PRORAIL INCIDENT HANDLING

An Incident Handler Coverage Allocation Optimization Model

Supervisors University of Twente

Dr. Ir. E.A. Lalla

Dr. Ir. A.G. Leeftink

Dr. Ir. J.M.J. Schutten

Supervisors ProRail

W. van Engelen

L. van den Hoven

P. Snoeks

Author

J.I. Schenk (Jurgen)

January 22

UNIVERSITY OF TWENTE.

ProRail

Management Summary

Railway incidents occur randomly and can have a significant impact on train operations. ProRail, owner of the Dutch railway tracks, has a special department that solves incidents on railway tracks. This department is called the incident handling department (ICB). The ICB department defines incidents as undesirable events involving a train with disruptive effects on the rail traffic system or on services of participants in the rail traffic system, such that continuation is endangered or already hindered. These events immediately result in imminent or already occurred injury/damage to people, animals, goods or the environment. Contractors, in general, solve all other incidents not involving trains. Different incidents require different specialised incident handlers. Currently, the planning is not based on computational intelligence and is not specialism-specific, meaning ProRail currently plans a general incident handler. However, ProRail wants to move to specialisms rather than general incident handlers. Besides, ProRail is looking to improve its planning and scheduling by considering information on the allocation of specialism-specific incident handlers to cover incidents at a threshold coverage level. Therefore, we define the goal of this project as:

Create an optimization model that guarantees a threshold coverage by allocating specialization-specific incident handlers based on historical data to support scheduling.

Context

The specialisms we consider in this research are the general leader (AL) and the ICB team (ICB). The AL is responsible for the operational management and coordination of incident handling. The ICB team is responsible for actually solving incidents. Incident handling takes on average 10 to 20 % of the working time of an incident handler. Therefore, incident handlers (AL and ICB) perform preventive and secondary tasks for the remainder of the working time. As soon as an incident occurs and an incident handler gets called in, the incident handler stops with his current preventive or secondary task and hurries to an incident. Currently, ProRail works with two shifts on weekdays and one during a weekend day. Almost all shifts include on-standby time where an incident handler is at home but can get called in for incident handling. The Port of Rotterdam, Kijfhoek (freight shunting yard near Rotterdam) and Schiphol airport are specific ProRail locations that do not have standby shifts because of the importance of these locations.

Incidents occur at random (unpredictable) moments, and one incident is often completely different from another. To adequately respond to an incident with the correct specialisms and equipment, ProRail uses an incident classification system. As soon as an incident is called in, an incident is given a classification, and the appropriate incident handlers respond to the call. There are two incident classification systems. The first is the Train Incident Scenario (TIS), and the second is the incident labelling. The latter gives a more detailed description of the incident. The datasets available for this research, all limited to the period between the 24th of June 2017 and the 26th of March 2021, include the TIS and incident labels. Based on discussions with experts of ProRail, we exclude (filter) several TIS scenarios and incident labels that do not require the deployment of incident handlers from the data. Besides, we only look at 2018, 2019 and 2020 because 2017 and 2021 are incomplete. Also, one of the datasets represents the deployment of incident handlers. Unfortunately, the deployment dataset does not cover the same time period, because ProRail started registering deployment from 2020 onwards. Therefore, we extrapolated the deployment data per incident label to the incident dataset.

Method

In the past few decades, a great deal of research has been done on positioning emergency services, predominantly focussed on and referred to efficient positioning of emergency medical services. Therefore, it is not unexpected that we find many optimization models proposed for solving the capacity allocation problem to guarantee coverage in the literature. For our problem, the Location Set Covering Problem (LSCP) by Toregas, Swain, ReVelle, and Bergman (1971) and the TIMEXCLP model of Repede and Bernardo (1994) are two very relevant models. We combine the LSCP and TIMEXCLP model to create a model that minimizes the number of incident handlers over different time instances by allocating them such that we achieve the constraint of a threshold coverage. In this way, the number of incident handlers is unrestricted and optimized by the model. Covering models typically use a binary coverage view, meaning a node is either covered or not. There exist functions that change this binary view to a gradual coverage view. We extend the developed model with a gradual coverage function where coverage decreases when distance increases away from an incident handler using a step-wise coverage function.

Location covering problems are typically optimization problems that belong to a complex class of combinatorial optimization problems (Rajagopalan, Saydam, & Xiao, 2008). Because of the complexity of such problems, various metaheuristic search methods have been developed to find near-optimal solutions in reasonable computational time. Of these metaheuristic search methods, simulated annealing is an appropriate method. Simulated annealing is a method that has been successfully implemented to solve covering models. It is a method that is usually easily implemented. Also, it generally requires less computational effort than more sophisticated procedures such as tabu search and genetic algorithms (Galvão, Chiyoshi, & Morabito, 2005).

The simulated annealing algorithm obtains solutions based on a deterministic input. However, incidents do not occur on a deterministic basis. Instead, they occur randomly. Due to this stochastic element, a good solution obtained with the SA algorithm using deterministic input might not be that good when considering stochasticity. Therefore, it is vital to analyse the performance of a solution in a stochastic environment. With a stochastic analysis, we can determine the robustness of a solution. Therefore, to analyse this stochastic performance, we use a simheuristic. Simheuristic algorithms are often used to simulate real-world problems under uncertain conditions (Chica & Juan 2017). The simheuristic algorithm uses simulation to allow stochastic scenarios to be evaluated for a fixed solution. In this way, we can analyse the feasibility of a solution under uncertain conditions. We can select the most robust solution of our solutions when doing so for various solutions. To do so, we fix the allocated employees in the solutions and simulated different incident environments where incidents occur more often or less often. In each simulation run, we calculate the coverage and after all simulation runs of one solution, we analyse the coverages. The simheuristic algorithm we are using to analyse the stochasticity is Monte-Carlo simulation. This technique allows to simulate many different instances, every time slightly adjusted. We want to evaluate the performance of different solutions by simulating stochasticity in the incident scenarios, and with this technique, we can. The simheuristic with Monte-Carlo simulation is proven to be efficient and reliable (Lalla-Ruiz, Heilig, & Voß, 2020).

Results

We analyse in total 40 different scenarios. We consider two specialisms, ten different time periods as input data and a weekday or weekend day. The final results, which are the most robust solutions to every scenario obtained with the simheuristic, let us see that the main areas of focus during the week are Rotterdam, Amsterdam and the theoretical triangle Zwolle – Enschede – Arnhem, followed by the

South (Limburg) and the North of the country. Also, the required number of employees on a weekend day decreases by around 19% compared to the number of employees required during a weekday. This decrease in the number of employees mainly occurs in the North of the Netherlands, as we also show in Figure 1, where we see the allocation of employees during a weekday and weekend day based on incident data of the years 2018 and 2019. The graphs in this figure represent the cumulative allocation of employees of all hours on the day.

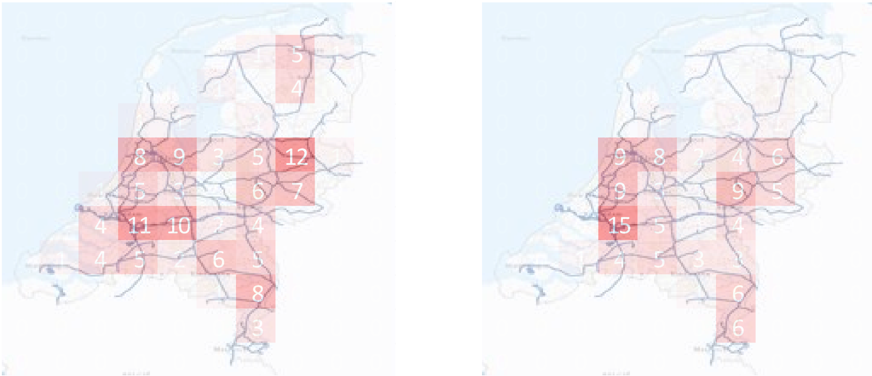


Figure 1: Allocation of employees during a weekday (left) and weekend day (right) based on incident data of 2018 and 2019

For all results, we considered a threshold coverage of 80%. We determined this value in consultation with ProRail. However, ProRail is also interested in what happens if we increase this threshold coverage level. Figure 2 shows the effect when increasing the threshold coverage level. The figure plots the increase in the required number of employees compared to the situation where the threshold coverage level is 80%. We see that the number of employees needed grows when increasing the threshold coverage value. In this way, reaching a 95% coverage level is unrealistic as this requires 64% more employees, which is, in reality, a significant number of extra employees and not possible for ProRail. However, we assume that these percentages are reliable when requiring a higher coverage level for a small specific region. In that case, increasing the number of employees in that region only is possible.

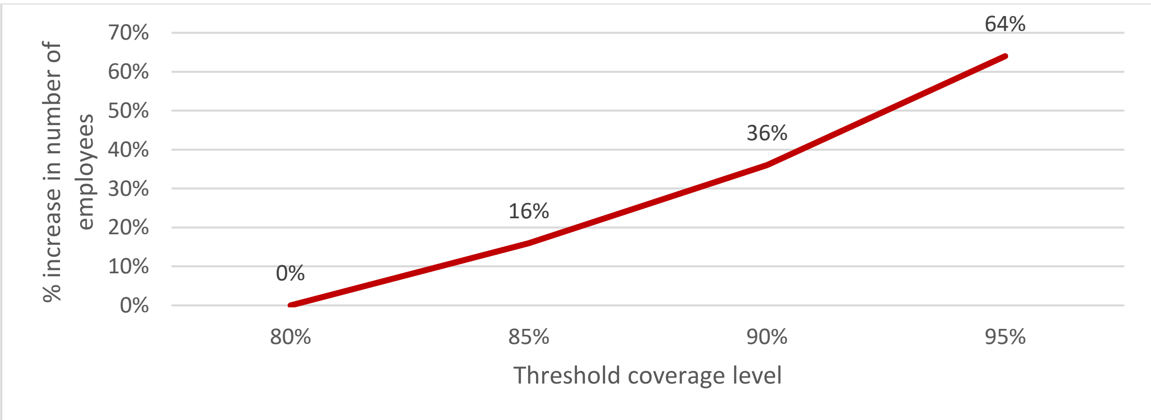


Figure 2: Effect on number of employees when increasing the threshold coverage level

Recommendations

This research provides insight into the number of employees required per hour of the day and the location of those employees. However, this is influenced by several assumptions. First, we extrapolated deployment data per incident label. We recommend analysing the deployment per incident label in more detail. Besides, we do not consider travelling times and only look at the starting moment of incidents. We recommend doing further research to develop a model that considers both travelling times and incident duration. This will create a more detailed model that can be used at a lower planning level.

If we look at the final results on the distribution of incident handlers across the day and compare this with the current working shifts ProRail uses for its incident handlers, we see possible improvements. We recommend considering the shift-change from morning shift to afternoon shift of the ICB team to be 1 hour earlier, from 13.00 to 14.00. Besides, we recommend looking at variable shift hours, where employees do not start and end their shifts simultaneously. This would improve coverage, as employees are more spread out during the day. Lastly, we recommend considering if the employees in the afternoon shift should all go on standby during the night. With only half of the employees on standby during the night, enough employees remain to cover demand.

Preface

Hereby I present you my master thesis ‘ProRail Incident Handling: An Incident Handler Coverage Allocation Optimization Model’. This is the final part of my Master Industrial Engineering and Management at the University of Twente and finalizes my time as a student. I would like to thank several persons who contributed to this thesis.

First of all, I would like to thank my company supervisors Wesley van Engelen, Linda van den Hoven and Pascale Snoeks of ProRail. Although I worked almost the complete duration of this project at home due to the COVID pandemic, they were always willing to help and quickly replied to any of my questions. I appreciate the open and informal working environment. Moreover, I would like to thank all other colleagues at ProRail who helped me during this project.

Furthermore, I would like to thank my supervisors Eduardo Lalla, Gréanne Leeftink and Marco Schutten from the University of Twente for their support. Eduardo, my main supervisor, always finds time to discuss my thesis. He encouraged me to bring this thesis to a higher level. Gréanne, my second supervisor, provided me with interesting ideas till the moment she went on maternity leave. Finally, Marco stepped in and gave me useful feedback that improved my thesis even further.

Finally, I would like to thank my girlfriend, family and friends for their support. You supported me to all ups and downs that came with this project and encouraged me to make the most out of it. I look forward to the further, with all its opportunities and challenges.

I hope you enjoy reading this report.

Jurgen Schenk,
Januari 2022

List of Figures

List of Tables

List of Abbreviations

AL	General Leader (Algemeen Leider)
BOA	Community Service Officer (Buitengewoon OpsporingsAmbtenaar)
CRG SO	Cargo specialist
GM	General Member Technical
ICB	Incident Handling (Incidentenbestrijding)
ILP	Integer Linear Program
IQR	Inter Quartile Range
LWB	Leader Safe Workplace (Leider Werkplek Beveiliging)
SA	Simulated Annealing
TC	Technician
TC SO	Technical Specialist
TIS	Train Incident Scenario (Treinincidentscenario)
TL	Team Leader
TM	Team Member
VNS	Variable Neighbourhood Search

Contents

1. Introduction

This report describes the graduation assignment conducted at ProRail for the master Industrial Engineering and Management at the University of Twente. The research focuses on capacity allocation optimization of incident handlers at the department of incident handling (ICB). This chapter introduces the research conducted at ProRail. First, we provide a company description and describe the ICB department in Section 1.1. Next, we define and motivate the problem in Section 1.2. Section 1.3 covers the research objective, including the research goal, scope and limitations. Section 1.4 states the research questions, and Section 1.5 finalises the chapter by describing the research design.

1.1. Company description

The Netherlands has one of the best railway networks in the world (Economy, 2019). Every day more than 1.2 million people travel by train, and 100,000 tons of goods are transported using a network of more than 7,000 kilometres of railway tracks (CBS, 2009). Transporting these numbers of people and goods requires a solid infrastructure. ProRail is the company responsible for this infrastructure. They renew stations to state-of-the-art stations, arrange safe journeys of passenger and freight trains and take care that all trains arrive on time, a challenging job on one of the busiest railways in the world (RailTech, 2019). In Figure 3, we see the complete infrastructure ProRail is responsible for.

One of the departments of ProRail is ICB. The ICB department acts when incidents occur on the railway network. They send employees, called incident handlers, to the location of the incident. The incident handlers ensure that the railway track is cleared as soon as possible after an incident. They are equipped with specialized vehicles that allow them to access every location, whether easy or hard to reach. The incident handlers work together with other parties in the handling of incidents. These parties could vary from public transport parties, which receive passengers from stranded trains, to the fire brigade when a fire has broken out. Contractors generally solve incidents that do not influence passengers. Contractors repair or renew parts of tracks when necessary. In addition, ICB takes measures to guarantee the safety of passengers, train personnel, residents and emergency services. Many incident handlers are also community service officers. They are authorized to take active action against vandals, copper thieves or track runners.

Figure 3: Overview railway network The Netherlands

1.2. Problem description

This project is conducted at the ICB department and focuses on allocating incident handlers across the country. Incident handlers are employees of ProRail who are responsible for solving incidents. Here, we define an incident as an undesirable event involving a train with disruptive effects on the rail traffic system or on services of participants in the rail traffic system, such that continuation is endangered or already hindered. These events immediately result in imminent or already occurred injury/damage to people, animals, goods or the environment. Contractors, in general, solve incidents not involving trains.

Besides solving incidents, incidents handlers also perform preventive and secondary tasks. Preventive tasks can vary from checking railway crossings to surveillance and active action against vandals, copper thieves or track runners on railway tracks and stations. Secondary tasks are, for example, doing maintenance on their vehicles or managing the materials.

Incident handlers are scheduled 24 hours a day divided over a couple of regions, overlapping with the Dutch provinces, to cover all tasks and solve incidents. This schedule includes different shifts for the week and weekend days. The number of incident handlers scheduled is based on common sense and does not include any computational intelligence (e.g., looking in historical data where incidents frequently occur or where surveillance is most effective).

Currently, every incident handler can solve every type of incident and every task coherent with that type of incident. That means that an incident handler can, for example, solve an incident in a technical manner but can also be responsible for a safe workplace during the solving of an incident or the handling of passengers. ProRail is currently reorganising this into specialisms such that incident handlers are better capable of performing their specific tasks.

For the project, ProRail is looking for nationwide capacity advice based on computational intelligence and specialism. To clarify, nationwide means not looking into regions anymore, but covering the whole country. Computational intelligence represents the use of historical incident data. Moreover, specialism means advising to allocate *X* technical handlers and *Y* community service officers to a location instead of *Z* incident handlers (without specialism-specific information). In this way, ProRail expects to create a more advanced and efficient schedule of incident handlers.

From this, we define the problem of this project as:

The allocation of incident handlers during a day is not specialization and location-specific and not based on a historical (incident) data to support accurate and precise scheduling of incident handlers.

1.3. Objective of research

In this section, we discuss the research goal. Besides that, we describe the scope and limitations of the research.

1.3.1. Research goal

This research aims to use an optimization model to allocate incident handlers across the country to optimize coverage to a threshold. The coverage optimisation coheres with the expected incidents in a region for an upcoming period. Therefore, the research goal is defined as follows:

Create an optimization model that guarantees a threshold coverage by allocating specialization-specific incident handlers based on historical data to support scheduling.

1.3.2. Research scope and limitations

The allocation problem can be very extensive and complex when including many aspects. As the time of a master thesis project is limited, the scope of the research is narrowed and limited. The scope and limitations are defined as follows:

Scope

- The optimization model includes allocating ICB personnel nationwide.
- A historical incident data analysis is part of the research. With the help of this data analysis, we create incident heatmaps and frequency tables. The data analysis focuses on the types of incidents, duration, frequency and location.
- The research includes looking at the influences of seasons, day of the week and day hours regarding incidents.

Limitations

- This research does not consider the operational scheduling of incident handlers; the outcome is only advice of the capacity required at locations at different timestamps on a day.
- The available data is limited to July 2017 till March 2021.
- It is not part of the research to deliver a tool, only the approach to obtain the results and capacity and allocation advice based on the results.
- We do not consider the costs of allocating personnel.
- We consider Euclidean travel distance from location X to location Y, thus without considering actual travel distances following roads.
- We do not consider incident duration in the model

1.4. Research questions

The project's research goal is defined, and we reach this goal by systematically answering the sub research questions and finally answering the main research question. First, the main research question is defined, followed by the sub research questions. The main research question is as follows:

How can optimization techniques be used to guarantee coverage by allocating specialization-specific incident handlers based on data?

Using an optimization model, we can obtain a suitable capacity allocation of specialization-specific incident handlers. We look if we should use an algorithm that identifies appropriate solutions in reasonable time as part of the solution approach. Four research sub research questions are defined to answer the main research question systematically. These questions will vary from understanding the current process and proposed methods in literature to solve capacity allocation problems to solution implementation, experimentation and evaluation. Lastly, we define questions for the conclusions and recommendations to maintain the same guideline throughout the report. The research sub-questions are explained below.

Analysing the current situation of the problem context

The first question focuses on the current allocation and scheduling process of incident handlers and the incidents themselves and helps understand the problem context. We identify the process's KPIs, constraints, and requirements and acquire and analyse available data. In Chapter 2, we cover the answer to this question.

How is the allocation and scheduling of incident handlers currently working?

- What are the characteristics of an incident, and how is an incident handled?
- How is the current capacity determined?
- What are the capacity allocation problem's KPIs, constraints, and requirements?
- What method is used at ProRail with regards to scheduling incident handlers?
- What data is available at ProRail concerning incidents?

Literature review and analysis

The second question focuses on identifying and understanding different optimization techniques proposed in the literature for similar capacity allocation problems. Various techniques are identified, explained and analysed on suitability for solving the capacity allocation problem in an incident-driven environment. By doing so, we create a strong foundation for the solution approach. We describe the literature review and analysis in Chapter 3.

What has been proposed in the literature for solving the specialism-specific capacity allocation problem to guarantee coverage?

- Which optimization models have been proposed in the literature for solving the capacity allocation problem to guarantee coverage?
- Which optimization techniques are proposed in the literature to solve allocation problems?
- What are the advantages and disadvantages of the optimization models and techniques proposed in the literature?

Design of solution approach

Following the first two questions, the third question relates to designing the solution approach and is covered in Chapter 4. We design a model and select optimization techniques to solve the model in a reasonable time to reasonable solutions. Besides, we identify metrics to analyse the solutions' performances and allocation strategies.

How should the solution approach be designed for the capacity allocation problem?

- What are the KPI metrics that can be used to analyse the performances for the capacity allocation solution?
- Which solution approaches are suitable for solving the capacity allocation problem in an incident-driven environment?
- Which allocation strategies should be considered in the solution approach?

Experimentation and evaluation

Once we finish the solution approach and design the model, we evaluate the model's performance. In this phase, the coherent research question focuses on the model's performance for different incident scenarios. Chapter 5 covers the experiments, results, and evaluation of the model.

How does the model perform for different scenarios of historical incidents?

- How do we validate the capacity allocation solution obtained from the model?
- What are the different scenarios and experimental setups that need to be considered to analyse the model?
- How does the model perform for the scenarios and experimental setup considered?

Recommendations and conclusions

To maintain a guideline throughout the report, we define questions that we answer in the recommendations and conclusions in Chapter 6. We define these questions as follows:

What can be concluded and recommended from the results of the experiments?

- What are the advantages and disadvantages of the model?
- What can be recommended to ProRail based on the results of the experiments?
- Which further research can be done following the results of this research?

1.5. Research design

We divide the research into four different phases, which all together form the research design. The research design systematically answers the main research question by sequentially solving sub research questions in each phase. The first two sub research questions correspond to the first phase, followed by one question per phase for the remaining three phases. This systematic approach is a proven method used to solve business problems (Heerkens, 2015).

The phases are as follows:

- Problem identification & analysis
- Solution generation & choice
- Solution experimentation
- Evaluation & implementation

Figure 4 schematically displays the phases, corresponding sub research questions and input required to answer the main research question.

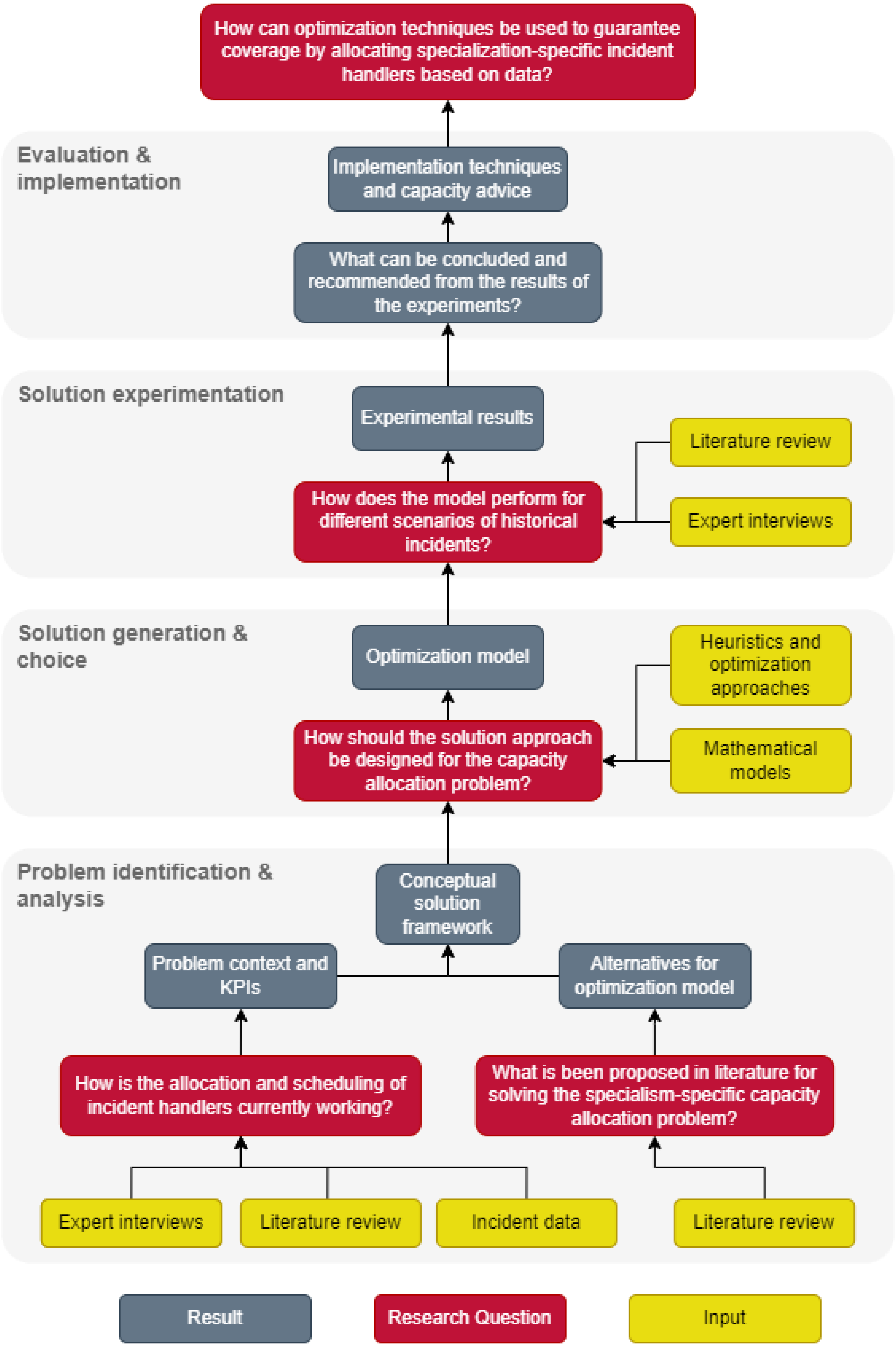


Figure 4: Graphical representation of the research design

2. Problem context

In this chapter, we provide a detailed description of the problem. First, we explain the incident handling department Section 2.1. Next, in Section 2.2, we describe the railway sections, followed by the incident classification in Section 2.3. Section 2.4 discusses the data regarding incidents and deployment available at ProRail. Finally, Section 2.5 concludes this chapter.

