiment S2 investigates the system behaviour when relaxing this boundary. The number of times parking locations are excluded in all instances is higher compared to the number of times parking locations are included in all instances. On average, more parking locations are closed in the best solutions when the capacity buffer is removed. This is in accordance with the fact that there is less capacity needed. Lowering the amount of available capacity to the bare minimum seems to have little effect on the travelling costs. Interestingly in this experiment, *On* is closed considerably more often and *Lw* is opened more often when compared to the other experiments. *On* and *Lw* are two parking locations that serve the northwest part of the Dutch railway network. With the removal of the capacity buffer, the ULSA finds that On is used relatively less than *Lw*. The amount of trains needed in that area can suffice with the capacity offered by *Lw* alone. In other experiments the number of trains needed in that area require more capacity than the 55% of *Lw's* total track length and thus it is cheaper to open *On*, which is several times larger. This example implies that the size of a parking location can overrule the strategic positioning to some extend.

VI. CONCLUSION

The new neighbourhood search algorithm ULSA is tested against the random neighbourhood algorithm RSA used by Qin (2012). The evaluation is performed for both search algorithms using multiple instances of the Sim-Opt. The results of this evaluation show that the ULSA converges sooner than the RSA. Furthermore, the spread in the best solutions across all instances is tighter when applying the ULSA. When looking at the details of the solutions, it can be seen that the ULSA finds solutions across all instances which are more alike. These findings indicate that the ULSA gives more robust solutions when compared to the RSA and makes the SA process more efficient.

According to the simulation used in this research, ProRail can potentially reduce the annual cost related to the travelling and parking of maintenance trains by ~20%. A large part of this improvement can be achieved by rethinking the parking location strategy. This research found that the best solutions in terms of overall cost includes on average 22 parking locations, which can reduce the maintenance cost by ~30%. This is in stark contrast with the current situation, where 49

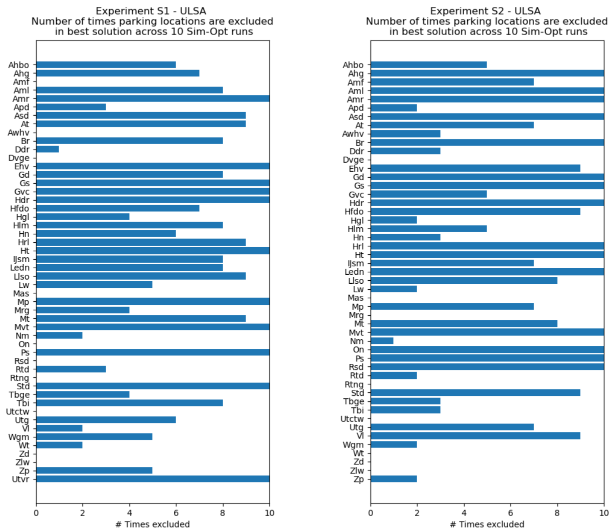


Fig. 6. Number of times parking locations are excluded in best solution across 10 Sim-Opt runs for Experiments S1 and S2.

parking locations are used for maintenance trains. The total travel costs show a relatively small increase of ~9%. The capacity for the optimal solution is close to the minimum capacity (including a capacity buffer of 100%) requested by the contractors. This implies that the increase in travel costs does not justify the use of more capacity than twice the train fleet length. The improved performance of the ULSA can encourage ProRail to use the relative usage level as a metric for determining the value of a parking location.

The included parking locations in the Sim-Opts are spread across the network, but are preferably located near the project locations. The number of projects in a certain area of the network is related to the amount of infrastructure. As a result, parking locations near the outer part of the railway network are excluded more often and parking locations towards the centre of the railway network are included more often in the Sim-Opt. Special case S1, shows that the use of a single parking location, located in the centre of the railway network, would decrease the total cost considerably (by 24%). Such a situation is thought to be impractical, since the added travel time for contractors would reduce the effective labour time for the engineer.

Special case S2 shows that in an extreme scenario wherein no extra space is needed for shunting, any change in travel costs still does not justify the use of more capacity than the absolute minimum. The parking locations selected by the model are, again, located across the network. This implies that a nationwide coverage is important. However, S2 also shows that the size of a parking location can overrule the strategic positioning to some extend.

VII. Reflection and Recommendations

This research shows that DES is a feasible evaluation method for finding possible solutions to the FLP in a Sim-Opt approach. However, a lot of simplifications are needed to bring the calculation time down. Furthermore, the solution space of such a problem is relatively large. The optimisation process would benefit from a selection process, on top of the improved neighbourhood function applied in this research, in order to reduce the solution space.

A permutation size of one has been used in the neighbourhood search function, but a larger permutation size can reduce the number of iterations needed to find an optimum. Furthermore, a larger permutation size can counteract the limitations found in special case S2.

The results show that the capacity buffer is a prominent limiting factor. The buffer is meant to ease the shunting process on parking locations. Research should be done to get insight into the shunting process and a cost-benefit analysis should be performed in order to determine the capacity buffer size.

The unknown maintenance train fleet made the use of a fictional train fleet necessary. It is necessary for future research to find a way to specify the train fleet that operates and resides within the Netherlands. In order to do this, the dynamics with the rest of Europe should be investigated.