2.1. Incident handling department

The ICB departments main task is solving incidents. Incidents refer to disruptive effects on the railway network. We can visualise the effect of incidents on the train schedule by a bathtub model (Ghaemi, Cats, & Goverde, 2017), see Figure 5. The model includes three phases, where phase 2 represent the incident handling itself. However, it is important to note that ProRail describes the duration of an incident from the moment an incident is known at ProRail until the end of the incident handling and thus restarting the train schedule. The time between intake and the start of phase 2 represents travel time to the incident location and waiting time. Here, waiting time could vary from, for example, clearance after police investigations when a train has hit a person or clearance from the railway operator who reroutes all scheduled trains such that the track is clear. For the project, it is interesting to look at the time of intake because allocating incident handlers to highly demanded areas mainly reduces travel time as the incident handlers are assumed to be close to the incident location. Different types of incidents also require different types of incident handlers, from here on called specialisms. We explain these specialisms in Section 2.1.1. In general, incidents require a team of 4 incident handlers.

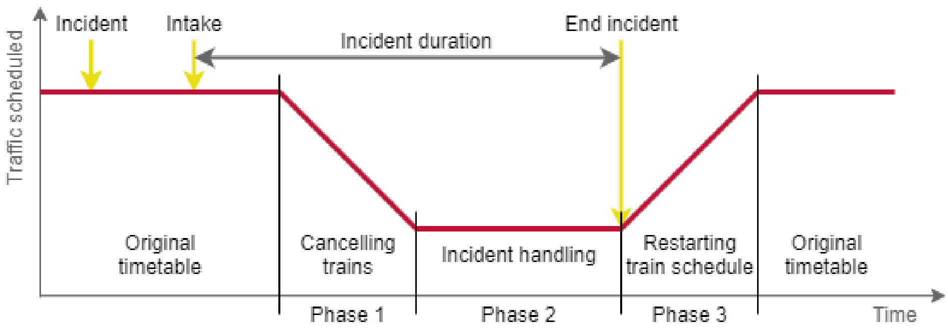


Figure 5: Bathtub model indicating the effect of incidents on traffic over time

To solve incidents quickly, ProRail works with a smart alarming system. That means that the location of incident handlers is always known. The system notifies the nearest incident handlers as soon as an incident occurs. However, this system does not advise the capacity needed on a day or hour. It only calls the nearest incident handler to an incident. Therefore, we are interested in the number of incident handlers required on a specific hour.

2.1.1. Specialisms

An incident handler is a general term. In fact, the ICB department works with specialisms to increase the specialised knowledge in specific themes. ICB is divided into five themes in which several specialisms exist. Figure 6 shows the themes and underlying specialisms (Dutch abbreviations used in figure). Each theme can be seen as a sub-department with its own manager. To have a complete overview, the safety, technical and cargo themes are added to the figure, but these themes are less important for the project, as these are fixed to certain locations or very specific. For example, the equipment of the technical department is located in Utrecht and is only used when needed. The abbreviations of those specialisms are not further explained here but are described in the List of Abbreviations.

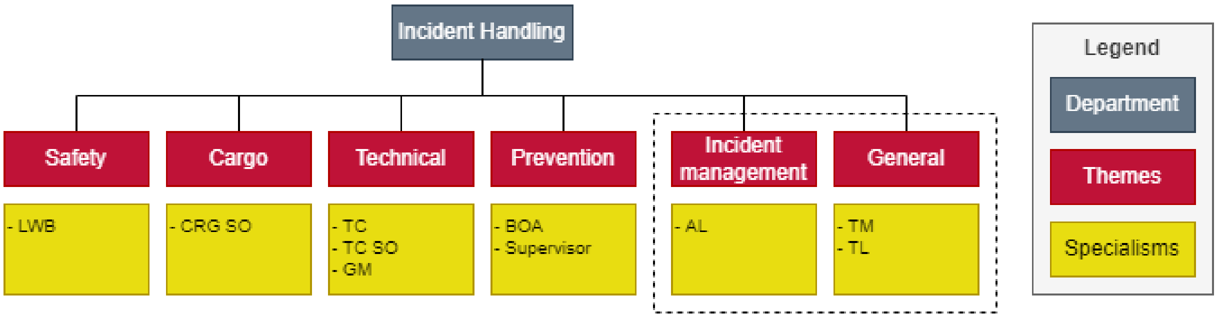


Figure 6: Schematic overview of ICB themes and specialisms

The themes Prevention and Incident Management are of great importance. The incident handlers of these themes are constantly preventively outside (e.g. checking railway sections and crossings) and thus can be allocated to specific areas. Next to the themes, all incident handlers are general ICB members. Therefore, in the figure, a General table can be seen. This, because there exist incidents that do not require any expertise but only an ICB member to handle an incident.

Below, first, the role of general ICB members is explained, followed by the underlying specialisms of the two themes.

General

Incident handlers are the operational members who respond to incidents with priority vehicles. They are responsible for:

- Recovering of derailed trains and assistance with incident handling (clearing the railway track)
- Assisting with emergency services
- Assisting in the evacuation of travellers from stranded trains

From the role of supervision and enforcement, they also have a preventive role (identifying and reducing copper theft, unauthorized rail trespassers, awareness sessions at schools, actions at railway crossings). The incident handlers (TM) are divided into groups led by a team leader (TL). A team typically has four members for regular incidents and six members for incidents with hazardous substances.

Incident management

Incident management is the theme that covers the general leader (AL) specialism. The AL is responsible for:

- Operational management and coordination of incident handling
- Coordination with emergency services
- Ensuring a safe workplace (railway clearance)
- Issuing expected incident handling duration and managing the handling within the forecasted duration
- Coordination with neighbouring infrastructure managers about incidents on border track sections

Important to note that there are some incidents the AL can handle from off-site, and those incidents do not require any ICB member on site. In those cases, the AL is in direct contact with contractors and manages and coordinates the process off-site.

2.1.2. Optimal deployment of ICB

Incident handling takes on average 10 to 20% of the working time of an incident handler. The rest of the time, incident handlers perform preventive and secondary tasks. Preventive tasks are already mentioned, for example checking railway crossings. Besides the preventive tasks, there are also legal requirements. The ICB department has to inspect cargo units to see if the mentioned material is actually in the unit, called a WLIS (Wagon Load Information System) task, and check if a contractor did the correct work, called a WIBON (abbreviation of a Dutch law) task.

At ProRail, the scheduling of tasks is performed such that a working day of an incident handler is filled with tasks as good as possible. To do this, they use a system called Optimal Deployment ICB. However, as soon as an incident occurs and an incident handler gets called in, the incident handler stops with his current preventive or secondary task and hurries to an incident. The scheduling of tasks is based on the area in which the incident handler works. The model developed in this research allocates incident handlers to an area at different hours of the day. Based on the results of this project, ProRail schedules preventive tasks for incident handlers.

2.1.3. Planning process

Currently, the ICB department works with several shifts. In these shifts, incident handlers can either be actively working or be on standby from home. Shifts that include standby time are overnight shifts or weekend shifts. During the morning shift handover to the afternoon shift, there is an overlap of 1 hour to hand over the work and highlight important notes of that day. The Port of Rotterdam, Kijfhoek (freight shunting yard near Rotterdam) and Schiphol airport are specific ProRail locations that do not have standby shifts because of the importance of these locations. Also, the shifts at those locations do not vary between week and weekend. There is also a different shift for the ALs compared to the general ICB members. In Table 1, we see an overview of the shifts. On the weekend, employees need to work 5.5 hours, but these hours are not fixed. For the remainder of the day, employees scheduled for a weekend shift will be on standby.

Table 1: Overview of the possible shifts at the ICB department in general and for specific locations

Netherlands		
	Weekday	**Weekend day**
ICB	06.00 – 15.00	5.5 hrs working, rest standby
	14.00 – 19.30, standby till 06.00	
AL	07.00 – 15.00, standby 03.00 – 07.00	
	14.00 – 22.00, standby 22.00 – 03.00	

Port of Rotterdam, Kijfhoek, Schiphol airport		
	Weekday	**Weekend day**
ICB & AL	07.00 – 15.00	07.00 – 15.00
	15.00 – 23.00	15.00 – 23.00
	23.00 – 07.00	23.00 – 07.00

If we look at the number of employees scheduled in the shifts, we are only interested in the specialism we consider. On a weekday, every shift contains a minimum of 20 ICB members. During the weekend, this minimum is 17 ICB members per shift. By discussing the scheduled number of employees with ProRail, we found that this minimum number never occurs. In fact, always more employees are scheduled. That is because, by contract, every member needs to work a certain number of hours and needs to have enough resting hours. For the ALs, the minimum number is 6 and holds for both the week and the weekend. Also, again, often more ALs are working than this minimum number.

2.2. Railway sections

The Dutch railway network is similar to a road network. Both networks do have sections, junctions and vehicle-specific sections. At ProRail, sections on the railway network are denoted by geocodes and location markers. In this way, a location can be described precisely. The geocode specifies an area, and the location markers specify per hundred meters the location within an area. Figure 7 displays geocode areas in pink and names them with bold numbers. The non-bold numbers in the figure represent the hectometre markers. The ICB department uses this locating system to address locations to incidents.

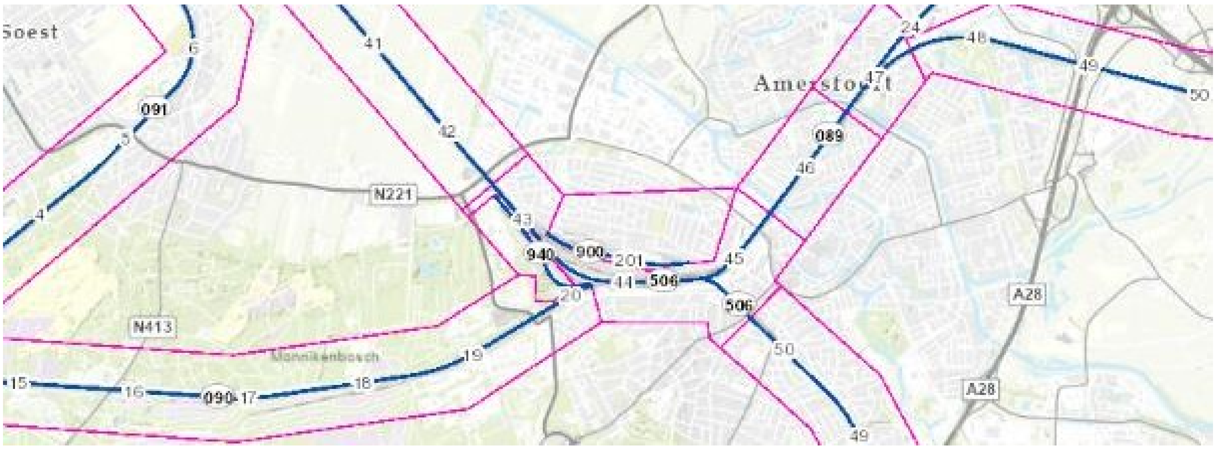


Figure 7: Railway location marking system by ProRail

2.2.1. Schiphol

In Figure 8, we see the railway section at Schiphol, the main Dutch airport. This section is almost entirely underground in a tunnel. Being in a tunnel makes incident handling very different and specific compared to regular railway sections. Besides being in a tunnel, this section also has a high intensity of trains passing by and is an essential link in the Dutch railway network caused by many routes passing this section. Due to this section's specific incident handling and importance, many incidents are called in at this location. A minor delay or smoke alarm immediately gets called in. Therefore, ProRail has an ICB office at Schiphol to adequately respond to incidents in the tunnel of Schiphol. Incident handlers located at this office are allowed to leave this office and go to incidents outside the Schiphol area. However, in reality, they often stay seated and allow other incident handlers in the surroundings to go to incidents that are not near the Schiphol railway section. For this research, this will not add constraints to the model, but we expect to have many employees allocated to Schiphol/Amsterdam.

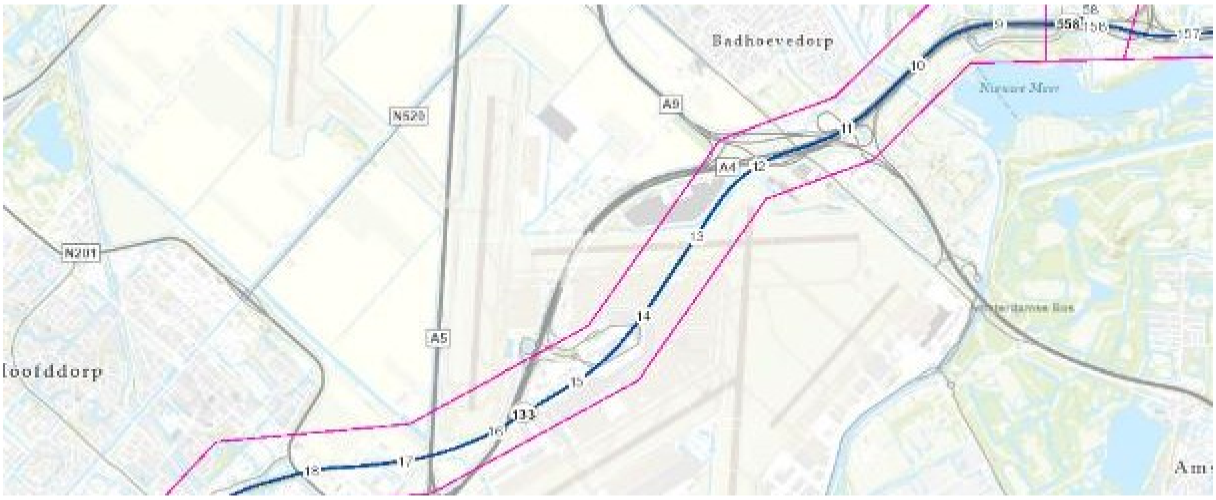


Figure 8: Schiphol railway section

2.2.2. Port of Rotterdam

The port of Rotterdam is the biggest in Europe (Sinha, 2021). One of the ways to transport the freight to the inland is by train. Many container terminals are linked to a railway to do this as efficiently as possible. In Figure 9, we see that the complete port is covered by railways. Contrary to the regular public transport railway network, the network in the Port of Rotterdam is only used to transport freight. Freight brings different kinds of incidents. Especially liquids or gasses require precise incident handling. Besides, incidents with any form of freight can have a significant impact on the environment and the operation of the Port of Rotterdam. Therefore, it requires a specific team of incident handlers, and similar to the Schiphol area, the Port of Rotterdam has its own office and is always manned. Incidents with freight mainly occur at the Port of Rotterdam and rarely on the mainland. For this project, we expect incidents with hazardous substances at the Port of Rotterdam to require an ICB team of 6 employees instead of 4 for regular incidents based on discussions with experts of ProRail.

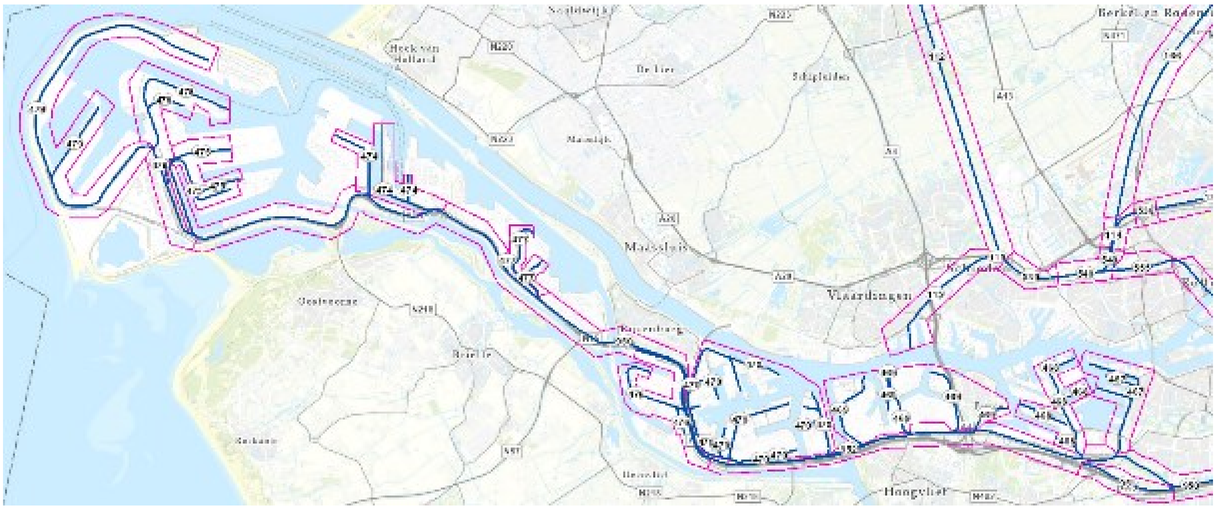


Figure 9: Port of Rotterdam railway section

2.3. Incident classification

Incidents occur at random (unpredictable) moments, and one incident is often completely different from another. To adequately respond to an incident with the correct specialisms and equipment, ProRail uses an incident classification system. As soon as an incident is called in, an incident is given a classification, and the appropriate incident handlers respond to the call.

There are two incident classification systems. The first is the Train Incident Scenario (TIS), and the second is the incident labelling. The latter gives a more detailed description of the incident. We first discuss the TIS classification. In general, there are five main scenarios, each with several sub-scenarios. The main scenarios describe the type of incident, and the sub-scenario describes the complexity of the incident. In Table 2, we see the main scenarios and the frequency of occurrence over the past four years (July 2017 – March 2021). In Appendix A, we describe the main TIS scenarios with all sub-scenarios.

Table 2: Train incident scenarios (TIS) with description and frequency

TIS	Description	Freq
TIS 1	Disruption train operations	64272
TIS 2	Fire	376
TIS 3	Collision or derailment	1842
TIS 4	Hazardous substances	200
TIS 5	Bomb threat	107

The TIS scenarios already give an idea of the type of incident but still provide minimal information. Therefore, ProRail also classifies incidents using incident labels. Table 3 provides an overview of the top 10 incident labels based on the frequency of occurrence over the past four years (July 2017 – March 2021). In fact, there exist more than 80 incident labels, all of which can be found in Appendix A (including frequencies).

Table 3: Top 10 of incident labels based on the frequency

#	Incident Label	Freq
1	Defective material	18163
2	Disturbance due to persons on or near the track	11747
3	Section failure	5348
4	Switch failure / defect	4818
5	Crossing failure / defect	4233
6	Disturbance due to order / assistance of emergency services	2486
7	Disturbance due to object / vehicle / animal(s) on or near the track	2267
8	Track condition	1980
9	Other	1974
10	Slippery tracks	1668

The frequencies in Table 2 and Table 3 give an idea of the types of incidents and how often incidents occur in a timeframe. However, for the project, not all data is used. The TIS scenarios all have four sub-scenarios (e.g., TIS 2.1 till TIS 2.4) except the TIS 1, which also has TIS 1.0. Incidents that are handled by contractors, thus not by the ICB department, use this scenario. Therefore, for the project, TIS 1.0 is excluded from the data. TIS 1.1 is also excluded from the data as ProRail aims to solve TIS 1.1 incidents from the back office and thus does not send any incident handler.

Based on discussions with experts at ProRail, we also exclude some labels. Either the frequency is too low, the incident label does not require the deployment of any incident handler (although the sub-TIS scenarios of those incident labels indicate more complex incidents than TIS 1.0 and 1.1), or the incident is too specific for this project. We highlighted the deleted incident labels in grey in Appendix A.

2.4. Data

We use multiple datasets for the project, all limited to the period between the 24th of June 2017 and the 26th of March 2021. The datasets are linked to each other by the incident ID. The datasets include, amongst others, incident start and end time in date format, TIS scenario, incident label and location information. Figure 10 displays the datasets, including their attributes. Unfortunately, the deployment dataset does not cover the same time period because ProRail started registering deployment as of 2020. Therefore, we extrapolated the deployment data per incident label to the incident dataset.

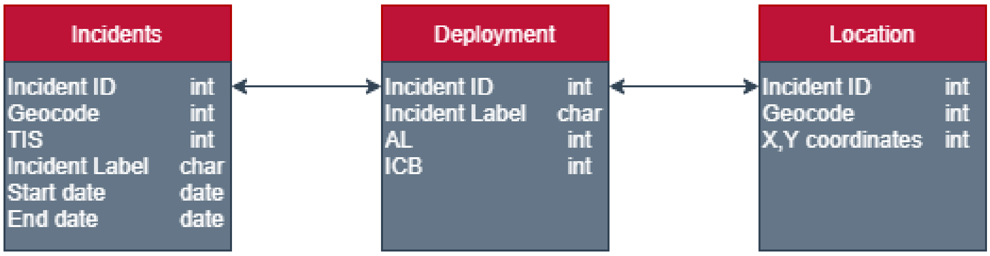


Figure 10: Graphical representation of the datasets

With this data, we can obtain interesting insights. We will analyse the frequency of incidents over the years, seasons, months, days and hours. All data used is filtered based on the criteria mentioned above. Figure 11 shows the frequency of incidents of all years in the dataset. As the data of 2017 and 2021 are not complete, those years are less interesting. Besides, 2020 was the year COVID-19 influenced the Dutch public transport significantly. We also see this influence in the frequency of incidents in that year.

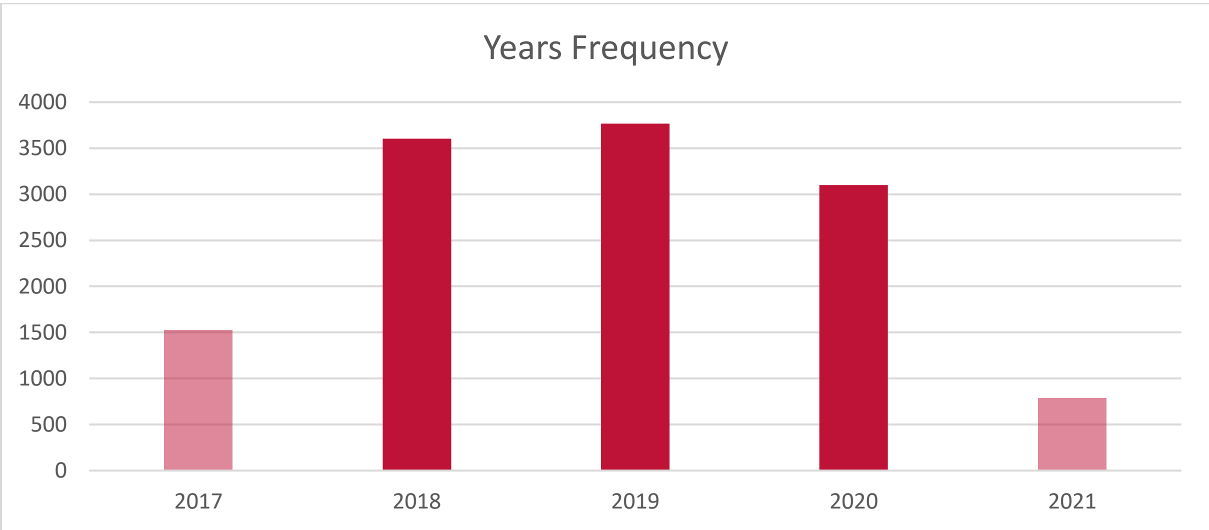


Figure 11: Frequency of incidents over the years

From here on, we only focus on whole years, thus excluding 2017 and 2021. Going from yearly view to seasonal view, in Figure 12, we see that most incidents happen during the summer and less in spring. Autumn and winter are similar to each other. That means that the capacity can be equal during autumn and winter, less during spring and more during summer. Again, in the year 2020, COVID happened. It is interesting to see that the lockdown in the Netherlands during spring resulted in fewer incidents. However, for the remainder of the year, the frequency of incidents is similar to other years.

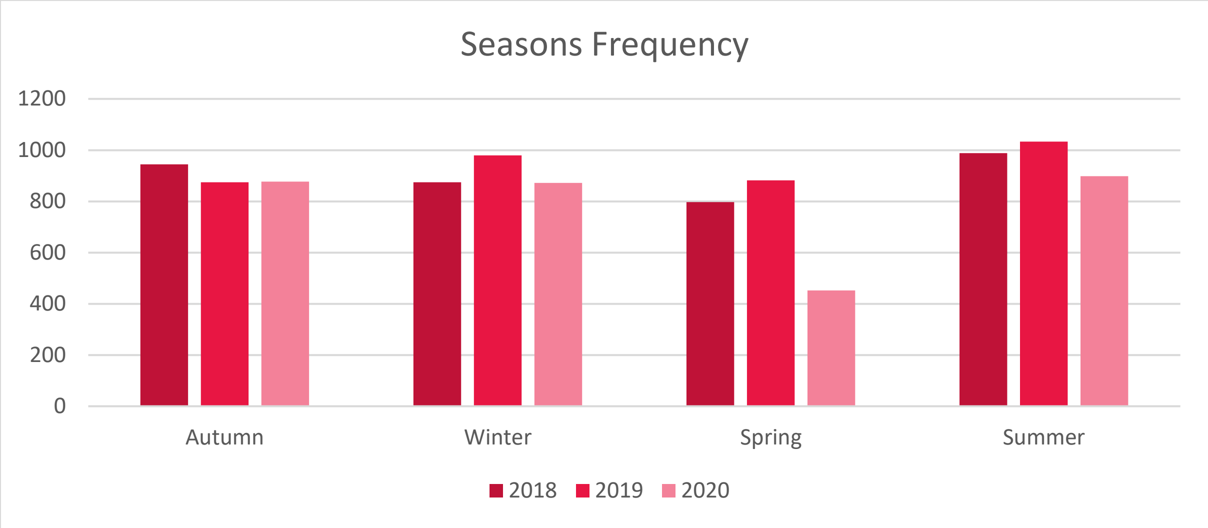


Figure 12: Frequency of incidents over the seasons

Another interesting and helpful insight is the frequency of incidents across the different days of the week. ProRail currently works with different shifts for the weekend. According to Figure 13, this is appropriate as fewer incidents occur during the weekend. When looking at the different working days, we do not see many fluctuations during the week. Incidents frequency on Saturday and Sunday is also similar to each other.

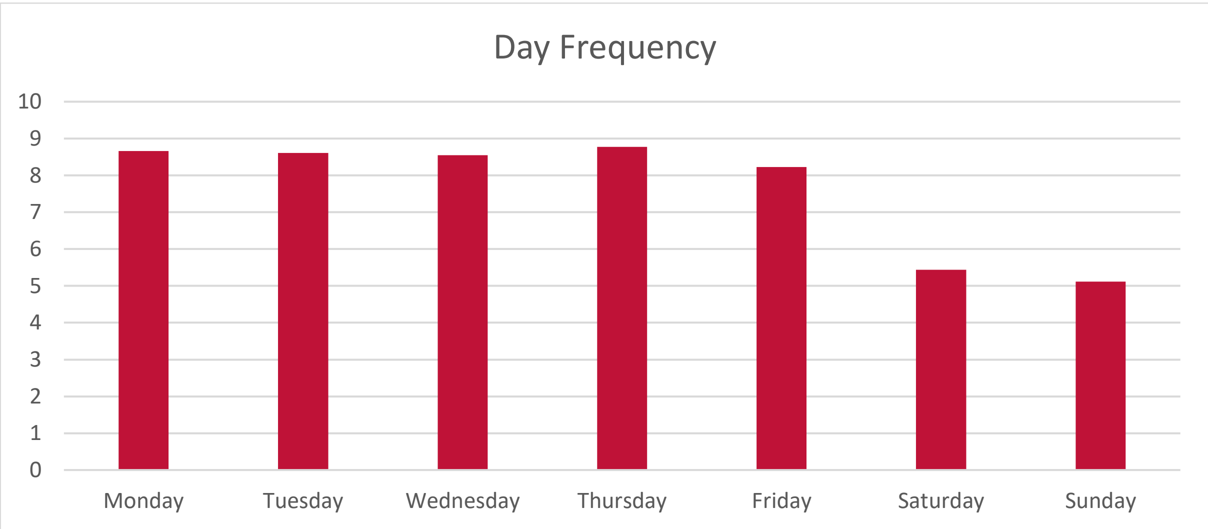


Figure 13: Frequency of incidents over the days

Lastly, we look at the different hours of the day. Figure 14 shows the frequency of incidents during a working day as well as a weekend day. From this data, we can conclude that most incidents occur during the afternoon rush hour. The pattern between a working and weekend day is similar, except for the difference in the morning. During the week, incidents occur earlier on the day compared to the weekend. Also, during the evening and night, the frequency of incidents is equal or higher compared to a working day.

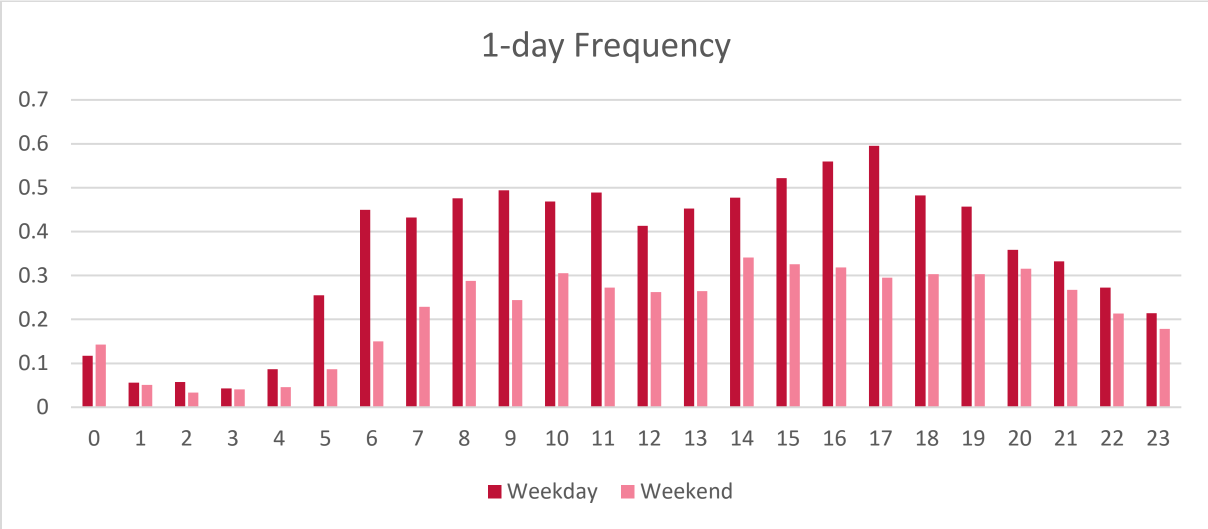


Figure 14: Frequency of incidents over the hours

Besides the distribution of incidents, we can also make density plots of the incident locations. These density plots help to understand the outcome of this research. Figure 15 displays two density plots, the first includes all data (except TIS 1.0 and 1.1 because out of scope), and the second includes only TIS 3.1 (collision with a person, bicycle or other small objects). The left plot shows that the most incident-dense area is in the Randstad. Outside the Randstad, incidents mainly occur in big cities. When looking at TIS 3.1 only, this type of incident occurs almost everywhere and is spread more evenly across the country. The location density plots and the frequency tables provide us insight and an idea of the outcome of this research. The density plots suggest that the area of the Randstad requires more incident handlers.

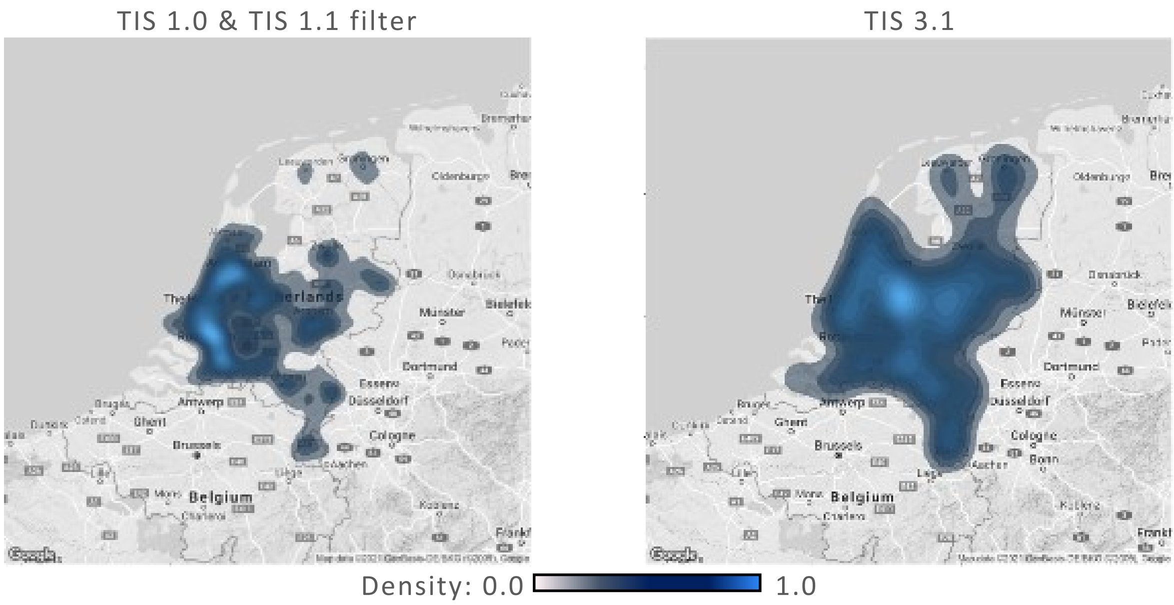


Figure 15: Incident location density plots

Considering the density plots in Figure 15 and incidents in general, we only know areas where incidents occur more frequently. However, this does not mean that incidents do not occur in other areas. As incidents themselves are unpredictable, also the locations and impact are unpredictable. Therefore, it is important to have a robust schedule of incident handlers. That means that the scheduled incident handlers can cover many scenarios of incidents. The data only shows historical incidents and locations.

It is important to evaluate not only the data but also a stochastic environment where incidents can occur more or less frequently.

2.5. Conclusion

We analyse the current situation of the ICB department. For this research, we are primarily interested in the starting moment of incidents. Also, incident handlers are primarily busy with preventive and secondary tasks, and as soon as an incident happens, they drop that task and hurry to that incident. ProRail already uses a system that assigns preventive and secondary tasks to incident handlers (this system is called Optimal Deployment ICB). ProRail labels the incident handlers across five different themes and underlying specialisms. The shifts are the same for the whole country, except for the Port of Rotterdam, Kijfhoek and Schiphol. The current capacity is indicated by a minimum number. In reality, capacity on a day is more than this minimum number.

The data available for the project is limited to the period between the 24th of June 2017 and the 26th of March 2021. We can quickly analyse the peak moments and areas from this data, giving us an idea of the outcome. As input to the final model, we filter the data to incidents requiring incident handlers' deployment.

3. Literature review

In this chapter, we perform a literature review on incident handling and disturbance management. First, in Section 3.1, we discuss different planning levels. In Section 3.2, we analyse different covering models and discuss the pros and cons of the models. In Section 3.3, we analyse algorithms proposed in the literature to solve covering models. Finally, we conclude the literature review in Section 3.5.

3.1. Planning levels

Planning decisions can be classified at three levels: operational, tactical, and strategical (Swamidass, 2000). The operational level considers online and offline planning. Online planning occurs during the day (adapting to changes throughout the day), and offline planning is the planning for the next day or week. Tactical planning is the planning for the next month(s), and strategical planning is the long term planning, ranging from multiple months to multiple years.

Currently, ProRail already works with an optimal deployment field service system. This system optimally divides non-emergency tasks over a set of incident handlers available at a time moment. The system is based on the median routing problem adapted to railway incident operations by Huizing, Schäfer, van der Mei, and Bhulai (2020). With this system, incident handlers perform non-emergency jobs in a specific area such that they always remain close to incident hotspots. The system works with a given number of incident handlers in an area. In contrary to that system, this research focuses on the latter. It aims to determine an optimal number of incident handlers and, therefore, plans at another planning level. The optimal deployment field service system can be classified at the operational level; this research can be classified at the strategic level as this is general advice on the planning of incident handlers on a general day to use for multiple years.

3.2. Emergency service

Incidents come in various ways and have various causes. Therefore, a well-organized and well-coordinated emergency response is required to cope with incidents. From a public point of view, this means adequate and quick response at all times. However, this is hard to realize as resources are costly and limited to cover only a certain number of incidents adequately and quickly.

This research focuses on reaching a certain coverage ratio by efficiently positioning incident handlers based on historical incident data. In the past few decades, a great deal of research has been done on positioning emergency services, predominantly focussed on, and referred to, efficient positioning of emergency medical services (EMS) (Li, Zhao, Zhu, & Wyatt, 2011). The problem of positioning emergency services to cover incidents is in its simplest form related to the Facility Location Problem (FLP). In the FLP, the aim is to minimize the sum of distances from each demand point to the nearest facility. Our problem is related to the FLP, but with some modifications. Instead of facilities, we determine the number of incident handlers required and position them efficiently to reach a threshold coverage. Contrary to the FLP, not all demand points need to be covered in our case because incidents with less impact are allowed to have a longer solving time. Furthermore, incident handlers each have their specialism and thus cannot cover all incident types.

3.2.1. Covering models

Looking at models which consider maximizing demand covered and minimizing the number of required resources, among others, one of the first models proposed are the Location Set Covering Problem (LSCP) by Toregas et al. (1971) and the Maximum Coverage Location Problem (MCLP) by Church and ReVelle (1974). The LSCP considers a set of locations where facilities might be opened and a set of demand nodes. A matrix is created that includes the distance for each possible facility location to a demand node based on these two sets. If a facility is opened and the distance to a demand node is less or equal to a threshold distance, that demand node is covered. In the LSCP, the objective is to minimize the facilities used to cover all demand nodes. Similar to the LSCP, the MCLP also has a set of locations where facilities might be located, a set of demand nodes and a distance matrix. However, in the MCLP, the objective is to maximize coverage given a fixed number of facilities.

Both the LSCP and the MCLP model are already fifty years old and have been applied in many planning problems, for example, the positioning of fire stations, ambulances and hospitals (Bianchi & Church, 1988). Nowadays, these models act as the foundation of many variants. The MCLP is still one of the well-known location problems (Bansal & Kianfar, 2017).

The LSCP simplifies reality by assuming the system to be static and deterministic. As the objective is to minimize the number of resources used, considering the constraint that all demand nodes have to be covered, the resources are considered infinite. Besides, it assumes that a resource can serve all demand nodes within its reach (it could not be that the resource is busy). Despite these rough assumptions, it is still a useful model on a strategic level to determine the minimum resources required to provide complete coverage (Li et al., 2011). An interesting derivative of the LSCP is the Probabilistic Location Set Covering Model (PLSCM), especially the a-reliable p-center problem. The a-reliable p-center problem can be stated as follows: find the position of p facilities that minimize the maximum time (or distance) within which service is available with a reliability (Revelle & Hogan, 1989). This problem determines the minimum number of facilities under the constraint that it needs to be more or equally reliable than a, which is already close to our problem.

The MCLP efficiently positions facilities to maximize coverage but does not consider the capacity of each facility. Schilling, Elzinga, Cohon, Church, and ReVelle (1979) developed the Tandem Equipment Allocation Model (TEAM) and the Facility Location and Equipment Emplacement Technique (FLEET) model to position facilities and allocate equipment simultaneously. With this, it is possible to locate more emergency handlers at one facility to cover areas with many incidents. Also, the MCLP assumes that the equipment is always available. Especially in EMS, it is possible that an emergency responder located within the service distance already serves another demand, and additional responders are needed to guarantee coverage (Tavakoli & Lightner, 2004).

Bianchi and Church (1988) developed the Multiple Cover, One-unit FLEET (MOFLEET) model, which maximizes the expected coverage by simultaneously locating and allocating a fixed number of facilities and emergency responders, respectively. The MOFLEET model considers every emergency responder to be of the same type. However, there exist situations where each emergency requires different types of responders (e.g., in a medical emergency, either send an ambulance or a helicopter or both). Therefore, Jayaraman and Srivastava (1995) introduced the Multiple Equipment Multiple Cover Facility Location-Allocation (MEMCOLA) model. The MEMCOLA model maximizes the expected coverage within the service distance by efficiently locating a fixed number of facilities and different types of equipment. By considering multiple types of equipment, the MEMCOLA can also define the service distance and busy fraction for each type independently. Busy fraction meaning the probability an emergency handler being busy at the moment a new incident occurs.

The Maximal Expected Coverage Location Problem (MEXCLP) developed by Daskin (1983) extends the MCLP by considering that when demand arrives as the system, it cannot directly be served as facilities are already engaged serving other demand. An emergency responder k is busy with a fraction q. The expected coverage of a demand node by an emergency responder is then $E_k = 1 - q^k$. It is assumed that this probability of a random emergency responder being busy is independent of any other emergency responder being busy and can be seen as the probability of at least one success in k

independent Bernoulli experiments. The increase in expected coverage at a node is then given by $E_k - E_{k-1} = \left(1 - q^k\right) - \left(1 - q^{k-1}\right) = (1 - q)q^{k-1}$.

Many other probabilistic model variations have been developed (Li et al., 2011). Those models focus on the calculation of an emergency handler being busy. Revelle and Hogan (1989) developed two attractive versions of the Maximal Availability Location Problem (MALP I and MALP II), formulated as a constrained stochastic program (Charnes & Cooper, 1959). In both models, the objective is to maximize demand covered with a given probability α. The number of emergency responders to locate is given. The difference between MALP I and MALP II relies upon the busy fraction calculation. MALP I assumes all emergency responders are equally busy, and MALP II associates the busy fraction with a demand node and the availability of emergency handlers in its service area.

Besides the MALP models, Batta, Dolan, and Krishnamurthy (1989) proposed an adjusted version of the MEXCLP model (AMEXCLP) in which the objective is multiplied by a correction factor. In this way, emergency responders can be viewed as dependent servers in a queuing system. In the AMEXCLP model, the busy fraction is the same for the entire system. Marianov and Revelle (1994) extended this view and proposed a model in which the busy fractions are location specific, called the Queuing Probabilistic Location Set Covering Model (QPLSCP).

Repede and Bernardo (1994) stated that the assumption of independence of a random emergency responder being busy together with the fact that demand is assumed to be constant over time results in an overestimation of coverage. They developed the Time-dependent Maximum Expected Coverage Location Problem (TIMEXCLP). The TIMEXCLP model includes time periods and maximizes the expected coverage over the time horizon (e.g., time periods of an hour and time horizon of a day). By adding time periods to the model, the busy fraction and the demand can vary over time. In the model, there is no relationship between different time periods. Therefore, each time period can be seen as independently. This idea has the drawback that there can be significant differences in the positioning of emergency responders between time periods, potentially resulting in high costs for the emergency service provider (van den Berg & Aardal, 2015).

Another model that considers the possibility of an emergency responder being busy is the Double Standard Model (DSM) by Gendreau, Laporte, and Semet (1997). A demand node is considered covered in this model if at least two emergency responders can reach it. However, contrary to the MEXCLP, the possibility of both responders being busy is not considered.

An overview of all discussed models, including their objective and constraints on coverage, locations sites and types and number of emergency responders, as well as the formulation of the busy fraction, can be seen in Table 4.

3.2.2. Coverage functions

Until now, all the discussed models consider a demand node fully covered if that node is within service distance of a positioned emergency responder; everything outside the service distance is not covered at all. However, in our case, the further away an incident occurs from an incident handler, the lower the possibility that that incident handler covers the incident. Therefore, coverage decreases over distance. Several researchers suggested gradual coverage functions to replace the binary view of coverage (see Figure 16). Figure 16a displays the binary view of coverage; if within service distance, then a demand node is fully covered. Figure 16b shows a stepwise coverage function (Berman & Krass, 2002; Church & Roberts, 1983). When outside the 100%-coverage service distance, a demand node can still be covered, only at a lower percentage (e.g. longer travel time, so not within the desired duration at a location, but still covering at a secondary desired duration). The coverage function in

Figure 16c is a graphical representation of the "quality of service"-function (Araz, Selim, & Ozkarahan, 2007; Pirkul & Schilling, 1991) and Figure 16d similar to the function proposed by Berman, Krass, and Drezner (2003). These gradual coverage functions can be added to the model either in the objective, as constraints or as (pre-processed) parameters.

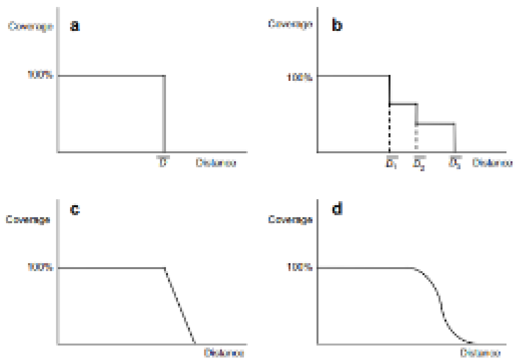


Figure 16: Gradual coverage functions (Eiselt & Marianov, 2009)

3.3. Optimization techniques

Location covering problems are combinatorial optimization problems that belong to a complex class of optimization problems. They are typically classified as NP-hard problems requiring exponential time to be solved optimally (Rajagopalan et al., 2008). Because of the complexity of such problems, various metaheuristic search methods have been developed to find near-optimal solutions in reasonable computational time (Osman & Laporte, 1996) such as; Tabu Search (Adenso-Díaz & Rodríguez, 1997; Doerner, Gutjahr, Hartl, Karall, & Reimann, 2005; Gendreau et al., 1997; Rajagopalan et al., 2008; Rajagopalan, Vergara, Saydam, & Xiao, 2007), Simulated Annealing (Chiyoshi & Galvão, 2000; Galvão et al., 2005; Rajagopalan et al., 2007), Genetic Algorithm (Aickelin, 2002; Aytug & Saydam, 2002; Beasley & Chu, 1996; Jaramillo, Bhadury, & Batta, 2002; Jia, Ordóñez, & Dessouky, 2007), Lagrangian Relaxation (Galvão & ReVelle, 1997; Jia et al., 2007; Karasakal & Karasakal, 2004), Ant Colony Optimization (Doerner et al., 2005) and Local Search (Aytug & Saydam, 2002; Iannoni, Morabito, & Saydam, 2007). More recently, researchers gained interest in integrating machine learning techniques into metaheuristics for solving combinatorial optimization problems, such as covering problems with the aim to improve their performance in terms of solution quality, convergence rate and robustness (Karimi-Mamaghan, Mohammadi, Meyer, Karimi-Mamaghan, & Talbi, 2021).

Of all mentioned metaheuristic search methods, simulated annealing (SA) is a method that is usually easily implemented. Also, it generally requires less computational effort than more sophisticated procedures such as tabu search and genetic algorithms (Galvão et al., 2005). Moreover, SA has been successfully used for covering models (Brotcorne, Laporte, & Semet, 2003; Chiyoshi & Galvão, 2000; Galvão et al., 2005; Rajagopalan et al., 2007). SA is a probabilistic search method that approximates the global optimum in a large solution space for an optimization model. SA can escape from local optima by accepting worse solutions with some probability. The general SA method requires an initial solution, which can be generated entirely at random. Besides the initial solution, SA also needs a starting temperature, cooling factor and Markov chain length. In the SA algorithm, neighbour solutions are generated by slightly adjusting the current solution. The neighbour solution is accepted if better than the current solution. If not better, it can still be accepted against a probability. By accepting worse solutions, neighbour regions in the solution space are explored, and the algorithm can escape from local optima. The probability of accepting worse solutions is based on the Boltzmann distribution. The probability of acceptance follows a cooling scheme and decreases iteratively. When the temperature is high, the probability of accepting worse solutions is also high. When the temperature gradually decreases with the cooling factor, the probability of accepting worse solutions also decreases. Accepting many worse solutions is called diversification, and gradually lowering the probability of accepting worse solutions is called intensification.

3.4. Robustness

Traditionally formulated, optimization problems are static and deterministic. However, the reality is dynamic and uncertain meaning that input parameters fluctuate due to environmental changes, human behaviour, material wear, etc. (Chica & Juan 2017). Excluding uncertainty from optimization problems leads to potentially unstable solutions sensitive to small changes in the input parameters. Simheuristics (Juan, Faulin, Grasman, Rabe, & Figueira, 2015) provide a framework that allows us to address real-world problems with uncertain parameters by using both metaheuristics and simulation. This allows obtaining computationally efficient solutions while considering their impact on dynamic and uncertain (stochastic) scenarios. The simulation process considered in the framework allows to model and reproduce complex stochastic problems. By analysing the information provided by the simulation, it is possible to estimate the feasibility of the best solutions in stochastic scenarios. Based on that, we can select those solutions that maximize robustness. That is, the selected solutions are not minimizing the objective function of the deterministic problem, but the one that also meets certain criteria considering the uncertainty, e.g., maximizes the robustness of the coverage of allocated incident handlers (de León, Lalla-Ruiz, Melián-Batista, & Moreno-Vega, 2021). The work of De Armas, Juan, Marquès, and Pedroso (2017) shows that simheurstics can easily be used to provide robust solutions for the FLP problem.

3.5. Conclusion

In the past few decades, a great deal of research has been done on positioning emergency services, predominantly focussed on and referred to efficient positioning of emergency medical services. Therefore, it is not unexpected that we find many optimization models proposed for solving the capacity allocation problem to guarantee coverage in the literature. These models vary from theoretical and simplistic models, such as the facility location problem, to sophisticated models where emergency handling can be seen as queuing systems. From all reviewed models, we are primarily interested in the TIMEXCLP model because of the possibility of solving multiple time instances within the same model and obtaining an overall coverage level.

Covering models typically use a binary coverage view, meaning a node is either covered or not. There exist functions that change this binary view to a gradual coverage view. In these functions, coverage of an allocated incident handler gradually decreases when increasing the distance from that incident handler. This creates a more realistic view of coverage.

Location covering problems are typically optimization problems that belong to a complex class of combinatorial optimization problems. Because of the complexity of such problems, various metaheuristic search methods have been developed to find near-optimal solutions in reasonable computational time. Of these metaheuristic search methods, simulated annealing is an appropriate method. Simulated annealing is a method that has been successfully implemented to solve covering models. It is a method that is usually easily implemented. Also, it generally requires less computational effort than more sophisticated procedures such as tabu searches and genetic algorithms. By combining SA with simulation, we create a simheuristic that allows us to evaluate solutions in stochastic scenarios

Chapter 4 discusses the model and the simulated annealing algorithm in detail, theoretically and mathematically. This includes discussing the operators and parameters of the metaheuristic.