A big limitation in this research was the amount of data available. As a result, only one year of maintenance projects was simulated. Railway infrastructure lifespan exceeds this period greatly. Future research should thus be based on multiple years of data, which at least covers a sizeable period of the lifespan of infrastructure parts.

The simulation in this research allowed trains to reside at any available parking location. In real life however, some contractors are assigned to specific parking locations due to any specific facilities needed or adjacent buildings. Any additional measurement trains also reside on a few specific parking locations in real life. This extra constraint should be added in future models.

The costs that contractors make during journeys to and from project locations is incorporated into the value of a yearly contract. However, the exact numbers are not known and thus this research used a simple estimation. The portion of the contract value which covers the related travel expenditures should be specified more clearly. Furthermore, the practical inconvenience of longer train journeys on a working day for contractors should be translated into a monetary metric in order to make a solid cost-benefit analysis in the future.

The results show that the parking locations are preferably placed near the project locations. The assumption that the amount of infrastructure in a certain vicinity is a measure for the amount of projects, leads to a recommendation for the methodology in future research: A mathematical model whereby the amount and type of infrastructure is evaluated on the network nodes and edges can give an exact solution to the problem. Such a solving method can be further explored in future research.

References

[1] Derya Celik Turkoglu and Müjde Erol Genevois. A comparative survey of service facility location problems. Annals of Operations Research, 292, 09 2020. doi: 10.1007/s10479-019-03385-x.

[2] Leon Cooper. Location-Allocation Problems. Operations Research., volume 11. 1963. ISBN 331-343. C.D. Dao, Andreas Hartmann, and Rob Basten. Maintenance scheduling for railway tracks under limited possession time. Transportation engineering journal of ASCE, 144, 04 2018. doi: 10.1061/ JTEPBS.0000163.

[3] Mark S. Daskin. Network and Discrete Location: Models, Algorithms, and Applications. 2013. ISBN 978-1-118-53696-4

[4] Juan Frausto-Solis, E. F. Román, David Romero, Xavier Soberon, and Ernesto Liñán-García. Analytically tuned simulated annealing applied to the protein folding problem. pages 370–377, 2007.

[5] Daniel Erlandsson Fredrik Persson. Solving location problems with simulation modelling - a case from the construction industry. 2013 Winter Simulation Conference, 2013. doi: 978-1-4799-2076-1

[6] Hasan Hosseini nasab and Fatemeh Mobasheri. A simulated annealing heuristic for the facility location problem. Int. J. of Mathematical Modelling and Numerical Optimisation, 4:210 – 224, 01 2013. doi: 10.1504/IJMMNO.2013.056532

[7] Jeffrey Joines, R. Barton, Keebom Kang, Paul Fishwick, James Swisher, Sheldon Jacobson, Paul Hyden, and Lee Schruben. A survey of simulation optimization techniques and procedures. 01 2001.

[8] M.L. Kho. Facility location optimization and maintenance routing in a railway network, 2016.

[9] Scott Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. Science (New York, N.Y.), 220:671–80, 06 1983. doi: 10.1126/science.220.4598.671.

[10] Tomas Lidén. Railway infrastructure maintenance - a survey of planning problems and conducted research. Transportation Research Procedia, 10, 07 2015. doi: 10.1016/j.trpro.2015.09.011.

[11] Siqi Liu, Boliang Lin, Jiaxi Wang, and Jianping Wu. Modeling the multi-period and multi-classificationyard location problem in a railway network. Symmetry, 10:135, 04 2018. doi: 10.3390/sym10050135.

[12] D. Maringer. Portfolio management with heuristic optimization. 8, 01 2003. doi: 10.1007/b136219

[13] Jeroen van Rutten ProRail, Marcel Drenth. Referentiekader: Opstellen van materieel ten behoeve van werkzaamheden voor beheer., 2020.

[14] Jin et al. Qin. Combined simulated annealing algorithm for the discrete facility location problem. 01 2012. doi: 10.1100/2012/576392.

[15] Ragheb Rahmaniani, Abdolsalam Ghaderi, and Mohammad Saidi-Mehrabad. A simulated annealing algorithm within the variable neighbourhood search framework to solve the capacitated facility location-allocation problem. Journal of Optimization in Industrial Engineering, 5:45–54, 03 2012.

[16] C.S. ReVelle, H.A. Eiselt, and M.S. Daskin. A bibliography for some fundamental problem categories in discrete location science. European Journal of Operational Research, 184(3):817 – 848, 2008. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2006.12.044.

[17] Masoud Hekmatfar Reza Zanjirani Farahani. Facility Location - Concepts, Models, Algorithms and Case Studies. Springer-Verlag, 2009. ISBN 978-3-7908-2150-5

[18] Robert G. Sargent. Verification and validation of simulation models. Engineering Management Review, IEEE, 37:166 – 183, 01 2011. doi: 10.1109/WSC.2010.5679166.

[19] G. Wainer and Pieter Mosterman. Discrete-Event Modeling and Simulation: Theory and Applications. 04 2016. ISBN 9781315218731. doi: 10.1201/9781315218731.