Table 4: Summary of models obtained from literature

Reference	Model	Objective	Coverage	Location sites	Types and amount	Busy fraction
Toregas et al. (1971)	LSCP	Minimize number of units located	Cover each demand node	At most one unit per site	One type, unlimited amount	N/A
Church and ReVelle (1974)	MCLP	Maximize demand covered	N/A	At most one unit per site	One type, amount given	N/A
Schilling et al. (1979)	TEAM	Maximize demand covered	N/A	At most one unit of each type per site, only type A if also type B at site.	Two types, amount given	N/A
Schilling et al. (1979)	FLEET	Maximize demand covered	N/A	At most one unit per site. number of sites given	Two types, amount given	N/A
Bianchi and Church (1988)	MOFLEET	Maximize expected demand covered	N/A	Multiple units per site, number of sites given	One type, amount given	Given fraction, identical for each unit
Jayaraman and Srivastava (1995)	MEMCOLA	Maximize expected demand covered	N/A	Multiple units, multiple types per site, number of sites given	Two types, amount given	Given fraction, identical for each unit, different per type
Gendreau et al. (1997)	DSM	Maximize demand covered at least twice	Proportion α of demand node covered within r_1, all demand nodes covered within r_2	Maximum number of units per site fixed	One type, amount given	N/A
Daskin (1983)	MEXCLP	Maximize expected demand covered	N/A	N/A	One type, maximum amount given	Given fraction, identical for each unit
ReVelle (1989)	MALP I	Maximize total demand covered with reliability α	N/A	N/A	One type, amount given	Given fraction, same for all sites
ReVelle (1989)	MALP II	Maximize total demand covered with reliability at least α	N/A	N/A	One type, amount given	Different according to demand node
Batta et al. (1989)	AMEXCLP	Maximize expected demand covered	N/A	N/A	One type, amount given	Different according to demand node, units not independent
Repede and Bernardo (1994)	TIMEXCLP	Maximize expected demand covered	N/A	N/A	One type, amount given for time period	Given fraction, identical for each unit, can vary of time
Revelle and Hogan (1989)	α-reliable, p-center problem	Minimize maximum service time with α reliability	At least proportion α of demand covered	N/A	One type, amount unlimited, minimum given	N/A
Marianov and Revelle (1994)	QPLSCP	Maximize total demand covered with reliability at least α	N/A	N/A	One type, minimum amount given	Different according to demand nodes

4. Solution design

In Section 4.1 of this chapter, we provide both a theoretical and mathematical model description. The theoretical description includes the model assumptions, a description of the specialisms and the objective function. We follow the theoretical model description by the mathematical model formulation with a detailed explanation of the constraints in Section 4.2. Lastly, Section 4.3 describes the initialisation, neighbourhood operators, and stopping criterium of the metaheuristic solution approach.

4.1. Model description

This research aims to allocate a minimum number of incident handlers across the country and across every hour of the day such that a threshold coverage is reached. As a result of the literature review, covering models are the appropriate models to solve this covering problem. There exist many variations, all with their specific advantages and disadvantages. For our problem, there are two models which are very relevant.

First of all, the LSCP model by Toregas et al. (1971). We recall that in the LSCP, the objective is to minimize facilities used to cover all demand nodes. In general, this is very similar to our objective. However, we are interested in a threshold level of demand nodes covered. Considering multiple variations of the Maximum Covering Location Problem, the second model, the TIMEXCLP of Repede and Bernardo (1994), is the most relevant. The TIMEXCLP has the advantage to solve the allocation problem to a threshold coverage and also to do this for multiple time instances at once. The drawback of the TIMEXCLP is that the objective is to maximize the expected demand covered for a given number of facilities to be located. We combine the LSCP and TIMEXCLP model to create a model that minimizes the number of incident handlers over different time instances by allocating them such that the constraint of a threshold coverage is achieved. In this way, the number of incident handlers is unrestricted and optimized by the model. In reality, the coverage decreases with the distance away from an incident handler increasing, and we incorporate this in our model with a gradual coverage function. We found multiple gradual coverage functions in the literature, as described in Section 3.2.2. As our model is on strategic level, we are not looking for any advanced gradual coverage function. Based on discussions with ProRail about different coverage functions, we found the stepwise gradual coverage function most appropriate because, with this function, we can easily describe coverage boundaries and levels. Therefore, we implement the stepwise gradual coverage function to our problem.

4.1.1. Assumptions

After reviewing the literature and discussions with experts of ProRail, we made several assumptions to simplify the modelling and reduce the complexity. These assumptions are:

- We do not take into account the incident duration; we only consider the starting moment of incidents;
- We do not take into account the possibility of incident handlers being busy;
- The reach of incident handlers to cover incidents is equal across the country, thus does not depend on the actual road infrastructure;
- Travelling of incident handlers is not relevant, meaning at one hour they could be at one side of the country and another hour on the other side;
- The stepwise coverage function is 100% within 30km radius, 50% within 30-45km radius and 25% within 45-60km radius and

4.1.2. Grid design

Besides these assumptions, we use a grid design. Incidents can happen anywhere across the railway network. This means that incidents can happen on every coordinate (as long as it represents a railway track). If we use the same structure in the model and allocate an incident handler to any coordinate on the map, the problem becomes gigantic and requires unrealistic computational time. Therefore, we transform the coordinate map to a grid design, as schematically shown in Figure 17. In this figure, we see the Dutch rail network and, in the left map in the figure, we indicate random incidents with red dots. All incidents that occur within the boundaries of a grid cell count as incidents on that grid cell, transforming the coordinate map into a grid design map on the right side. The darker the cell, the more incidents are counted in that cell. Subsequently, we can allocate incident handlers to a grid cell instead of a specific coordinate. Based on discussion with experts of ProRail, we choose to have a cell size of 10 by 10 kilometres and a total of 990 cells (30 by 33).

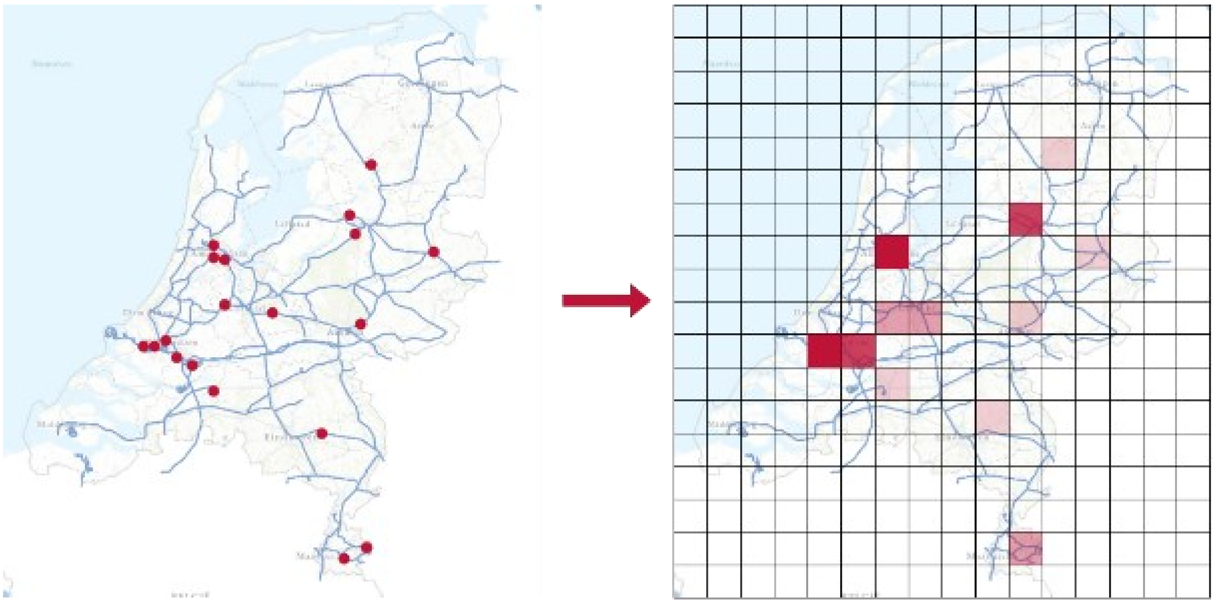


Figure 17: Grid design used in the model

As shown in Figure 17, the North Sea and parts of Germany and Belgium are also covered by grid cells. However, those cells do not have any demand, and it also makes no sense to allocate employees to those cells. Therefore, we exclude those cells from the model. By doing so, we significantly reduce the number of cells from 990 to around 400 and thus improve computational effort even more.

4.1.3. Scenarios

Looking at the research question regarding experimentation, we are interested in different scenarios representing different time frames (years, seasons) and different specialisms of incident handlers. In consultation with ProRail, we created a total of 40 scenarios, as can be seen in Table 5. We look at the years 2018 and 2019 combined, 2020 separately (to see COVID lockdown effects) and all seasons within those two options. As the incidents are significantly different and less during the weekend than the week, we separate the week and weekend. Lastly, we look at two different types of incident handlers, namely the AL and ICB team.

Table 5: Overview of experimental scenarios covering different time frames

	AL		ICB	
	Week	**Weekend**	**Week**	**Weekend**
2018, 2019	1	2	3	4
Winter '18, '19	5	6	7	8
Spring '18, '19	9	10	11	12
Summer '18, '19	13	14	15	16
Autumn '18, '19	17	18	19	20
2020	21	22	23	24
Winter '20	25	26	27	28
Spring '20	29	30	31	32
Summer '20	33	34	35	36
Autumn '20	37	38	39	40

4.1.4. Pre-processing

As mentioned in the model description in Section 4.1, we use the stepwise coverage function to incorporate the fact that the further away from an allocated incident handler, the less the coverage will become. The decrease in coverage is stepwise; the first 30 kilometres away from the incident handler are fully covered (100%). From 30 to 45 kilometres, this coverage lowers to 50%, and from 45 to 60 kilometres, only 25% is covered. Outside this area, an incident is not covered. Using this function, the model aims to cover the areas with high incident density the most, thus allocating incident handlers to high incident density areas.

To incorporate this in the model, we look at two options. The first option is to calculate the reach whenever an incident handler is allocated by adding constraints to the model. However, this requires a computational effort that can be avoided. In this second option, we determine the reach to any other cell in advance for every cell on the grid and implement this to the covering model as a reach parameter. This reach array represents the reach from node i to node j based on the stepwise function. The benefit of pre-processing this array is that it only needs to be calculated once and can then be used as a parameter in the model. Therefore, we use the second option in our model to reduce computational effort. In Figure 18, we see the pseudocode to create this reach parameter.

Pre-process reach procedure

$\boldsymbol{for}\ (i \in grid)\ \boldsymbol{do}$
    $\boldsymbol{for}\ (j \in grid)\ \boldsymbol{do}$
        $distance[i,j] = \sqrt{(i_x - j_x)^2 + \left(i_y - j_y\right)^2} * CellSize$
        $\boldsymbol{if}\ distance[i,j] \leq 30\ \boldsymbol{then}$
            $reach[i,j] = 1.0$
        $\boldsymbol{elif}\ distance[i,j] \leq 45\ \boldsymbol{then}$
            $reach[i,j] = 0.5$
        $\boldsymbol{elif}\ distance[i,j] \leq 60\ \boldsymbol{then}$
            $reach[i,j] = 0.25$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ for}$
$\boldsymbol{end\ for}$

Figure 18: Pre-process procedure to obtain reach parameter

We calculate the distance using the row and column index of the grid cells and multiple that by the cell size to obtain the actual distance. Figure 19 shows the distance calculation for a small example instance. This distance is the Euclidean distance from one cell to another.

1,1	2,1	3,1	4,1
1,2	2,2	3,2	4,2
1,3	2,3	3,3	4,3
1,4	2,4	3,4	4,4

Grid distance from (2,2) to (3,4):

$$\sqrt{(2-4)^2 + (2-3)^2} = \sqrt{5}$$

Actual distance from (2,2) to (3,4):

$$\sqrt{5} * 10 \approx 22.4\ km$$

Figure 19: Example calculation of grid cell distance

4.2. Mathematical model

For the model, we want to allocate a, to be determined, number of employees to ensure a coverage level. Incident handlers can be allocated to any cell on the grid and have a certain coverage reach from their position. First, we introduce and describe the indices and their sets, followed by the parameters and decision variables. Next, we formulate the objective function as well as the constraints, and finally, we formulate the variable restrictions. Afterwards, we explain the objective function, constraint and variables restrictions in detail and provide a graphical explanation.

Indices

i	node $(i = 1, \dots, N)$
j	node $(j = 1, \dots, M)$
t	time $(t = 1, \dots, 24)$

Parameters

$threshold$	coverage threshold parameter (80%)
$demand_{it}$	demand of number of incident handlers at node i at hour t
$reach_{ij}$	reach from node i to node j (100%, 50%, 25%)
$DemandTotal$	total deployment of incident handlers at all hours and all nodes

Variables

$Coverage_{it}$	coverage at node i at hour t
$Employees_{it}$	number of emergency responders located at node i at hour t

Objective function

$$\min \sum_i \sum_t Employees_{it} \tag{1}$$

Threshold coverage constrains

$$\sum_i \sum_t Coverage_{it} \geq threshold * DemandTotal \tag{2}$$

$$Coverage_{it} \leq demand_{it} \qquad \forall i, t \tag{3}$$

Workforce required for threshold coverage constraint

$$Coverage_{it} \leq \sum_j reach_{ij} Employees_{jt} \qquad \forall i, t \tag{4}$$

Variable restrictions

$$Coverage_{it} = nonnegative \qquad \forall i, t \tag{5}$$

$$Employees_{it} = nonnegative\ integer \qquad \forall i, t \tag{6}$$

In this model, we write the objective function (1) to minimize the number of employees used on every node on the grid over every hour of the day. Looking at every hour, the hours with less demand will use fewer employees and vice versa. As a result, we can easily see when we require the most employees on a day.

Constraint (2) ensures that we reach the threshold coverage (80%) of the total demand. Here, total demand represents the total deployment of all incidents on one day. To reach the threshold coverage level, the total coverage obtained by allocating employees to any node at any hour needs to be equal to or higher than the threshold coverage level.

Without the use of constraint (3), allocating employees at night, when demand is low, would have the same impact as when allocating employees in rush hour. Therefore, constraint (3) limits the coverage the be less than or equal to the demand in a grid cell at a specific hour.

Constraint (4) links the required coverage to required employees. Due to the stepwise coverage function (implemented in the model as the reach parameter), coverage can be a decimal number. However, an employee can only be allocated yes or no. Therefore, this constraint ensures that the required employees to at least obtain the coverage are located within reach. If a node is covered multiple times, the coverage can be higher than the employees allocated at one node.

Finally, constraint (5) and constraint (6) are the variables restrictions. As already mentioned, the coverage variable can be a decimal number, more specific a nonnegative decimal number. The employee variable can only be a nonnegative integer number.

We show the model's working with a graphical example in Figure 20. In this example, the position and number of employees is random and not optimized. We show two locations where we allocate employees (Employees Loc 1 & Employees Loc 2). At both locations, we allocate 5 employees, indicated with bold white font. Next, in the Reach grid, show the coverage obtained by the allocated employees. We see that the reach decreases when distance away from the allocated employees increases. Besides, between the two locations, the reach is higher than the number of employees allocated to one location as these nodes can be reached from both employee locations. Thus, the coverage of nodes covered twice is higher. Next, we see the grid nodes with actual demand in the Demand grid and the actual coverage in the Coverage grid. Every node in the Coverage grid is the minimum of the Reach and Demand grid.

Employees Loc 1							
1	1	3	1	1	0	0	0
1	3	3	3	1	1	0	0
3	3	5	3	3	1	1	0
1	3	3	3	1	1	0	0
1	1	3	1	1	0	0	0
0	1	1	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Employees Loc 2							
0	0	0	0	1	0	0	0
0	0	0	1	1	1	0	0
0	0	1	1	3	1	1	0
0	1	1	3	3	3	1	1
1	1	3	3	5	3	3	1
0	1	1	3	3	3	1	1
0	0	1	1	3	1	1	0
0	0	0	1	1	1	0	0

Reach all employees							
1	1	3	1	2	0	0	0
1	3	3	4	2	2	0	0
3	3	6	4	6	2	2	0
1	4	4	6	4	4	1	1
2	2	6	4	6	3	3	1
0	2	2	4	3	3	1	1
0	0	2	1	3	1	1	0
0	0	0	1	1	1	0	0

Demand							
0	0	0	0	4	0	0	0
0	0	0	0	3	0	3	0
0	0	5	0	2	0	3	0
0	0	0	0	1	0	0	0
0	0	4	0	0	4	0	0
7	0	0	2	0	0	0	0
0	0	5	0	0	0	0	0
0	5	0	0	0	0	0	0

Coverage							
0	0	0	0	2	0	0	0
0	0	0	0	2	0	0	0
0	0	5	0	2	0	2	0
0	0	0	0	1	0	0	0
0	0	4	0	0	3	0	0
0	0	0	2	0	0	0	0
0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0

Sum demand grid:

48

Sum coverage grid:

25

Coverage:

$$\frac{25}{48} \approx 52\ \%$$

Figure 20: Example of working of ILP model

In the example of Figure 20, we allocate 10 employees, meaning that the objective value is 10 employees (sum of allocated employees). The total demand in the Demand grid is 48 (sum of all nodes), and the total coverage in the Coverage grid is 25 (sum of all nodes). That means that 52% of the total demand is covered. As we want to obtain a threshold coverage level of 80%, this solution would be infeasible.

4.3. Metaheuristic solution approach

Covering models are typically classified as NP-hard, meaning they are hard to solve and require exponential time to be solved to optimality (Rajagopalan et al., 2008). Various methods have been developed to find good solutions in a reasonable time, of which one of them is called Simulated Annealing (SA). Simulated annealing is based on random local search. It starts with diversification and iteratively moves towards intensification. In this way, the SA algorithm, a metaheuristic, is able to approximate a global optimum in a large solution space (Kirkpatrick, Gelatt, & Vecchi, 1983). Besides the ability to approximate a global optimum, SA can quickly find a good solution. Therefore, we choose to use the SA algorithm to solve our model.

The SA algorithm starts with initializing its parameters. SA requires a cooling scheme, which includes a starting temperature (T_{start}), stopping temperature (T_{stop}) and a decrease factor (α), which iteratively decreases the temperature (T) until it reaches the stopping temperature, also known as the stopping criterium. Besides, it requires a Markov chain length and initial solution. This initial solution can be generated entirely at random and initialises the current and best solutions.

We adjust the general SA algorithm, described in Section 3.3, to comply with our problem. We see the pseudocode of the SA algorithm adapted to our problem in Figure 21. Here, we generated a neighbour solution by changing the number of employees or the location of employees in the current solution. We can accept the neighbour solution as current solution or as best solution. We accept the neighbour solution if it is better than the current solution. In two cases, the neighbour solution is better than the current solution:

1. The number of employees used in the neighbour solution (E') is less (better) than the employees used in the current solution (E) and coverage (C') is equal to or more than the threshold coverage (C^T);
2. The number of employees used in the neighbour solution (E') is equal to the employees used in the current solution (E), but the coverage of the neighbour solution (C') is better than the coverage of the current solution (C).

If the neighbour solution is not better than the current solution, we still accept the neighbour solution against a probability. This probability is based on the Boltzmann probability distribution and only considers the objective value of the number of employees used ($e^{-\frac{|E-E'|}{T}}$). A worse solution can be more employees and higher coverage (unnecessary deployment of employees) or coverage below the threshold level (independently of the number of employees). If the coverage is below the threshold coverage, the solution is infeasible. However, it could be that when swapping the employees to different locations, the coverage will be above the threshold coverage level. Due to the second case of accepting a neighbour solution described above, this process of optimizing positions in (at first) infeasible solutions is allowed and could lead to fewer employees, but in better positions.

By applying this Boltzmann probability, SA starts with accepting many (worse) solutions to explore the solution space (diversification). As the temperature iteratively decreases, the probability of random acceptance also decrease, and in the end, SA intensifies on improving the current solution only (intensification). Due to this process, the objective and coverage of the current solution (E, C) can also get worse. Therefore, the SA algorithm also stores the overall best solution so far (E^*, C^*). We update the best solution based on the same improvement conditions as when updating the current solution but now look at the number of employees used in the best solution (E^*) and the coverage of the best solution (C^*). Besides storing the objective and coverage of a solution, also the solution itself is stored. The solution represents the allocated employees and is described with E_{loc} for the current solution, E^*_{loc} for the best solution and E'_{loc} for the neighbour solution.

The SA algorithm explores the solution space by creating neighbour solutions, also known as neighbour solutions. The neighbour solution is created by the use of operators. In our case, an adjustment to the current solution is changing the number of employees or changing the position of employees. Changing the number of employees depends on the coverage of the current solution. As we allow worse solutions to be accepted based on the Boltzmann probability, the coverage of the current solution can be lower than the threshold coverage. If that is the case, we add an employee to the current solution at a random node at a random hour. On the other hand, if the coverage of the current solution is above the threshold coverage, we delete an employee from the current solution. However, in the case of coverage below the threshold coverage, moving employees to other nodes could also result in a solution with enough coverage. Therefore, we propose a 2-opt neighbourhood swap within the same hour, meaning we swap the employees on two random nodes, of which one need to have at least one employee. In Figure 22, we see the pseudocode of the procedure that creates the neighbour solution. We use a 50% probability to swap or add/delete an employee.

Simulated annealing

initialize $T = T_{start}, T_{stop}, MarkovLen, \alpha, C_T$
$E^*, C^*, E_{loc}^* = E, C, E_{loc} = initSolution$
while $T > T_{stop}$ **do**
    for $(m = 1;\ m < MarkovLen;\ m++)$ **do**
        $E', C', E'_{loc} = findNeighborSolution(E_{loc}, C)$
        if $\big((E' < E)$ **and** $(C' \geq C_T)\big)$ **or** $\big((E' = E)$ **and** $(C' > C)\big)$ **then**
            if $\big((E' < E^*)$ **and** $(C' \geq C_T)\big)$ **or** $\big((E' = E^*)$ **and** $(C' > C^*)\big)$ **then**
                $E^*, C^*, E_{loc}^* = E', C', E'_{loc}$
            end if
            $E, C, E_{loc} = E', C', E'_{loc}$
        else
            if $random(0,1) \leq \exp\left(-\frac{|E-E'|}{T}\right)$ **then**
                $E, C, E_{loc} = E', C', E'_{loc}$
            end if
        end if
    end for
    $T = \alpha T$
end while
return E^*, C^*

Figure 21: Pseudocode of simulated annealing algorithm

$findNeighborSolution$ procedure

if $random(0,1) \leq 0.5$ **then**
    $tmp = employees[hour, loc1]$
    $employees[hour, loc1] = employees[hour, loc2]$
    $employees[hour, loc2] = tmp$
elif $solutionCoverage < thresholdCoverage$ **then**
    $employees[loc, hour] \mathrel{+}= 1$
else
    $employees[loc, hour] \mathrel{-}= 1$
end if
return $employees, coverage(employees)$

Figure 22: Pseudocode of the neighbour solution procedure

To evaluate the performance of the procedure, we analyse the outcome of using only the add/delete structure (from now on called SA-1) versus the add/delete and swap structure (from now on called SA-2). The decision to swap or add/delete employees occurs with a probability. Lalla-Ruiz et al. (2020) proposed simulated annealing with variable neighbourhoods (SA-VNS). This eliminates the idea that the decision to swap or add/delete depends on a probability by introducing parameter k to the algorithm. Parameter k decides to swap ($k = 1$) or add/delete ($k = 2$). Initially, k is set to 1 and remains 1 as long as the current solution keeps improving. As soon as a worse solution is accepted, k is set to 2 and remains 2 until the current solution improves again.

As mentioned, whether or not a neighbour solution is better than the current (best) solution depends not only on the number of employees but also on the coverage of that neighbour solution. The easiest way to calculate the coverage of the neighbour solution is to re-evaluate every node on the grid. To do this, for every node on the grid, we need to know if a node is reached by employees allocated on any other node on the grid. This requires two nested for-loops checking every node on the grid against every other node on the grid, which, in terms of computational effort, results in poor running times. To overcome this problem, we make use of delta computations. Instead of updating every node on the grid, we only update the nodes close to the ones we change in the neighbour solution. In this manner, we define close as all nodes within reach of the node(s) where the solution is changed. This significantly reduces running times.

4.4. Conclusion

For our problem, the Location Set Covering Problem (LSCP) by Toregas et al. (1971) and the TIMEXCLP model of Repede and Bernardo (1994) are two models which are very relevant. In Section 4.1, we describe that we combine the LSCP and TIMEXCLP model to create a model that minimizes the number of incident handlers over different time instances by allocating them to achieve the constraint of a threshold coverage. In this way, the number of incident handlers is unrestricted and optimized by the model. Covering models typically use a binary coverage view, meaning a node is either covered or not. There exist functions that change this binary view to a gradual coverage view. We extend the model with a gradual coverage function where coverage decreases when distance increases away from an incident handler using a step-wise coverage function. We describe the mathematical model in Section 4.2.

We adapt the SA algorithm to suit our problem in Section 4.3. To keep computational effort to a minimum, we use a grid map instead of a map based on coordinates and use delta computations in the SA algorithm to calculate coverage. Besides, we define different versions of the SA algorithm regarding the neighbourhood structure. We evaluate the performance of the different versions in the next chapter, Chapter 5, more specifically in Section 5.1.2. We compare SA-1, SA-2 and SA-VNS, all using delta computations and for all scenarios.

5. Solution results

This chapter starts with analysing the performance of the solution design in Section 5.1. First, in Subsection 5.1.1, we evaluate the parameters of the SA algorithm. Next, in Subsection 5.1.2, we analyse the performance of the different neighbourhood structures. Finally, in Subsection 5.1.3, we compare the performance of the SA algorithm with the ILP model. After analysing the performance of the solution design, we analyse the results in a stochastic environment in Section 5.2. In Section 5.3, we provide insight based on all results and in Section 5.4, we analyse the influence of the threshold coverage level using a sensitivity analysis.

5.1. Simulated annealing performance

This section considers the performance of the SA algorithm. We discuss several experiments. We code all experiments in Python 3.8.8 in the Spyder software on a computer with an Intel Core i7-8750H processor.

5.1.1. Parameter tuning

Simulated annealing requires an initial solution and parameters. The initial solution can be entirely at random. There is an option to start with an empty solution for the initial solution, meaning no employees are allocated. In that case, the SA algorithm starts as a constructive heuristic until it reaches a feasible initial solution. The SA operators will keep adding employees until it reaches the threshold coverage. However, even though we use delta computations, we found that calculating the coverage repeatedly for every new employee randomly allocated requires unnecessary computational effort, as also discussed in Section 4.3. Therefore, we create an initial solution by randomly assigning a fixed number of employees to any hour and cell on the grid. After finishing the allocation, the coverage of the complete solution is calculated once without using delta computations for the complete solution. We found that the SA algorithm works best if the coverage is already close or above the threshold coverage level. Far below the threshold coverage results in the SA algorithm requiring significant computational effort to at least obtain a feasible solution. We found that by randomly assigning 400 AL employees, or in the case of the ICB specialism 800 ICB employees, the coverage level is around the threshold coverage of 80% or above.

The parameters of the SA algorithm determine the performance of the SA algorithm. We need to tune the parameters in a way that the SA algorithm, at first, accepts many worse solutions (this mainly depends on T_{start}), and in the end, only accepts better solutions (T_{stop}). The time it takes to go from T_{start} to T_{stop} depends on the decrease factor α and the number of solutions evaluated at a certain temperature T depends on the length of the Markov chain.

For the SA algorithm to accept many worse solutions at the start of the algorithm, the acceptance ratio at T_{start} needs to be close to one. In this context, we define the acceptance ratio as the number of worse solutions accepted divided by the number of worse solutions proposed. Figure 23 shows the acceptance ratio of the SA algorithm using a starting temperature of 50. Based on this figure, we fix T_{start} to 30, as the ratio is close to 1 at this temperature. Using a higher temperature does not influence the final solution much but requires more computational effort. Similarly, we also determine the value of T_{stop}. However, now we want the acceptance ratio to be zero to intensify on the best solution. Figure 23 shows the acceptance ratio per temperature. We see that around 0.2 degrees, the acceptance ratio becomes zero. From there on, we do not accept worse solutions anymore. However, we cannot fix T_{stop} based on Figure 23 only, because we observed that the solution still improves when the acceptance ratio is zero. Based on the figure, we only know that $T_{stop} \leq 0.2$ to have an acceptance ratio of zero in the final parts of the SA algorithm.

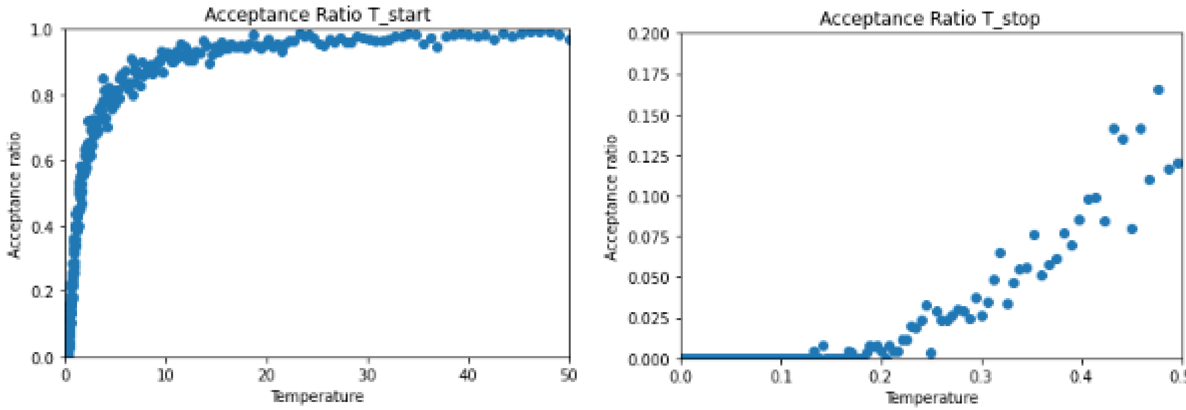


Figure 23: Acceptance ratio of the SA algorithm to determine T_{start} and T_{stop}

We analyse the value of T_{stop}, as well as the values of the other parameters (α and $MarkovChainLength$), by performing several experiments. We analyse all other parameters using experiments where we change the values of these parameters. Therefore, we perform 30 experiments to evaluate the performance using different settings of these parameters. We divide the experiments into groups of five. In almost all groups, we change the value of one parameter and fix the others per experiment within that group. For one group, we change α and $Markov$ chain length simultaneously to evaluate the influence they have on each other. Table 6 shows the groups and goals of these groups.

Table 6: Goal of a group of experiments to tune SA algorithm parameters

Group	Experiments	Goal
T_{stop}	1-5	Determine value of T_{stop}
$Markov$	6-10	Determine value of $Markov$
α	11-15	Determine value of α
α & $Markov$	16-20	Evaluate effect of increasing $Markov$ and decreasing α
$Markov$	21-25	Evaluate effect of low $Markov$ values with lower T_{stop}
$Markov$	25-30	Evaluate effect of $Markov$ with a slightly higher value of α

We run the SA algorithm three times in each experiment and store the overall best solution objective, average solution objective, and average time. In Table 7, we see the settings of the 30 experiments and the results of every experiment. Considering experiments 1-5, we see the best results with $T_{stop} = 0.01$. Next, in experiments 6-10 we look at the effect of the $Markov$ chain length. We see that we obtain the best results with $Markov = 1{,}000$. We also observe that running times significantly increase when increasing $Markov$. In experiments 11-15 we consider different values of α with T_{stop} and $Markov$ fixed. Here, we select $\alpha = 0.97$, because this provides good results and with allowable running time. Increasing α does slightly improve the solution, but against the cost of high running times. Therefore, in experiments 16-20 we evaluate different values of α and $Markov$ simulatenously to see if we can obtain good solutions with less running time. From these experiments, we see that we obtain the best solutions with $\alpha = 0.93$ and $Markov = 1{,}250$. However, by doing these experiments, we also observe that the solution improves significantly at the final stages of the SA algorithm. Therefore, we want to extend this period and do not choose $\alpha = 0.93$, but keep $\alpha = 0.97$. Moreover, we look at the effect of a shorter $Markov$ chain length with a lower T_{stop} to really focus on the final stages of the SA algorithm in experiments 21-25 with $\alpha = 0.97$. We see that $T_{stop} = 0.001$, $\alpha = 0.97$, $Markov = 500$ performs the best with allowable running times of around 3 minutes. Finally, in experiments 26-30, we want to see if increasing α (and also increasing T_{stop} to keep allowable running times) has a significant impact. However, this is not the case and thus we select $T_{stop} = 0.001,\ \alpha = 0.97, Markov = 500$ as the final parameter values of the SA algorithm. This gives us good solutions within 3 minutes. Longer running times do not significantly improve the objective anymore.

Table 7: Experiments to tune SA parameters

Grp.	Exp.	T_{start}	T_{stop}	α	$Markov$	Best	Average	Time [s]
T_{stop}	1	30	1	0.95	200	251	257	11.52
	2	30	0.5	0.95	200	221	230	13.38
	3	30	0.1	0.95	200	188	195	18.50
	4	30	0.05	0.95	200	183	190	20.91
	5	**30**	**0.01**	**0.95**	**200**	**179**	**183**	**24.93**
$Markov$	6	30	0.01	0.95	100	176	178	25.73
	7	30	0.01	0.95	200	172	174	42.58
	8	30	0.01	0.95	500	171	178	85.91
	9	**30**	**0.01**	**0.95**	**1000**	**162**	**164**	**168.81**
	10	30	0.01	0.95	2000	164	168	311.67
α	11	30	0.01	0.90	1000	165	166	73.87
	12	30	0.01	0.95	1000	163	165	145.81
	13	**30**	**0.01**	**0.97**	**1000**	**162**	**163**	**238.64**
	14	30	0.01	0.98	1000	160	161	356.24
	15	30	0.01	0.99	1000	158	159	713.22
α & $Markov$	16	30	0.01	0.94	1000	165	176	130.59
	17	**30**	**0.01**	**0.93**	**1250**	**164**	**165**	**155.00**
	18	30	0.01	0.92	1500	165	175	146.33
	19	30	0.01	0.91	2000	166	169	174.44
	20	30	0.01	0.90	5000	164	171	364.03
$Markov$	**21**	**30**	**0.001**	**0.97**	**500**	**163**	**164**	**188.68**
	22	30	0.001	0.97	400	165	169	147.68
	23	30	0.001	0.97	300	164	169	114.83
	24	30	0.001	0.97	200	167	168	84.47
	25	30	0.001	0.97	100	175	179	66.05
$Markov$	26	30	0.01	0.98	700	162	164	284.38
	27	30	0.01	0.98	600	163	170	237.11
	28	**30**	**0.01**	**0.98**	**500**	**162**	**166**	**208.94**
	29	30	0.01	0.98	400	182	184	152.81
	30	30	0.01	0.98	300	185	187	118.50
	FINAL	**30**	**0.001**	**0.97**	**500**	**163**	**172**	**171.51**

The final settings perform well for the AL specialism. We reduce the employees allocated in the final solution compared to the initial solution by a factor of two. When looking at the ICB specialism, we see the same reduction factor. However, the ICB specialism also requires two times more employees in the initial solution than the AL specialism initial solution. That means that the possible improvements also increase with a factor of two. For the ICB specialism, the general settings perform well. However, after analysing the performance of the SA algorithm for this specialism, we see that a Markov chain length of 1,000 performs significantly better but still obtains solutions in reasonable time.

5.1.2. Simulated annealing variants

For the neighbourhood structure, we evaluate three options. We first evaluate SA-1, which only adds or deletes employees based on the coverage of the current solution. Next, we evaluate SA-2, which besides adding or deleting employees based on the coverage, also swaps employees randomly. The decision to swap or add/delete occurs randomly with a probability of 50%. This percentage is evaluated with experiments and performs well. Lastly, we evaluate the SA-VNS algorithm. In the SA-VNS algorithm, the decision to swap or add/delete is not random anymore and now depends on a value k. The value k changes from 1 to 2 if the current solution is updated with a worse solution and resets to 1 if the current solution improves ($k = 1$ for swap, $k = 2$ for add/delete) following the SA-VNS idea of Lalla-Ruiz et al. (2020). We evaluate the three SA algorithms by running all scenarios once for each algorithm and store the objective and average running time. In Table 8, we see the performance of the three algorithms as well as the average duration using a specific algorithm. For the test, we run SA-1, SA-2 and SA-VNS simultaneously to obtain results quicker. All settings regarding software and processor are equal.

Table 8: Objective values and average durations of SA algorithm with different neighbourhood structures

	AL					
	Week			Weekend		
Scenarios	**SA-1**	**SA-2**	**SA-VNS**	**SA-1**	**SA-2**	**SA-VNS**
2018, 2019	237	**144**	208	184	**114**	185
Winter '18, '19	199	**104**	167	144	**76**	124
Spring '18, '19	182	**110**	180	144	**83**	138
Summer '18, '19	194	**113**	172	141	**82**	128
Autumn '18, '19	193	**117**	182	135	**75**	114
2020	204	**123**	181	162	**94**	156
Winter '20	180	**96**	155	113	**62**	101
Spring '20	160	**93**	139	81	**48**	76
Summer '20	175	**97**	157	111	**64**	99
Autumn '20	177	**97**	154	102	**62**	95
Average duration [sec]	135.95	186.68	124.36	121.18	172.62	124.93

	ICB					
2018, 2019	610	**423**	564	516	**344**	482
Winter '18, '19	527	**316**	463	400	**242**	345
Spring '18, '19	548	**332**	458	419	**254**	379
Summer '18, '19	511	**342**	485	425	**246**	367
Autumn '18, '19	528	**344**	478	389	**232**	333
2020	558	**375**	485	451	**279**	401
Winter '20	467	**280**	401	329	**195**	283
Spring '20	461	**271**	378	239	**154**	209
Summer '20	467	**292**	415	317	**197**	276
Autumn '20	456	**298**	423	300	**186**	268
Average duration [sec]	264.34	352.23	242.48	241.55	315.40	238.82

Based on these results, we conclude that the SA-2 structure performs the best in terms of the objective function (number of employees). For every scenario, the objective of SA-2 is the lowest. On average, the SA-2 algorithm performs 42% better than the SA-1 algorithm and 36% better than the SA-VNS algorithm in terms of objective. However, when looking at the average duration, we see that the SA-2 algorithm requires the longest running time. The main reason is that the swap operator needs to find a grid cell with at least one employee allocated to that cell. To swap, we randomly select a grid cell and continue to do so until we find a cell with at least one employee. Then we randomly select another cell and perform the swap. This process takes up significant time but also result in better solutions. Therefore, we select the neighbourhood structure used in the SA-2 algorithm. From here on, all results use this structure of randomly deciding to swap or add/delete employees from the solution.

5.1.3. ILP model vs SA

We also program the covering model as in ILP model. For this, we use the Python extension module Gurobi. This module allows us to program the model in Python easily. With the ILP model, we can check to performance, in terms of objective and duration, of the SA algorithm against the ILP model. Similar to the performance analysis on the neighbourhood structure in Section 5.1.2, we analyse the performance of the ILP model versus the SA algorithm. We again do this for every scenario and both specialism. All settings regarding software and processor remain the same. To run both the ILP model and SA algorithm for every scenario, we require 80 runs (two times all scenarios), which requires significant total running time. To obtain results in less total running time, we set a time limit for the ILP model. The SA algorithm obtains results in less than five minutes for the AL specialism, and to have similar running times, we select a time limit of five minutes for the ILP model. As soon as we reach the time limit, the ILP model will cut off and provide the best solution obtained within the time limit. We run every scenario one time only.

Table 9: Objective values of the ILP model and the SA algorithm and optimality gap of ILP model

	AL					
	Week			Weekend		
Scenarios	**ILP**	**ILP Gap**	**SA**	**ILP**	**ILP Gap**	**SA**
2018, 2019	135	2.22%	142	108	0.93%	115
Winter ‘18, ‘19	97	0.00%	102	73	0.00%	79
Spring ‘18, ‘19	103	0.00%	113	78	0.00%	82
Summer ‘18, ‘19	105	0.00%	112	75	0.00%	79
Autumn ‘18, ‘19	108	0.93%	115	71	0.00%	74
2020	116	0.00%	124	85	0.00%	90
Winter ‘20	85	0.00%	90	57	0.00%	65
Spring ‘20	81	0.00%	87	45	0.00%	48
Summer ‘20	89	0.00%	96	58	0.00%	63
Autumn ‘20	91	0.00%	99	56	0.00%	66
Average duration [s]	124.10		200.87	34.47		198.37

	ICB					
2018, 2019	405	0.99%	424	326	0.92%	346
Winter ‘18, ‘19	297	1.01%	313	220	0.00%	238
Spring ‘18, ‘19	312	0.64%	331	233	0.00%	256
Summer ‘18, ‘19	316	0.63%	336	224	0.00%	243
Autumn ‘18, ‘19	319	0.31%	344	215	0.00%	235
2020	355	1.13%	377	260	0.00%	282
Winter ‘20	260	0.00%	278	173	0.00%	191
Spring ‘20	244	0.00%	263	137	0.00%	152
Summer ‘20	270	0.37%	291	175	0.00%	190
Autumn ‘20	278	0.00%	300	168	0.00%	188
Average duration [s]	235.39		371.08	36.02		368.01

Table 10: Gap between the objective function of ILP model and SA algorithm

	AL		ICB	
Scenarios	**Wk**	**Wknd**	**Wk**	**Wknd**
2018, 2019	5.19%	6.48%	4.69%	6.13%
Winter ‘18, ‘19	5.15%	8.22%	5.39%	8.18%
Spring ‘18, ‘19	9.71%	5.13%	6.09%	9.87%
Summer ‘18, ‘19	6.67%	5.33%	6.33%	8.48%
Autumn ‘18, ‘19	6.48%	4.23%	7.84%	9.30%
2020	6.90%	5.88%	6.20%	8.46%
Winter ‘20	5.88%	14.04%	6.92%	10.40%
Spring ‘20	7.41%	6.67%	7.79%	10.95%
Summer ‘20	7.87%	8.62%	7.78%	8.57%
Autumn ‘20	8.79%	17.86%	7.91%	11.90%
Average	**7.00%**	**8.24%**	**6.69%**	**9.23%**

In Table 9, we see the solution objectives of every scenario and the optimality gap of the ILP model. Besides, in Table 10, we see the gap between the objective function of the ILP model and the SA algorithm. Based on these results, we conclude that the ILP model always outperforms the SA algorithm. Optimality gaps between both specialisms are similar but lower during the week scenarios compared to the weekend scenarios. During the weekend, the demand is lower and therefore the position where to allocate employees needs to be more precise. The SA algorithm is less able to do so. However, we run the scenarios one time only. Running the SA algorithm multiple times results in different solutions because of the random initial solution and possibly not optimal final solution (randomly found a good solution, but not proven to be the optimal solution).

5.1.4. Long run comparison

The ILP model finds optimal, or very good, solutions within the time limit of five minutes. However, we are also interested in what happens with long runs of the ILP model, especially in the case the ILP does not find optimal solutions within the five-minute time limit. In Table 11, we show the results of the long runs per scenario. The time limit is increased and set to 30 minutes. We see that the objectives and coherently ILP gaps reduce and more solutions are optimal. However, it requires significantly more running time without much improvement. Next, we show the gap to the SA objectives found in Table 9. The gap to the SA algorithm slightly increases.

Table 11: Objective values of the ILP model with optimality gaps and gap to SA algorithm

	AL					
	Week			Weekend		
Scenarios	**ILP**	**ILP Gap**	**SA Gap**	**ILP**	**ILP Gap**	**SA Gap**
2018, 2019	134	1.49%	5.97%	107	0.00%	7.48%
Winter '18, '19	97	0.00%	5.15%	73	0.00%	8.22%
Spring '18, '19	103	0.00%	9.71%	78	0.00%	5.13%
Summer '18, '19	105	0.00%	6.67%	75	0.00%	5.33%
Autumn '18, '19	108	0.93%	6.48%	71	0.00%	4.23%
2020	116	0.00%	6.90%	85	0.00%	5.88%
Winter '20	85	0.00%	5.88%	57	0.00%	14.04%
Spring '20	81	0.00%	7.41%	45	0.00%	6.67%
Summer '20	89	0.00%	7.87%	58	0.00%	8.62%
Autumn '20	91	0.00%	8.79%	56	0.00%	17.86%
Average	397.8 s	0.24%	7.08%	43.8 s	0.00%	8.34%

	ICB					
2018, 2019	405	0.99%	4.69%	325	0.62%	6.46%
Winter '18, '19	297	0.67%	5.39%	220	0.00%	8.18%
Spring '18, '19	311	0.00%	6.43%	233	0.00%	9.87%
Summer '18, '19	316	0.63%	6.33%	224	0.00%	8.48%
Autumn '18, '19	319	0.31%	7.84%	215	0.00%	9.30%
2020	354	0.85%	6.50%	260	0.00%	8.46%
Winter '20	260	0.00%	6.92%	173	0.00%	10.40%
Spring '20	244	0.00%	7.79%	137	0.00%	10.95%
Summer '20	270	0.37%	7.78%	175	0.00%	8.57%
Autumn '20	278	0.00%	7.91%	168	0.00%	11.90%
Average	1100.7 s	0.38%	6.76%	186.0 s	0.06%	9.26%

5.2. Stochastic analysis

The SA algorithm gives promising results. We see that during the busiest hours, the most employees are allocated, and during the more idle hours (night), we see fewer to no employees. We obtain solutions by solving the SA algorithm with deterministic input. However, incidents do not occur on a deterministic basis. Instead, they occur partly random. Partly because generally, incidents tend to occur more often when it is busier, either because of the hour of the day or the location. Due to this stochastic element, a good solution obtained with the SA algorithm using deterministic input might not be that good when considering stochasticity. Therefore, it is vital to analyse the performance of a solution in a stochastic environment. With a stochastic analysis, we can determine the robustness of a solution.

To analyse this stochastic performance, we use a simheuristic because we want to analyse multiple solutions and select the most robust solution for a given scenario. Simheuristic algorithms are often used to simulate real-world problems under uncertain conditions (Chica & Juan 2017). The simheuristic algorithm uses simulation to allow stochastic scenarios to be evaluated for a fixed solution. In this way, we can analyse the feasibility of a solution under uncertain conditions. We can select the most robust solution when doing so for various solutions. That is, instead of selecting the solution with the least employees, select the most stable solution regarding coverage, even though one solution requires more employees than another. The simheuristic algorithm we use to analyse the stochasticity is Monte-Carlo simulation. This technique allows us to simulate many different instances, every time slightly adjusted. We want to evaluate the performance of different solutions by simulating stochasticity in the incident scenarios, and with this technique, we can. The simheuristic with Monte-Carlo simulation is proven to be efficient and reliable (Lalla-Ruiz et al., 2020).

The simheuristic algorithm starts with initializing a deterministic scenario. Then, we obtain a solution using the SA algorithm with that deterministic scenario. The solution is saved and undergoes a short Monte-Carlo simulation. In every iteration of the Monte-Carlo simulation, the deterministic scenario is adjusted to simulate stochasticity. For every iteration, we store the coverage of the solution using the adjusted scenario. After the short Monte-Carlo simulation, we calculate and save the interquartile range (IQR) of the coverage. We repeat this process for every solution we want to evaluate. Next, we select the best solutions based on the IQR. We select solutions based on the IQR because we have already obtained good solutions in terms of the number of employees, and we now want to select the most robust solution of these solutions in terms of coverage. We re-evaluate these so-called elite solutions with a long Monte-Carlo simulation where we again store the coverage of every iteration of the simulation. After the long Monte-Carlo simulation, we generate box plots of the coverage of the solutions and calculate the final IQR to evaluate the robustness of the solutions. We select the overall best solution based on the IQR. In Figure 24, we see the pseudocode of the simheuristic. All steps explained here can be seen in this pseudocode.

Sim heuristic
$Det_{problem} =$ `deterministic problem`
$\boldsymbol{for}\ (i \in Solutions)\ \boldsymbol{do}$
$s[i] = SimulatedAnnealing(Det_{problem})$
$\boldsymbol{for}\ (j = 1; j < MC_{short}; j++)\ \boldsymbol{do}$
$values[j] = evaluate(s[i], Stoch_{problem})$
$\boldsymbol{end\ for}$
$statisticMeasure[i] = calculateStatistic(values)$
$\boldsymbol{end\ for}$
$eliteSolutions = statisticMeasure[top\ 5]$
$\boldsymbol{for}\ (i \in eliteSolutions)\ \boldsymbol{do}$
$\boldsymbol{for}\ (j = 1; j < MC_{long}; j++)\ \boldsymbol{do}$
$finalValues[i][j] = evaluate(s[i], Stoch_{problem})$
$\boldsymbol{end\ for}$
$\boldsymbol{end\ for}$
$Analysis(finalValues)$

Figure 24: Pseudocode of the sim heuristic

Looking at the Monte-Carlo simulation process more specific, we illustrate the process in Figure 25. The process starts by obtaining a solution with deterministic demand, the input for the simulation. Next, we fix the solution but adjust the demand. In our case, we adjust the current demand values of every demand node randomly +/- 10% of the current demand following a uniform distribution. Then we calculate the coverage of the fixed solution, but with the new (adjusted) demand, and store the coverage. The demand resets to the deterministic demand the process repeats. When the stopping criterium is reached, the simulation process stops.

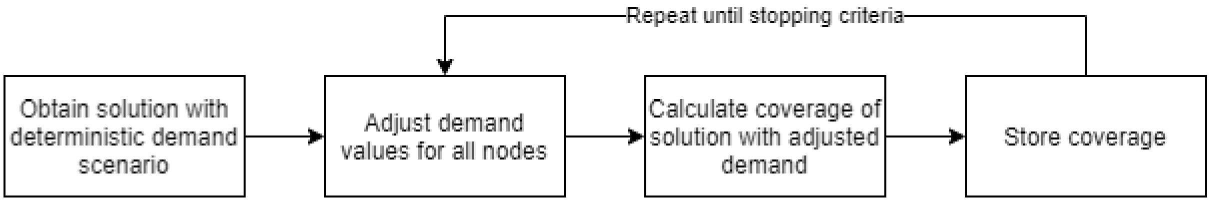


Figure 25: Monte-Carlo simulation process

As described, the simheuristic requires several parameters. It first requires the number of solutions to be evaluated. In our case, we first run the SA algorithm 25 times, meaning we create 25 deterministic solutions to be evaluated by the simheuristic. For each solution, we run a short Monte-Carlo simulation of 50 runs per solution in which we recalculate the coverage of the solution based on a stochastic scenario. After applying a short Monte-Carlo simulation on all 25 solutions, we select the five solutions with the lowest IQR, called the elite solutions. We apply a long Monte-Carlo simulation of 250 runs per solution for all five elite solutions and again recalculate the coverage of the solution based on the stochastic scenarios. With the coverage of different simulated scenarios stored per solution, we create box plots of these coverages. In Figure 26 we see the results of one scenario. We see the coverage (displayed on the y-axis) of five different elite solutions (displayed on the x-axis). The overall best solution is selected based on the box plot and the IQR.

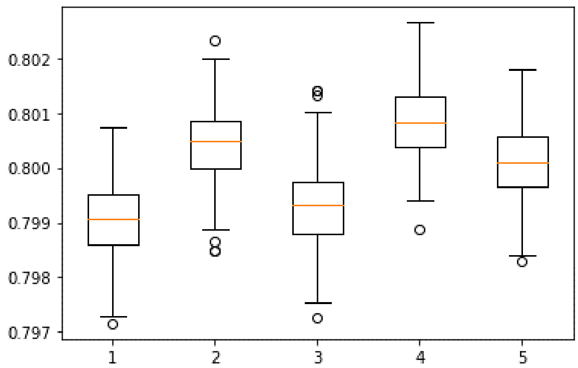


Figure 26: Boxplot results simheuristic scenario AL 2018-2019 week

5.3. Solution evaluation and comparison

As we tuned the parameters of the SA algorithm, selected the best-performing neighbourhood structure, evaluated the workings of the SA algorithm in comparison with the ILP model and implemented all this in a sim heuristic to be able to select a final solution this is good in terms of the number of employees as well as in terms of robustness; we can evaluate the solutions. We discuss the results of the simheuristic and a graphical representation of the solution for both specialisms. As we have 40 scenarios, we also have the results of all these scenarios. Here, we only provide insights based on the results. Specific results of the simheuristic to every scenario can be found in Appendix B. We choose the final solutions based on the IQR of the coverage because the lowest IQR means the most robust solution. Besides, we see that the number of employees required per solution does not change much, thus for the final, most robust solution, we do not select based on the number of employees.

We display the insights in two infographics. Figure 30 shows the first infographic and focuses on the AL specialism. Figure 31 shows the second infographic and focuses on the ICB specialism. In both infographics, first, we see the objective value (sum of all employees at all hours) for a weekday and weekend day for both 2020 and 2018 & 2019 as input. Next, we see the allocation of the solutions for 2018 and 2019. We do not provide insights in 2020 because these results are similar. Only fewer employees are allocated due to fewer incidents in the COVID year. Lastly, we provide the number of employees required per hour of the day in the infographics. We show the distribution of employees on weekdays and weekend days. As the solution obtained from the model includes the allocation for every hour of the day, we can make 24 allocation graphs, which is not very practical for the report. Therefore, we show the allocation results in one graph only. This graph is a cumulative representation of the allocation. That is, summing up all hours and plotting them in one map. We graphically explain this method in Figure 27, but for a case with only three hours.

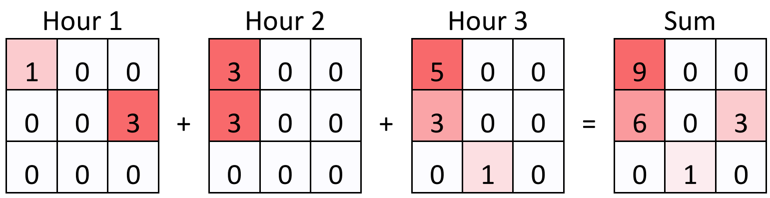


Figure 27: Graphical explanation of allocation graph

The final results, which are the most robust solutions to every scenario obtained with the simheuristic, let us see that the required number of employees decreases by around 19% during a weekend day compared to a weekday. The COVID year 2020, in general, requires 11% fewer employees during the week and 18% fewer employees during the weekend compared to the non-COVID years (2018, 2019). Considering the allocation, we see that the main focus areas during the week are Rotterdam, Amsterdam and the theoretical triangle Zwolle – Enschede – Arnhem, followed by the South (Limburg) and the North of the country. During the weekend, the North of the Netherlands disappears as a focus area.

The distribution of incident handlers across the hours of the day shows that, especially for the ICB specialism, there is less demand from 13.00 to 14.00. This suggests that the shift-change should be 1 hour earlier than the current shift change.

During the discussion of the data insights, which we provide in Section 2.4, we saw a peak in frequency during summer. This peak is also visible in the final results. However, this peak mainly occurs during the afternoon rush hour, as shown in Figure 28 and Figure 29. When discussing this with ProRail and looking at the input data, we see that hot weather conditions play an important role in causing a peak in incidents. During summer, many people use the train during afternoon rush hour; either to commute or for leisure.

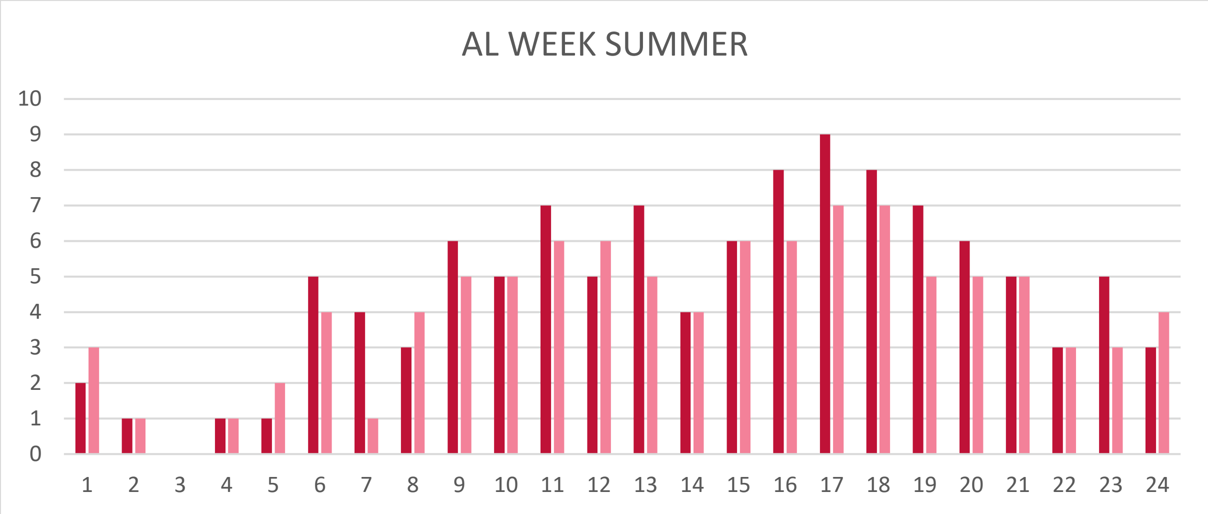


Figure 28: Distribution of ALs during a weekday in summer

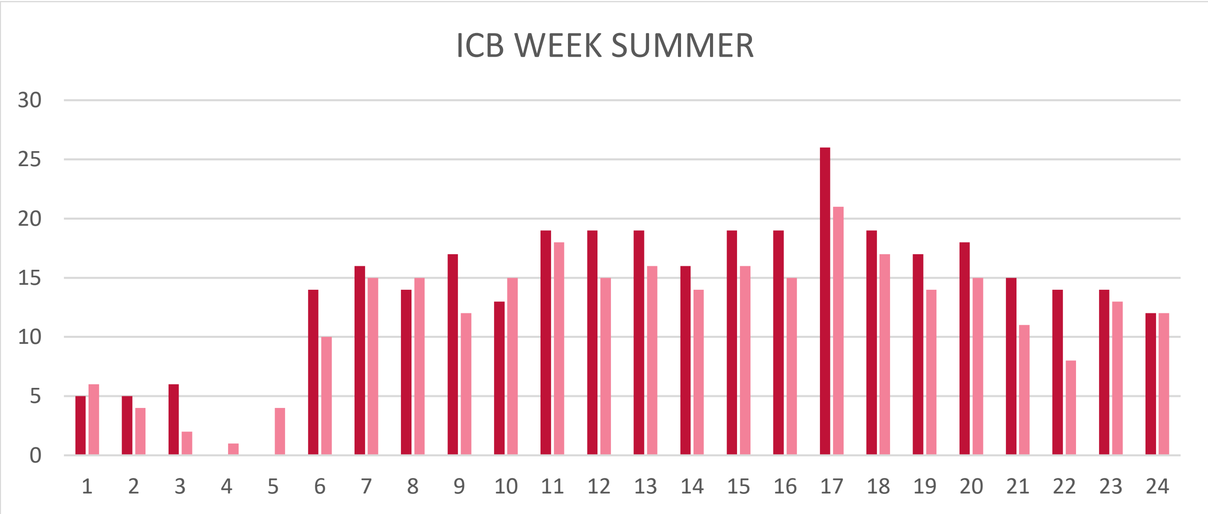


Figure 29: Distribution of ICB members during a weekday in summer

Based on all results, we recommend scheduling 7 AL employees in both shifts, one AL more than currently. The current number of minimal 20 ICB members is valid for the ICB team. On average, the number of ICB members is slightly higher, but as ProRail always schedules more than the minimum number, this minimum of 20 can remain the same. For the standby period during the evening, we can argue if the employees in the afternoon shift should all go on standby during the night. With only half of the employees on standby during the night, enough employees remain to cover demand.

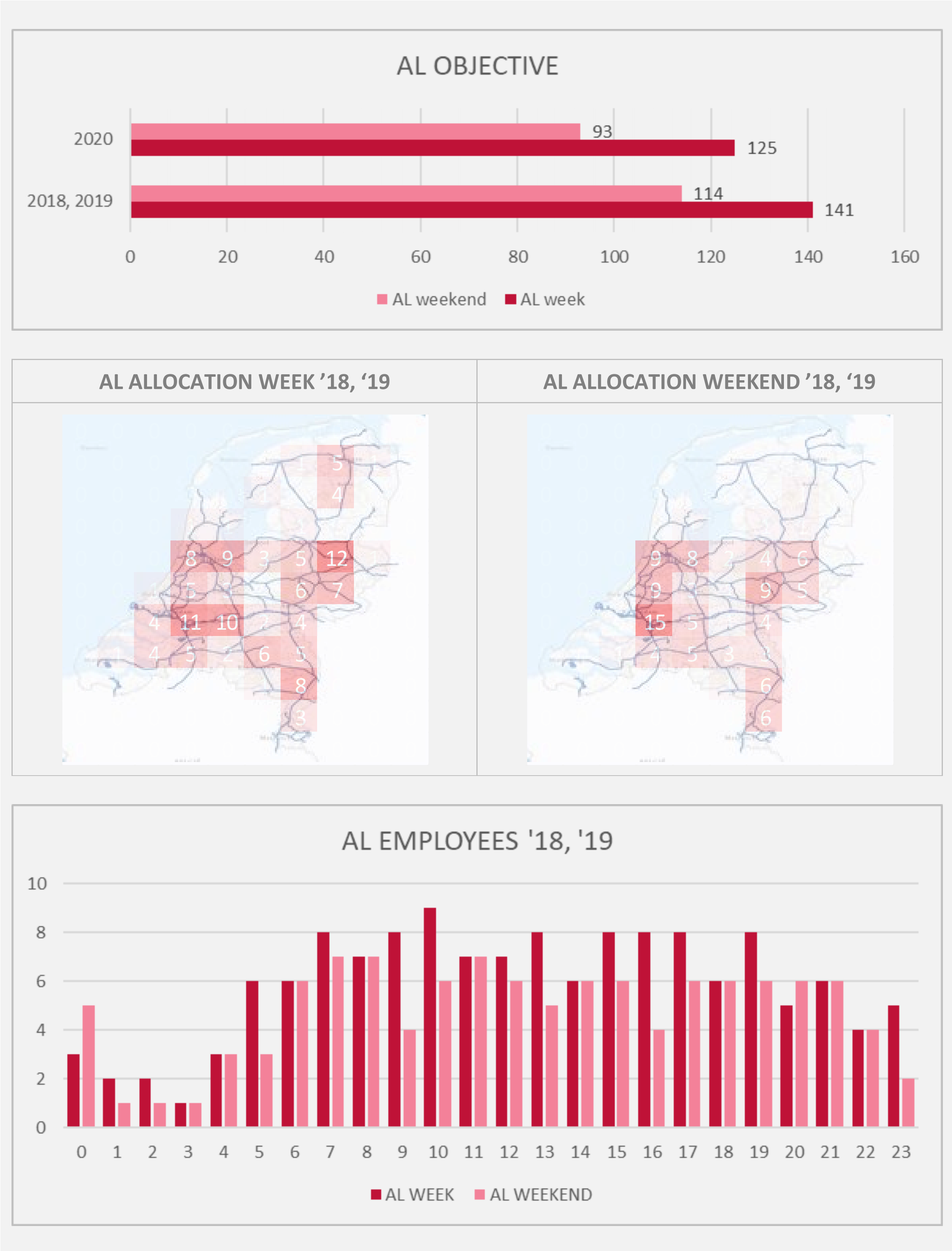


Figure 30: Infographic results AL specialism

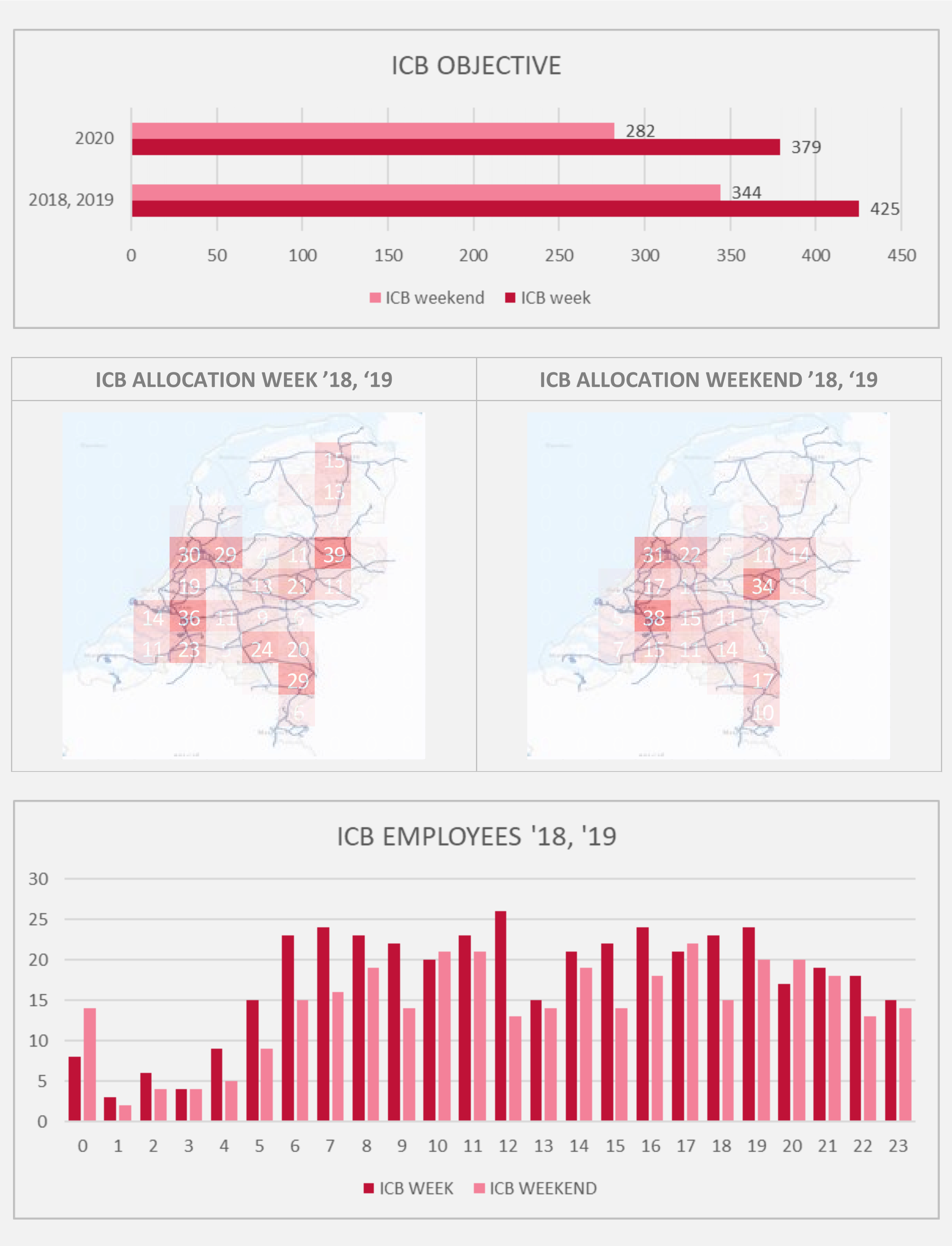


Figure 31: Infographic results ICB specialism (ICB members)

5.4. Sensitivity analysis threshold coverage

Until now, we considered a threshold coverage of 80%. We determined this value in consultation with ProRail. However, ProRail is also interested in what happens if we increase this threshold coverage level. This sensitivity analysis on the threshold coverage level is just theoretical to provide more insight into the influence of the threshold coverage value. However, it also has a practical benefit for ProRail, as, on holidays, generally many people use public transport, especially on, for example, Kingsday. When the required increase in employees is known to reach a higher coverage, they can use this on special occasions such as holidays. Therefore, we performed a sensitivity analysis on the threshold coverage. We increased the threshold coverage from 80% to 85%, 90% and 95% for both the AL specialism and ICB team. In Table 12, we see the results of this analysis. Here, we see the threshold coverage level in the most left column, and in the rightest column the average percentage change in the number of employees required compared to a threshold level of 80%. We also see, for every scenario, the number of employees and percentage change for that scenario. We run all scenarios once using the SA algorithm.

Table 12: Sensitivity analysis on threshold coverage

	AL				**ICB**				
	Week		**Weekend**		**Week**		**Weekend**		
Cov	**'18,'19**	**'20**	**'18,'19**	**'20**	**'18,'19**	**'20**	**'18,'19**	**'20**	**Avg**
80%	142	123	114	91	425	375	344	279	
	0%	0%	0%	0%	0%	0%	0%	0%	0%
85%	165	144	136	104	488	434	401	326	
	16%	17%	19%	14%	15%	16%	17%	17%	16%
90%	195	168	156	125	567	500	465	374	
	37%	37%	37%	37%	33%	33%	35%	34%	36%
95%	238	204	192	151	684	602	562	447	
	68%	66%	68%	66%	61%	61%	63%	60%	64%

Based on the results, we see that the number of employees needed exponentially grows when increasing the threshold coverage value. In this way, reaching a 95% coverage level is unrealistic as this requires 64% more employees, which is a significant number of employees. However, this sensitivity analysis implies a national coverage level. Our current model cannot provide results when only increasing coverage for a specific region. However, we assume these percentages to be reliable when requiring a higher coverage level for a specific region. This, because we still want to cover the same incidents but at a higher threshold coverage level.

5.5. Conclusion

The main goal of this chapter is to answer the fourth and last research question: *"How does the model perform for different scenarios of historical incidents?"*. To answer this model, we first analysed the performance of the SA algorithm in section 5.1. We tuned the different parameters used in SA such that the algorithm suits our problem and performs well. Besides, we analysed different neighbourhood structures and selected the neighbourhood structure where we included randomness in the decision between operators.

Section 5.2 describes the stochastic analysis. The SA algorithm obtains a solution based on deterministic input. In reality, incidents do not occur on a deterministic basis but randomly on a stochastic basis. Therefore, we used a simheuristic and analysed the results of the SA algorithm with this simheuristic. We performed a short Monte-Carlo simulation of 50 runs on 25 different solutions obtained with the SA algorithm and stored the IQR of the coverage of these solutions. We selected the top 5, the elite solutions, and performed a long Monte-Carlo simulation of 250 runs to select the most robust solution.

In Section 5.3, we analyse the final solutions. The main focus areas during the week are Rotterdam, Amsterdam and the theoretical triangle Zwolle – Enschede – Arnhem, followed by the South (Limburg) and the North of the country. During the weekend, the North of the Netherlands disappears as a focus area. Based on all results, we recommend scheduling 7 AL employees in both shifts, one AL more than currently. The current number of minimal 20 ICB members is valid for the ICB team.

Finally, Section 5.4 evaluates the influence of the threshold coverage level using a sensitivity analysis. We increased the threshold coverage from 80% to 85%, 90% and 95% for both the AL specialism and ICB team. We see that when increasing the threshold coverage level, the required number of employees exponentially grows. Nationwide it is not realistic to increase the number of employees exponentially to reach a higher coverage level. However, this is realistic in local areas. ProRail can decide to increase the local coverage during, for example, local events.

6. Conclusions and recommendations

In Section 6.1 of this chapter, we conclude the research carried out. We provide an answer to the main research question with the help of the sub-research questions answered in the previous chapters. Next, in Section 6.2, we provide recommendations based on the results of both the research and the literature review. These recommendations mainly address the implementation of results. Lastly, in Section 6.3, we discuss the limitations of this research and possible further research topics.

6.1. Conclusions

This section concludes the research carried out. The research starts with describing the problem. From this problem description, we found the main problem goal. That is, the development of an optimization model that guarantees a threshold coverage by allocating specialization-specific incident handlers based on historical data. As ProRail transfers from a regional scheduling approach to a nationwide scheduling approach, such a model would help create a more advanced and efficient schedule of incident handlers. Therefore, with this research, we focus on how to do this. We describe this with the following research question:

How can optimization techniques be used to guarantee coverage by allocating specialization-specific incident handlers based on data?

There is no straightforward answer to this question. Therefore, we use sub-research questions. All chapters contribute to answering the sub-research questions. With the answers to the sub-research questions, we answer the main research question. The sub-research questions are as follows:

- How is the allocation and scheduling of incident handlers currently working?
- What has been proposed in the literature for solving the specialism-specific capacity allocation problem to guarantee coverage?
- How should the solution approach be designed for the capacity allocation problem?
- How does the model perform for different scenarios of historical incidents?

The first sub-research question addresses the current process at ProRail. In Chapter 2, we discuss the current process in the problem context. Currently, the capacity is based on trial-and-error and adjusted accordingly. Besides, ProRail schedules incident handlers without considering their specialism as this was not used before and is a new idea at ProRail. Looking at the operational planning, ProRail has already optimized this with the Optimal Deployment system, creating a to-do list for incident handlers to perform the secondary tasks in the desired area. We obtain insight into the incident frequencies and incident density across the country from historical incident data. For example, the most incident-dense area is The Randstad and the busiest moment of the day is the afternoon rush hour.

The second sub-research question focuses on literature. Chapter 3 discusses the literature review, looking at what has been proposed for solving specialism-specific allocation problems to guarantee a coverage level. As there exist many variations, it is important to note that we classify our problem as a strategic problem. Classifying as a strategic problem significantly affects the models to be considered from the literature as we exclude operational and tactical aspects such as travelling speed, start location of shifts and shifts in general. Allocation problems with a guaranteed coverage level quickly converge to coverage models. There are many coverage models, but the TIMEXCLP model of Repede and Bernardo (1994) is most appropriate due to the possibility of solving the model for different time instances and reaching an overall coverage level. The TIMEXCLP model considers a node fully covered or not. There exist functions that change this binary view to a gradual coverage view. In these functions, coverage of an allocated incident handler gradually decreases when increasing the distance from that incident handler. This creates a more realistic view of coverage. Of these functions, the stepwise coverage function is most appropriate for our problem as this is a simple but, on a strategic level, realistic view of the coverage.

In general, covering problems are hard to solve to optimality. Therefore, we use a metaheuristic to solve our problem. Out of various metaheuristic search methods, simulated annealing is an appropriate metaheuristic. Simulated annealing is a metaheuristic that has been successfully implemented to solve covering models. It is a method that is usually easily implemented. Also, it generally requires less computational effort than more sophisticated procedures such as tabu searches and genetic algorithms. Simulated annealing can escape from a local optimum by allowing hill-climbing. Simulated annealing starts with accepting almost all solutions, but it step-by-step starts accepting only improvements (Kirkpatrick et al., 1983). In general, the heuristic finds reasonable good solutions in acceptable computational time. However, simulated annealing obtains a solution based on a deterministic input. Incidents do not occur on a deterministic base but a stochastic base. With a simheuristic, we can analyse the robustness of a solution by changing the deterministic using Monte-Carlo simulation.

Next, in Chapter 4, we design the solution approach to our problem, the third sub-research question. We consider several assumptions in our model, such as only considering starting moments of incidents and equally distributed service distances across the country. Using a coordinate-oriented map requires significant computational effort. Therefore, we transform a coordinate-oriented map into a grid map. Together with ProRail, we define scenarios we want to test. The final covering model we use to solve our problem is relatively simple but corresponds with a strategic planning level. Together with ProRail, we defined several scenarios. We look at all seasons as well as complete years. However, we split the data of 2020 from the dataset as 2020 is significantly influenced by COVID lockdowns. We obtained solutions for all scenarios and also considered the robustness performance with stochastic analysis.

Looking at the results of all experiments, we optimized the parameters of the simulated annealing algorithm to suit our model. Moreover, we checked the performance of the SA algorithm with different neighbourhood structures as well as versus a programmed ILP model. The final results, which are the most robust solutions to every scenario obtained with the simheuristic, let us see that the main areas of focus during the week are Rotterdam, Amsterdam and the theoretical triangle Zwolle – Enschede – Arnhem, followed by the South (Limburg) and the North of the country. The required number of employees decreases by around 19% during a weekend day compared to a weekday. The COVID year 2020, in general, requires 11% fewer employees during the week and 18% fewer employees during the weekend and holds for both the AL and ICB specialism.

Many different variations of the facility location and allocation problem exist for a long period already as described in the literature review in Chapter 3. With this research, we create a new application of such problems. We show the working and possibilities of applying simulated annealing to obtain good results quickly. Moreover, we provide robust solutions with the use of a simheuristic. Practically, we provide the number of employees, location and time of day to ProRail. This provides them insights usable for their planning systems.

6.2. Recommendations

We can make several recommendations based on the research; some regarding the results, others regarding the general working at ProRail. We start with the general recommendations. As mentioned in the problem context in Chapter 2, the deployment data is limited from 2020 onwards. This deployment data includes the deployment of ALs and ICB members in general. We cannot recommend obtaining the deployment data before 2020 as this is not registered. However, we can make recommendations on the way of registering deployment. ProRail aims to transfer from a general ICB member view to a specialism-specific view. To be able to make predictions on the number of specialism-specific employees required for every incident, we recommend registering specialism-specific deployment from now onwards. We assume that this requires a slight change in the current system, but it will significantly help the transfer to a specialism-specific ICB department. By doing so, more in-depth analyses about the deployment per incident label and precise estimations about the required employees of a specific specialism can be made.

Next to the general recommendations, we can also make recommendations based on the results. Currently, the morning shift starts at 06.00 and ends at 15.00. The afternoon/evening shift starts at 14.00 (meaning 1-hour overlap between shifts to handover work) and officially ends at 19.30. However, afternoon-shift employees remain on standby all night till the morning shift starts. Based on the results, we see a significant increase in demand from 06.00 in the morning onwards and, for the ICB team, see a decrease in demand during lunch hours, at around 13.00 to 14.00. Therefore, the starting moment of the shifts seems to be in line with the results, but the shift change could be 1 hour earlier. The transfer from working to on-standby at 19.30 in the evening is valid and does not require any change. If ProRail decides to reconsider shifts completely, it might be beneficial to have different variable shifts on a day. For example, the morning and afternoon shifts change from 1 starting moment to multiple starting moments. We graphically show this idea in Figure 32.

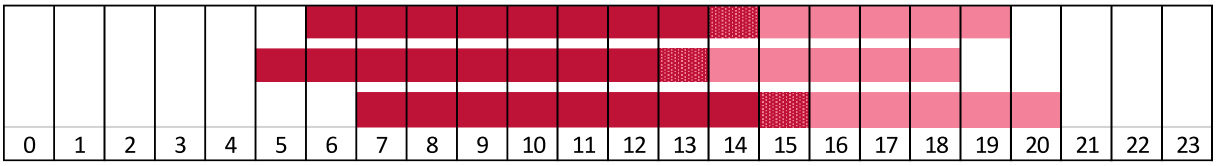


Figure 32: Variable shift hours

We recommend scheduling 7 AL employees in both shifts, one AL more than currently. The current number of minimal 20 ICB members is valid for the ICB team. On average, the number of ICB members is slightly higher, but as ProRail always schedules more than the minimum number, this minimum of 20 can remain the same. For the standby period during the evening, we recommend considering if the employees in the afternoon shift should all go on standby during the night. With only half of the employees on standby during the night, enough employees remain to cover demand.

6.3. Limitations and further research

This research aims to allocate incident handlers so that a service level is guaranteed. We do this with a mathematical model. However, this model has some limitations. As mentioned in Section 4.1.1, we assume no travelling times in the model. That means that the model can allocate employees at one hour to the North of The Netherlands and the other hour to the South of The Netherlands. This assumption is infeasible in reality as it is not possible to cross the country in one hour. We allow this limitation as we only look at strategic advice without any tactical or operational component.

Besides no travelling times, we also consider only starting moments of incidents and solve the model for every hour of a day. The incident handlers of ProRail solve most incidents within one hour. Therefore, we made this assumption not to consider the incident duration. However, a significant number of incidents still require more than one hour to be solved. That means that there is a possibility that an incident handler is busy for more than one hour. Therefore, the decision only to consider starting moments is a limitation on the model to not being able to consider the incident duration.

Further research of this research can be in the direction of developing a model which considers both travelling times and incident duration. This will create a more realistic model. For now, we choose not to consider this as ProRail is undergoing a transition to planning based on data and the current model already gives more than enough insights on a strategic level.

References

Adenso-Díaz, B., & Rodríguez, F. (1997). A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region. *Omega, 25*(2), 181-187. doi:10.1016/s0305-0483(96)00058-8

Aickelin, U. (2002). An indirect genetic algorithm for set covering problems. *Journal of the Operational Research Society, 53*(10), 1118-1126. doi:10.1057/palgrave.jors.2601317

Araz, C., Selim, H., & Ozkarahan, I. (2007). A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers & Operations Research, 34*(3), 705-726. doi:10.1016/j.cor.2005.03.021

Aytug, H., & Saydam, C. (2002). Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study. *European Journal of Operational Research, 141*(3), 480-494. doi:10.1016/s0377-2217(01)00260-0

Bansal, M., & Kianfar, K. (2017). Planar Maximum Coverage Location Problem with Partial Coverage and Rectangular Demand and Service Zones. *INFORMS Journal on Computing, 29*(1), 152-169. doi:10.1287/ijoc.2016.0722

Batta, R., Dolan, J. M., & Krishnamurthy, N. N. (1989). The Maximal Expected Covering Location Problem: Revisited. *Transportation Science, 23*(4), 277-287. doi:10.1287/trsc.23.4.277

Beasley, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research, 94*(2), 392-404. doi:10.1016/0377-2217(95)00159-x

Berman, O., & Krass, D. (2002). The generalized maximal covering location problem. *Computers & Operations Research, 29*(6), 563-581. doi:10.1016/s0305-0548(01)00079-x

Berman, O., Krass, D., & Drezner, Z. (2003). The gradual covering decay location problem on a network. *European Journal of Operational Research, 151*(3), 474-480. doi:10.1016/s0377-2217(02)00604-5

Bianchi, G., & Church, R. L. (1988). A hybrid fleet model for emergency medical service system design. *Social Science & Medicine, 26*(1), 163-171. doi:10.1016/0277-9536(88)90055-x

Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research, 147*(3), 451-463. doi:10.1016/s0377-2217(02)00364-8

CBS. (2009). Netherlands has busiest railway network in the EU. Retrieved from https://www.cbs.nl/en-gb/news/2009/09/netherlands-has-busiest-railway-network-in-the-eu

Charnes, A., & Cooper, W. W. (1959). Chance-Constrained Programming. *Management Science, 6*(1), 73-79. doi:10.1287/mnsc.6.1.73

Chica, M., & Juan , A. A. (2017). Why Simheuristics? Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation. *SSRN Electronic Journal*. doi:10.2139/ssrn.2919208

Chiyoshi, F., & Galvão, R. D. (2000). A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research, 96*(1/4), 61-74. doi:10.1023/a:1018982914742

Church, R. L., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association, 32*(1), 101-118. doi:10.1007/bf01942293

Church, R. L., & Roberts, K. L. (1983). Generalized coverage models and public facility location. *Papers of the Regional Science Association, 53*(1), 117-135. doi:10.1007/bf01939922

Daskin, M. S. (1983). A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution. *Transportation Science, 17*(1), 48-70. doi:10.1287/trsc.17.1.48

De Armas, J., Juan, A. A., Marquès, J. M., & Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society, 68*(10), 1161-1176. doi:10.1057/s41274-016-0155-6

de León, A. D., Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2021). A simulation–optimization framework for enhancing robustness in bulk berth scheduling. *Engineering Applications of Artificial Intelligence, 103*. doi:10.1016/j.engappai.2021.104276

Doerner, K. F., Gutjahr, W. J., Hartl, R. F., Karall, M., & Reimann, M. (2005). Heuristic Solution of an Extended Double-Coverage Ambulance Location Problem for Austria. *Central European Journal of Operations Research, 13*(4), 325-340.

Economy, T. G. (2019). Railroad infrastructure quality - Country rankings. Retrieved from https://www.theglobaleconomy.com/rankings/railroad_quality/

Eiselt, H. A., & Marianov, V. (2009). Gradual location set covering with service quality. *Socio-Economic Planning Sciences, 43*(2), 121-130. doi:10.1016/j.seps.2008.02.010

Galvão, R. D., Chiyoshi, F. Y., & Morabito, R. (2005). Towards unified formulations and extensions of two classical probabilistic location models. *Computers & Operations Research, 32*(1), 15-33. doi:10.1016/s0305-0548(03)00200-4

Galvão, R. D., & ReVelle, C. (1997). A Lagrangean heuristic for the maximal covering location problem. *Location Science, 5*, 72-72. doi:10.1016/S0966-8349(97)83342-6

Gendreau, M., Laporte, G., & Semet, F. (1997). Solving an ambulance location model by tabu search. *Location Science, 5*(2), 75-88. doi:10.1016/s0966-8349(97)00015-6

Ghaemi, N., Cats, O., & Goverde, R. M. P. (2017). Railway disruption management challenges and possible solution directions. *Public Transport, 9*(1-2), 343-364. doi:10.1007/s12469-017-0157-z

Heerkens, H. (Producer). (2015). Microlectures on MPSM and Research Cycle. Retrieved from https://vimeo.com/showcase/2938606

Huizing, D., Schäfer, G., van der Mei, R. D., & Bhulai, S. (2020). The median routing problem for simultaneous planning of emergency response and non-emergency jobs. *European Journal of Operational Research, 285*(2), 712-727. doi:10.1016/j.ejor.2020.02.002

Iannoni, A. P., Morabito, R., & Saydam, C. (2007). A hypercube queueing model embedded into a genetic algorithm for ambulance deployment on highways. *Annals of Operations Research, 157*(1), 207-224. doi:10.1007/s10479-007-0195-z

Jaramillo, J. H., Bhadury, J., & Batta, R. (2002). On the use of genetic algorithms to solve location problems. *Computers & Operations Research, 29*(6), 761-779. doi:10.1016/s0305-0548(01)00021-1

Jayaraman, V., & Srivastava, R. (1995). A service logistics model for simultaneous siting of facilities and multiple levels of equipment. *Computers & Operations Research, 22*(2), 191-204. doi:10.1016/0305-0548(94)e0013-w

Jia, H., Ordóñez, F., & Dessouky, M. M. (2007). Solution approaches for facility location of medical supplies for large-scale emergencies. *Computers & Industrial Engineering, 52*(2), 257-276. doi:10.1016/j.cie.2006.12.007

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2*, 62-72. doi:10.1016/j.orp.2015.03.001

Karasakal, O., & Karasakal, E. K. (2004). A maximal covering location model in the presence of partial coverage. *Computers & Operations Research, 31*(9), 1515-1526. doi:10.1016/s0305-0548(03)00105-9

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E.-G. (2021). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*. doi:10.1016/j.ejor.2021.04.032

Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671-680. doi:10.1126/science.220.4598.671

Lalla-Ruiz, E., Heilig, L., & Voß, S. (2020, 2020//). *Assessing Simulated Annealing with Variable Neighborhoods.* Paper presented at the Learning and Intelligent Optimization, Cham.

Li, X., Zhao, Z., Zhu, X., & Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research, 74*(3), 281-310. doi:10.1007/s00186-011-0363-4

Marianov, V., & Revelle, C. (1994). The queuing probabilistic location set covering problem and some extensions. *Socio-Economic Planning Sciences, 28*(3), 167-178. doi:10.1016/0038-0121(94)90003-5

Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research, 63*(5), 511-623. doi:10.1007/bf02125421

Pirkul, H., & Schilling, D. A. (1991). The Maximal Covering Location Problem with Capacities on Total Workload. *Management Science, 37*(2), 233-248. doi:10.1287/mnsc.37.2.233

RailTech. (2019). Dutch passenger trains set new punctuality record. Retrieved from https://www.railtech.com/policy/2019/01/03/dutch-passenger-trains-set-new-punctuality-record/?gdpr=accept

Rajagopalan, H. K., Saydam, C., & Xiao, J. (2008). A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research, 35*(3), 814-826. doi:10.1016/j.cor.2006.04.003

Rajagopalan, H. K., Vergara, F. E., Saydam, C., & Xiao, J. (2007). Developing effective meta-heuristics for a probabilistic location model via experimental design. *European Journal of Operational Research, 177*(1), 83-101. doi:10.1016/j.ejor.2005.11.007

Repede, J. F., & Bernardo, J. J. (1994). Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. *European Journal of Operational Research, 75*(3), 567-581. doi:10.1016/0377-2217(94)90297-6

ReVelle, C. (1989). Review, extension and prediction in emergency service siting models. *European Journal of Operational Research, 40*(1), 58-69. doi:10.1016/0377-2217(89)90272-5

Revelle, C., & Hogan, K. (1989). The maximum reliability location problem and α-reliable p-center problem: Derivatives of the probabilistic location set covering problem. *Annals of Operations Research, 18*(1), 155-173. doi:10.1007/bf02097801

Schilling, D., Elzinga, D. J., Cohon, J., Church, R., & ReVelle, C. (1979). The Team/Fleet Models for Simultaneous Facility and Equipment Siting. *Transportation Science, 13*(2), 163-175. doi:10.1287/trsc.13.2.163

Sinha, S. (2021). 10 Major Ports in Europe. Retrieved from https://www.marineinsight.com/know-more/major-ports-in-europe/

Swamidass, P. M. (2000). Hierarchical Planning and Control. In P. M. Swamidass (Ed.), *Encyclopedia of Production and Manufacturing Management* (pp. 263-263). Boston, MA: Springer US.

Tavakoli, A., & Lightner, C. (2004). Implementing a mathematical model for locating EMS vehicles in fayetteville, NC. *Computers & Operations Research, 31*(9), 1549-1563. doi:10.1016/s0305-0548(03)00108-4

Toregas, C., Swain, R., ReVelle, C., & Bergman, L. (1971). The Location of Emergency Service Facilities. *Operations Research, 19*(6), 1363-1373. doi:10.1287/opre.19.6.1363

van den Berg, P. L., & Aardal, K. (2015). Time-dependent MEXCLP with start-up and relocation cost. *European Journal of Operational Research, 242*(2), 383-389. doi:10.1016/j.ejor.2014.10.013

Appendices

Appendix A – Frequency of TIS and Incident Labels

TIS	**Description**	**Freq**
TIS 1	Disruption train operations	16461*
1.0	*Various situations, handled by contractors (not used in project)*	47811
1.1	A failure causing delays of between 5 and 30 minutes	4995
1.2	A failure causing delays of more than 30 minutes	8103
1.3	Total blockade	3313
1.4	Total blockade with effect on a large part of the country	50
TIS 2	Fire	376
2.1	Bush fire	254
2.2	Small fire in a train or station	103
2.3	Major fire in a train	3
2.4	Major fire in a station or tunnel	16
TIS 3	Collision or derailment	1842
3.1	Collision with a person, bicycle or other small object	1748
3.2	Collision with shunting unit or small road vehicle	76
3.3	Derailment with casualties or collision between train or large road vehicle where carriages are not deformed, tilted or stacked	11
3.4	Derailment with casualties or collision between train or large road vehicle where carriages are deformed, tilted or stacked	7
TIS 4	Hazardous substances	200
4.1	Minor incident involving hazardous substances	195
4.2	Fire involving hazardous substances	3
4.3	Leakage of a hazardous substance where the effects are limited to the source area	2
4.4	Incident involving hazardous substances in which there is clearly an effect area	0
TIS 5	Bomb threat	107
5.1	Anonymous bomb threat or suspicious behaviour	39
5.2	Suspicious object or bomb discovery on the open track	2
5.3	Suspicious object or bomb discovery in a train at a station, at a station or in a tunnel	65
5.4	Bomb explosion in a train, at a station or in a tunnel	1

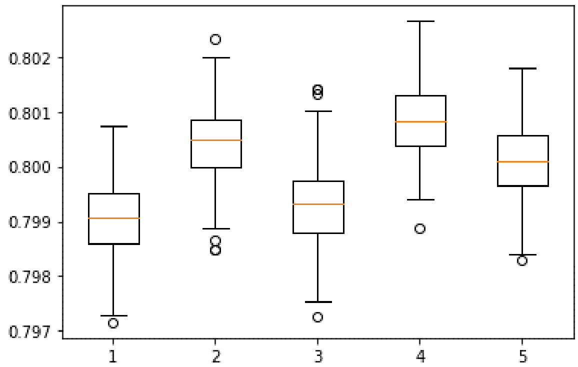
*) without TIS 1.0

Incident Label	Freq
(Bomb)explosion	*2*
(Imminent) failure of IT systems	172
Automatic failure detection	52
Bomb discovery NGCE	*17*
Bomb discovery or suspicious object	*79*
Bomb threat or suspicious behaviour	*23*
Bridge failure or defect	531
Collision (motor or moped) cyclist	19
Collision between trains	4
Collision buffer stop	33
Collision by train of traffic with bridge or viaduct	208
Collision large animal	68
Collision large road vehicle	22
Collision of shunting parts with each other	22
Collision person	888
Collision small road vehicle	98
Collision with (infra)object(s)	885
Crossing failure / defect	4201
Defective infrastructure	235
Defective material	17781
Defective overhead line or voltage-free	660
Derailment (casualties unknown)	1
Derailment (without casualties)	98
Disturbance due to calamity abroad	364
Disturbance due to health condition of traveller(s) or staff	916
Disturbance due to logistics problem/error	919
Disturbance due to object / vehicle / animal(s) on or near the track	2253
Disturbance due to order / assistance of emergency services	2444
Disturbance due to persons on or near the track	11667
Disturbance due to traveller(s) or staff by behaviour	1505
Disturbance due to vandalism or theft	303
Driving direction failure	211
Environmental damage on railway site	22
Failure control or communication systems	137
Failure control equipment GSM-Rail	21
Failure GSM-R / Intel system	41
Failure hill system	31
Failure security systems	144
Fire or smoke (notification) infra	37
Fire or smoke (notification) material	81
Fire or smoke (notification) post-T	2
Fire or smoke (notification) ProRail buildings	12
Fire or smoke (notification) roadside	211
Fire or smoke (notification) station	47

Fire or smoke (notification) tunnel	54
Fire or smoke (notification) under material	27
Gas leakage on railway site	8
Gas leakage outside railway site	17
High water	12
Infra ATB	382
Infra other	1371
Infra surroundings	961
Leaking or stinking carriage	192
Lightning strike	*8*
Low temperatures	11
Near collision	887
Other	315
Other security incidents	139
Overtime rail maintenance	334
Post failure	6
Power failure	360
Section failure	5276
Signal failure	928
Signalling failure	27
Slippery tracks	34
Smouldering railway sleeper/ switch fire	184
Snow conditions	*6*
Stop signal passage - no endangerment - train runs without permission	623
Stop signal passage - train runs without permission	*124*
Stop signal passage - train runs without permission with danger	*3*
Stop signal passage - train runs without permission without danger	*50*
Stop signal passage - with endangerment - train runs without permission	21
Strong wind	98
Suspicious behaviour or bomb threat	*9*
Suspicious object or bomb discovery	*7*
Switch failure / defect	4757
Track condition	1965
Train stoppage without contact in tunnel	*1*
Tunnel alarm due to gas detection (LEL)	*1*
Tunnel alarm due to high liquid level notification	*1*
Tunnel alarm due to train stoppage or (automatic) fire alarm	10
Tunnel failure	82
Urgent repairs	39

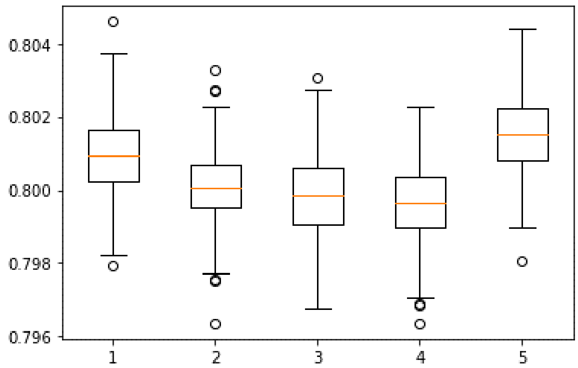
Appendix B – All results

AL 2018-2019 WEEK



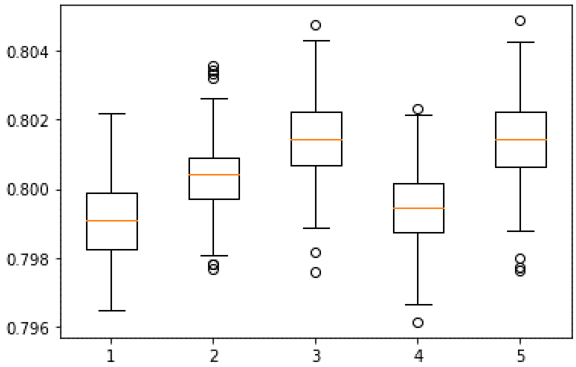
	1	2	3	4	5
Objective	141	141	141	143	144
Median	0.79907	0.80050	0.79932	0.80083	0.80010
Q1	0.79859	0.79999	0.79880	0.80038	0.79967
Q3	0.79952	0.80086	0.79974	0.80130	0.80057
IQR	0.00092	0.00087	0.00094	0.00092	0.00090

AL 2018-2019 WINTER WEEK



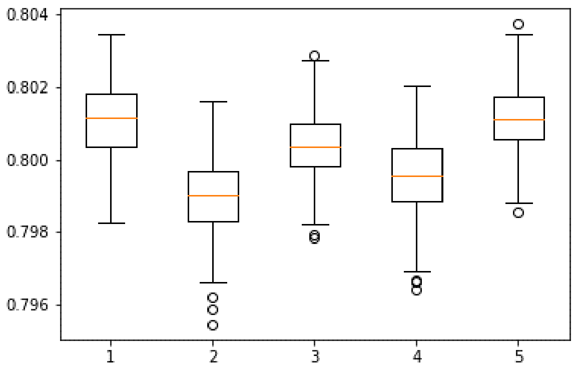
	1	2	3	4	5
Objective	104	103	103	103	103
Median	0.80095	0.80008	0.79984	0.79964	0.80153
Q1	0.80024	0.79952	0.79905	0.79898	0.80081
Q3	0.80166	0.80070	0.80060	0.80034	0.80224
IQR	0.00142	0.00118	0.00155	0.00136	0.00144

AL 2018-2019 SPRING WEEK



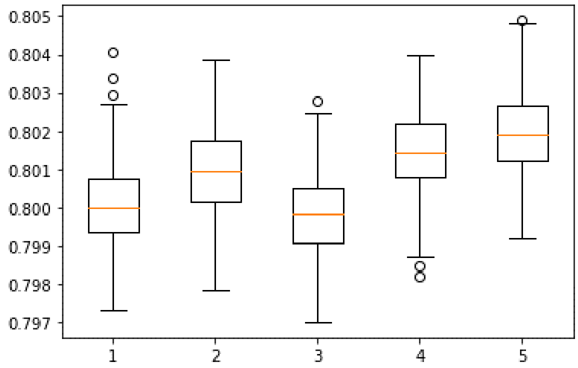
	1	2	3	4	5
Objective	110	116	113	109	113
Median	0.79912	0.80041	0.80141	0.79947	0.80142
Q1	0.79826	0.79971	0.80067	0.79873	0.80063
Q3	0.79989	0.80091	0.80221	0.80014	0.80222
IQR	0.00163	0.00120	0.00155	0.00141	0.00158

AL 2018-2019 SUMMER WEEK



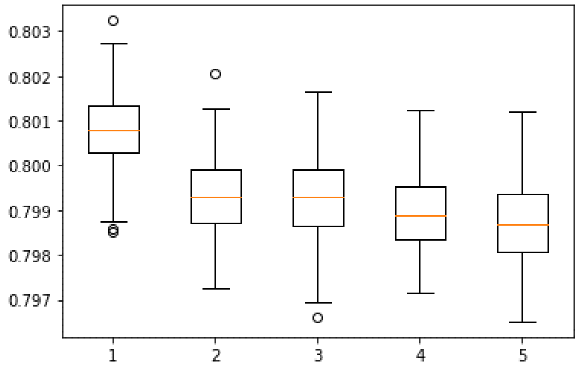
	1	2	3	4	5
Objective	114	114	111	111	115
Median	0.80114	0.79902	0.80036	0.79958	0.80112
Q1	0.80035	0.79832	0.79980	0.79886	0.80055
Q3	0.80181	0.79970	0.80100	0.80030	0.80176
IQR	0.00147	0.00138	0.00119	0.00144	0.00120

AL 2018-2019 AUTUMN WEEK



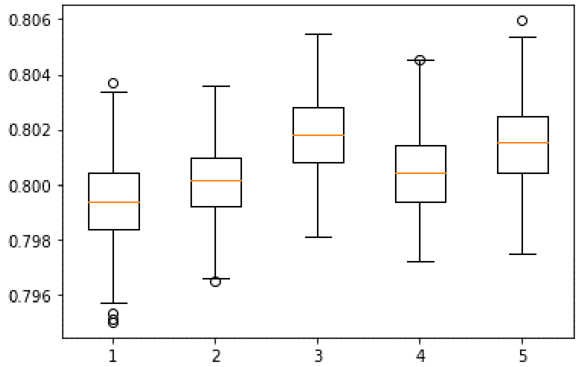
	1	2	3	4	5
Objective	115	115	114	116	114
Median	0.80001	0.80095	0.79983	0.80144	0.80191
Q1	0.79938	0.80015	0.79908	0.80078	0.80125
Q3	0.80077	0.80176	0.80052	0.80218	0.80268
IQR	0.00139	0.00161	0.00144	0.00141	0.00144

AL 2020 WEEK



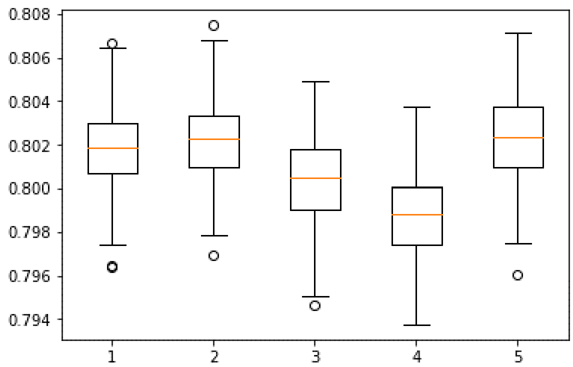
	1	2	3	4	5
Objective	125	123	124	124	124
Median	0.80079	0.79931	0.79931	0.79890	0.79867
Q1	0.80027	0.79872	0.79864	0.79837	0.79809
Q3	0.80135	0.79990	0.79990	0.79955	0.79936
IQR	0.00108	0.00118	0.00126	0.00118	0.00127

AL 2020 WINTER WEEK



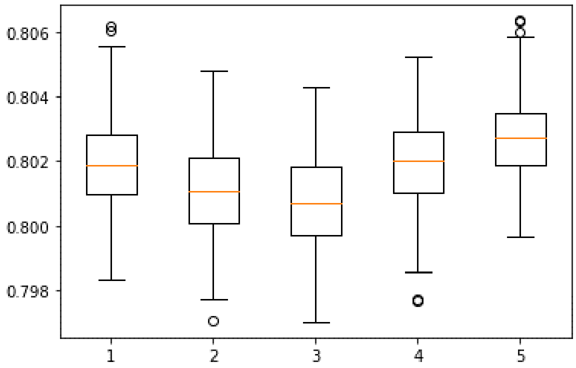
	1	2	3	4	5
Objective	92	93	91	89	93
Median	0.79937	0.80016	0.80180	0.80045	0.80153
Q1	0.79841	0.79922	0.80084	0.79935	0.80046
Q3	0.80042	0.80099	0.80279	0.80141	0.80245
IQR	0.00201	0.00178	0.00195	0.00206	0.00199

AL 2020 SPRING WEEK



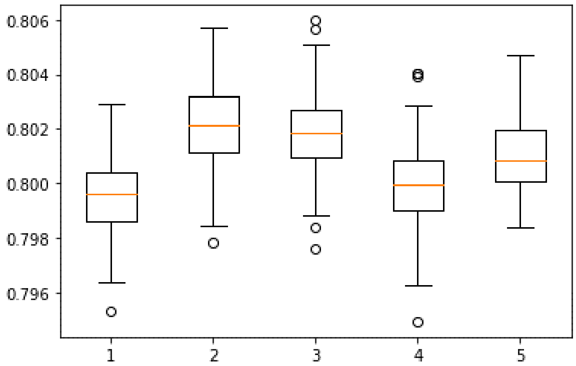
	1	2	3	4	5
Objective	89	86	89	98	89
Median	0.80187	0.80227	0.80048	0.79883	0.80236
Q1	0.80068	0.80098	0.79905	0.79742	0.80095
Q3	0.80299	0.80333	0.80184	0.80007	0.80377
IQR	0.00232	0.00235	0.00279	0.00265	0.00281

AL 2020 SUMMER WEEK



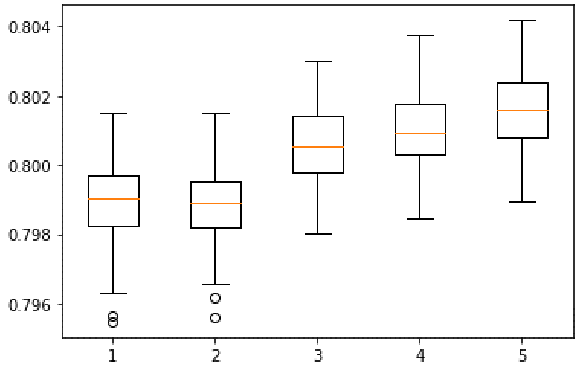
	1	2	3	4	5
Objective	98	96	94	96	94
Median	0.80187	0.80109	0.80069	0.80201	0.80272
Q1	0.80100	0.80009	0.79971	0.80104	0.80191
Q3	0.80283	0.80209	0.80182	0.80290	0.80349
IQR	0.00183	0.00200	0.00211	0.00186	0.00159

AL 2020 AUTUMN WEEK



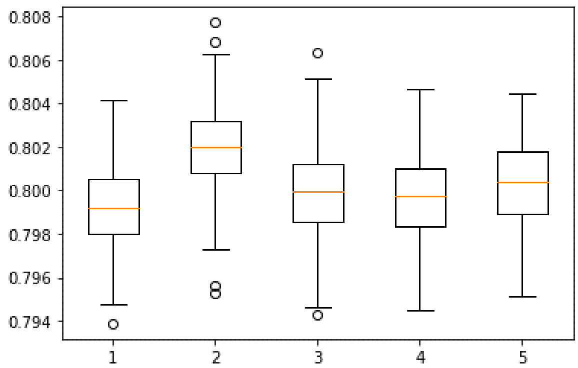
	1	2	3	4	5
Objective	96	99	102	96	100
Median	0.79963	0.80213	0.80184	0.79996	0.80085
Q1	0.79865	0.80111	0.80096	0.79900	0.80006
Q3	0.80041	0.80319	0.80266	0.80086	0.80198
IQR	0.00176	0.00208	0.00170	0.00186	0.00192

AL 2018- 2019 WEEKEND



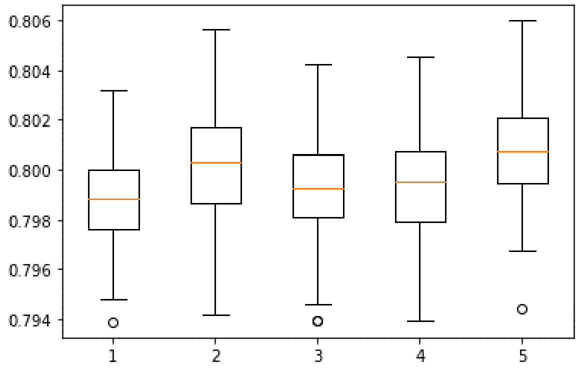
	1	2	3	4	5
Objective	112	114	115	115	119
Median	0.79905	0.79892	0.80052	0.80095	0.80162
Q1	0.79825	0.79820	0.79981	0.80034	0.80080
Q3	0.79971	0.79953	0.80140	0.80176	0.80240
IQR	0.00147	0.00133	0.00159	0.00142	0.00160

AL 2018-2019 WINTER WEEKEND



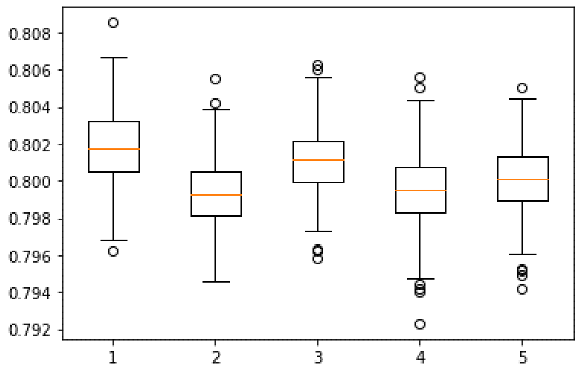
	1	2	3	4	5
Objective	78	78	81	78	80
Median	0.79921	0.80202	0.79995	0.79976	0.80035
Q1	0.79804	0.80077	0.79856	0.79839	0.79891
Q3	0.80053	0.80313	0.80123	0.80100	0.80181
IQR	0.00250	0.00236	0.00267	0.00260	0.00290

AL 2018-2019 SPRING WEEKEND



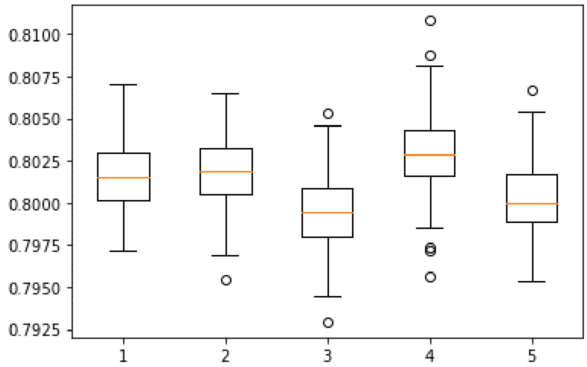
	1	2	3	4	5
Objective	88	85	88	83	84
Median	0.79883	0.80032	0.79924	0.79952	0.80074
Q1	0.79763	0.79870	0.79813	0.79791	0.79943
Q3	0.79997	0.80172	0.80062	0.80076	0.80206
IQR	0.00235	0.00302	0.00250	0.00285	0.00263

AL 2018-2019 SUMMER WEEKEND



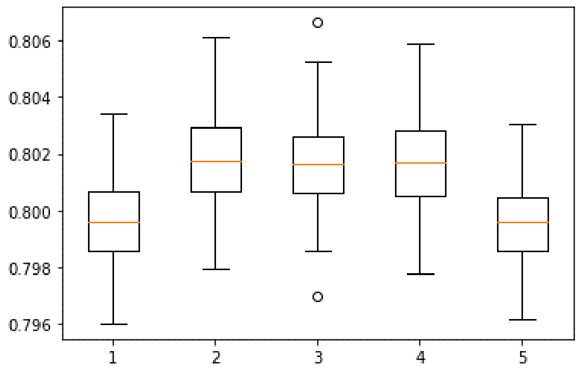
	1	2	3	4	5
Objective	81	80	81	78	81
Median	0.80170	0.79926	0.80117	0.79952	0.80014
Q1	0.80048	0.79817	0.79993	0.79829	0.79900
Q3	0.80323	0.80052	0.80218	0.80079	0.80131
IQR	0.00275	0.00235	0.00225	0.00249	0.00231

AL 2018-2019 AUTUMN WEEKEND



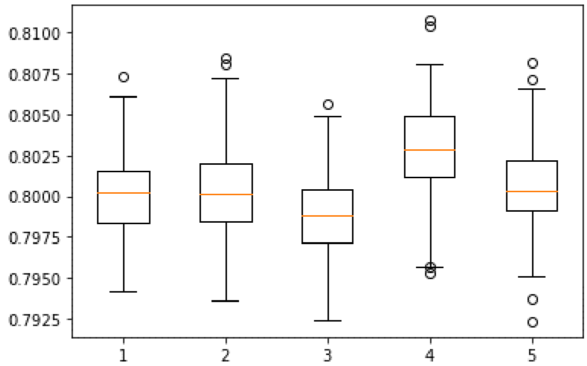
	1	2	3	4	5
Objective	74	76	75	74	76
Median	0.80153	0.80188	0.79942	0.80292	0.80002
Q1	0.80021	0.80052	0.79802	0.80157	0.79894
Q3	0.80298	0.80326	0.80083	0.80433	0.80166
IQR	0.00277	0.00274	0.00281	0.00277	0.00272

AL 2020 WEEKEND



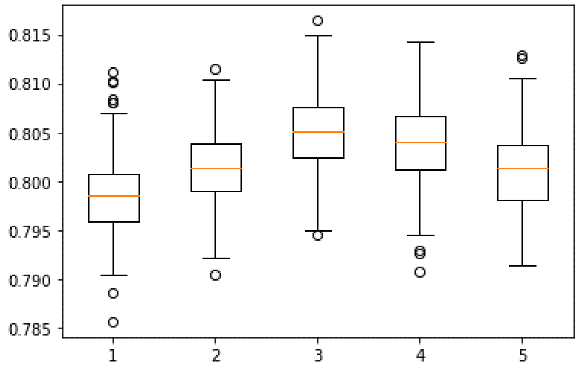
	1	2	3	4	5
Objective	92	92	92	89	93
Median	0.79959	0.80178	0.80166	0.80169	0.79962
Q1	0.79859	0.80071	0.80063	0.80055	0.79859
Q3	0.80068	0.80293	0.80262	0.80282	0.80045
IQR	0.00210	0.00222	0.00199	0.00227	0.00186

AL 2020 WINTER WEEKEND



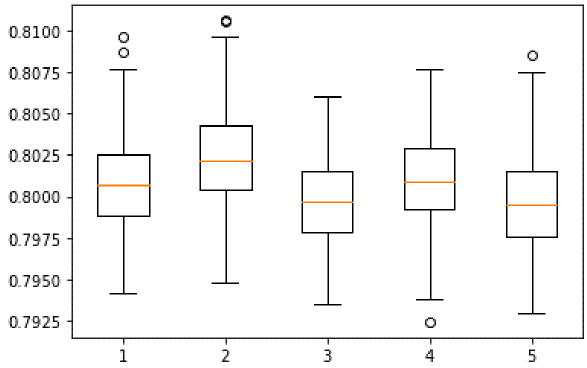
	1	2	3	4	5
Objective	63	60	63	63	63
Median	0.80020	0.80017	0.79889	0.80284	0.80036
Q1	0.79835	0.79847	0.79714	0.80120	0.79911
Q3	0.80157	0.80199	0.80045	0.80486	0.80219
IQR	0.00322	0.00353	0.00331	0.00366	0.00307

AL 2020 SPRING WEEKEND



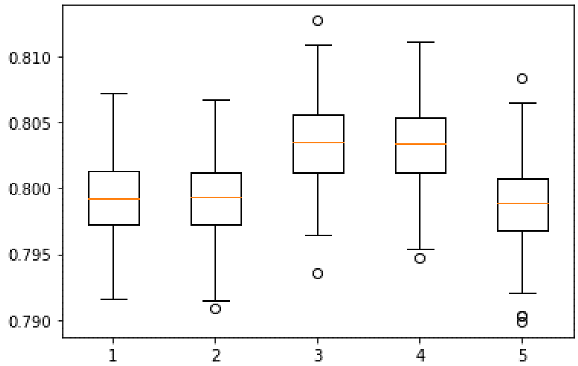
	1	2	3	4	5
Objective	48	49	50	48	50
Median	0.79859	0.80144	0.80506	0.80400	0.80141
Q1	0.79597	0.79910	0.80247	0.80125	0.79807
Q3	0.80068	0.80379	0.80756	0.80668	0.80368
IQR	0.00471	0.00469	0.00509	0.00543	0.00561

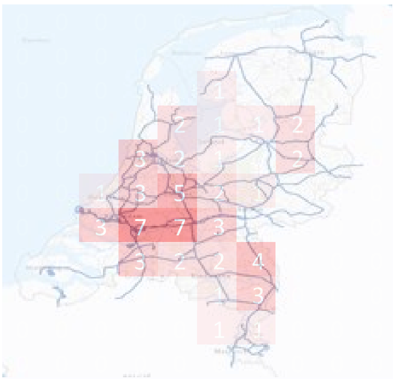
AL 2020 SUMMER WEEKEND



	1	2	3	4	5
Objective	62	64	64	60	63
Median	0.80072	0.80221	0.79968	0.80085	0.79947
Q1	0.79886	0.80042	0.79786	0.79922	0.79759
Q3	0.80256	0.80426	0.80153	0.80294	0.80156
IQR	0.00370	0.00384	0.00367	0.00373	0.00397

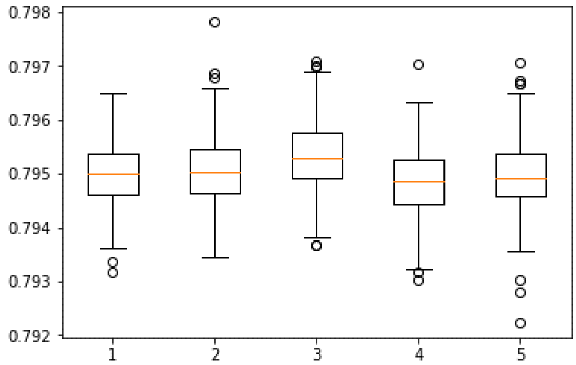
AL 2020 AUTUMN WEEKEND





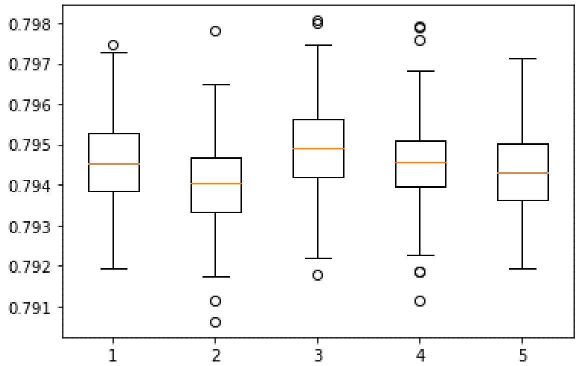
	1	2	3	4	5
Objective	60	64	65	60	61
Median	0.79924	0.79931	0.80352	0.80346	0.79890
Q1	0.79733	0.79729	0.80127	0.80121	0.79686
Q3	0.80126	0.80120	0.80554	0.80536	0.80077
IQR	0.00394	0.00391	0.00427	0.00415	0.00391

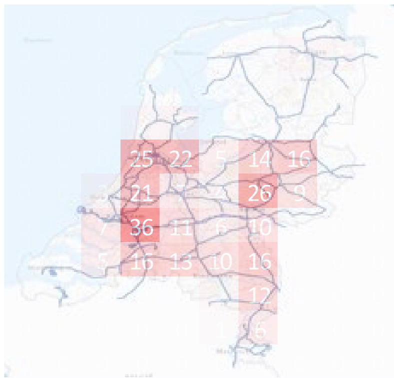
ICB 2018-2019 WEEK



	1	2	3	4	5
Objective	425	424	426	425	425
Median	0.79501	0.79502	0.79529	0.79486	0.79492
Q1	0.79461	0.79464	0.79493	0.79444	0.79457
Q3	0.79538	0.79545	0.79575	0.79526	0.79536
IQR	0.00077	0.00081	0.00081	0.00082	0.00079

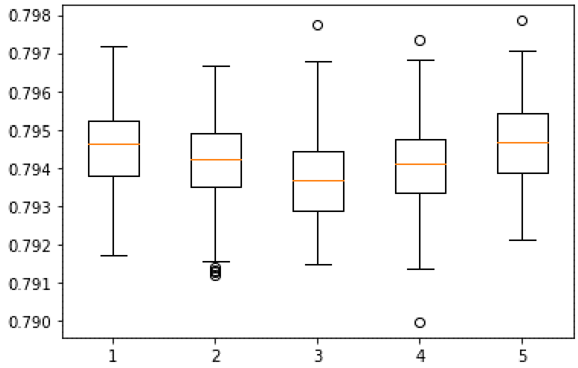
ICB 2018-2019 WINTER WEEK





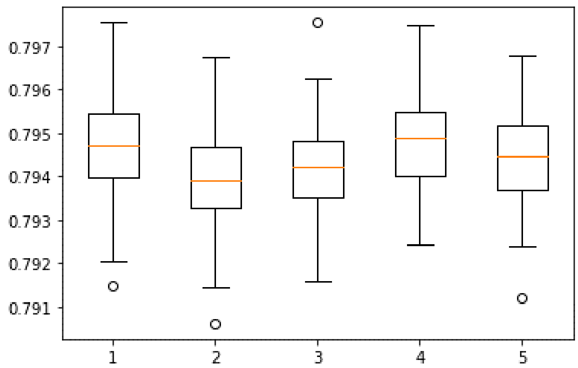
	1	2	3	4	5
Objective	314	313	315	313	318
Median	0.79455	0.79405	0.79492	0.79457	0.79431
Q1	0.79386	0.79332	0.79421	0.79396	0.79362
Q3	0.79527	0.79468	0.79562	0.79511	0.79503
IQR	0.00141	0.00135	0.00141	0.00115	0.00140

ICB 2018-2019 SPRING WEEK



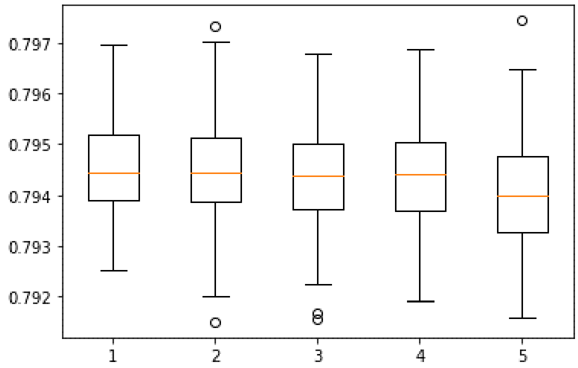
	1	2	3	4	5
Objective	330	329	332	334	330
Median	0.79464	0.79425	0.79368	0.79411	0.79466
Q1	0.79381	0.79350	0.79286	0.79335	0.79389
Q3	0.79523	0.79489	0.79443	0.79475	0.79541
IQR	0.00142	0.00139	0.00158	0.00140	0.00152

ICB 2018-2019 SUMMER WEEK



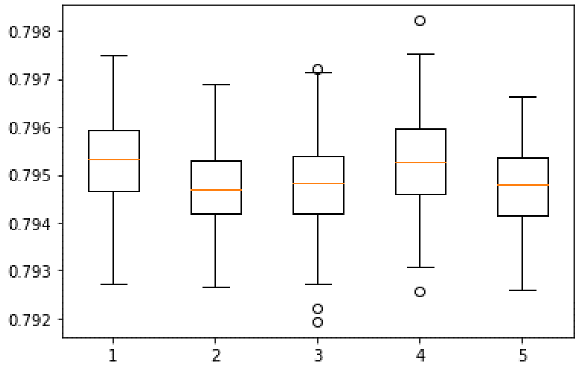
	1	2	3	4	5
Objective	335	335	336	340	336
Median	0.79470	0.79390	0.79422	0.79490	0.79447
Q1	0.79399	0.79327	0.79352	0.79402	0.79369
Q3	0.79544	0.79466	0.79482	0.79547	0.79518
IQR	0.00145	0.00139	0.00130	0.00145	0.00149

ICB 2018-2019 AUTUMN WEEK



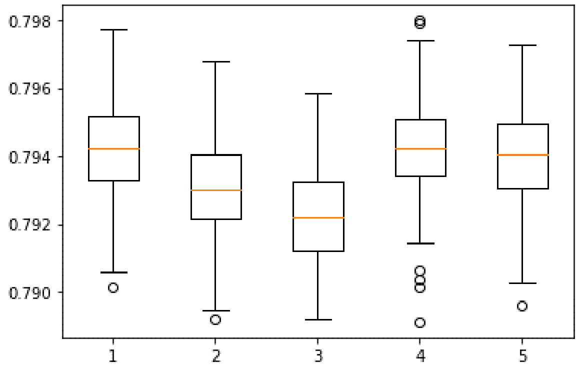
	1	2	3	4	5
Objective	344	339	341	339	341
Median	0.79444	0.79444	0.79437	0.79441	0.79398
Q1	0.79391	0.79386	0.79371	0.79368	0.79328
Q3	0.79518	0.79513	0.79499	0.79503	0.79477
IQR	0.00127	0.00127	0.00128	0.00136	0.00149

ICB 2020 WEEK



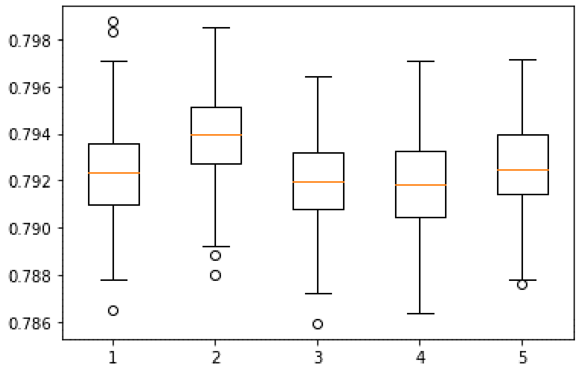
	1	2	3	4	5
Objective	378	379	373	377	374
Median	0.79535	0.79471	0.79482	0.79528	0.79478
Q1	0.79466	0.79420	0.79419	0.79461	0.79417
Q3	0.79594	0.79529	0.79539	0.79596	0.79536
IQR	0.00128	0.00110	0.00120	0.00135	0.00120

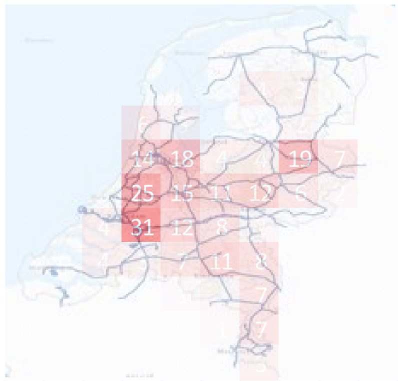
ICB 2020 WINTER WEEK



	1	2	3	4	5
Objective	283	284	280	280	279
Median	0.79425	0.79300	0.79221	0.79422	0.79406
Q1	0.79329	0.79215	0.79123	0.79342	0.79308
Q3	0.79520	0.79405	0.79325	0.79507	0.79494
IQR	0.00191	0.00190	0.00202	0.00166	0.00186

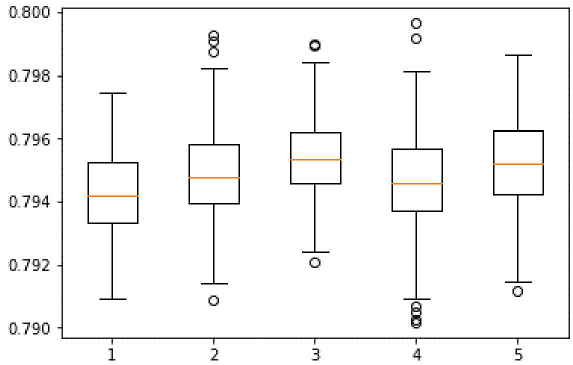
ICB 2020 SPRING WEEK

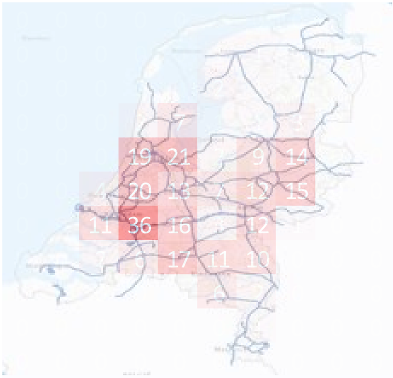




	1	2	3	4	5
Objective	266	267	266	265	266
Median	0.79237	0.79394	0.79195	0.79187	0.79251
Q1	0.79102	0.79273	0.79082	0.79047	0.79147
Q3	0.79360	0.79512	0.79318	0.79324	0.79397
IQR	0.00258	0.00239	0.00236	0.00276	0.00250

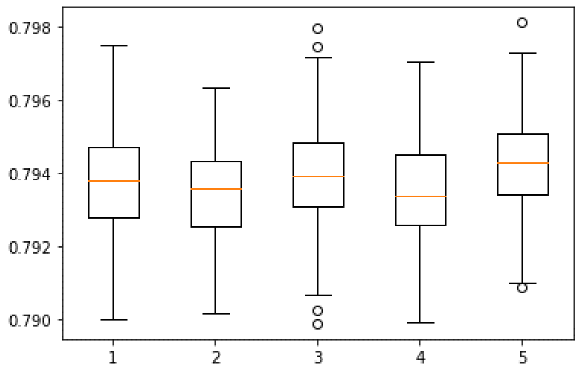
ICB 2020 SUMMER WEEK





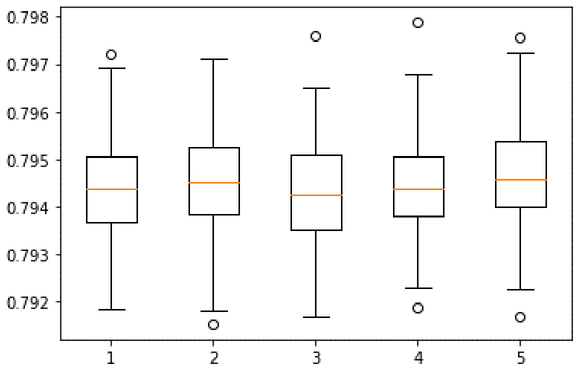
	1	2	3	4	5
Objective	294	289	289	292	288
Median	0.79419	0.79478	0.79536	0.79458	0.79519
Q1	0.79332	0.79395	0.79457	0.79372	0.79425
Q3	0.79522	0.79582	0.79621	0.79566	0.79623
IQR	0.00190	0.00187	0.00164	0.00194	0.00198

ICB 2020 AUTUMN WEEK



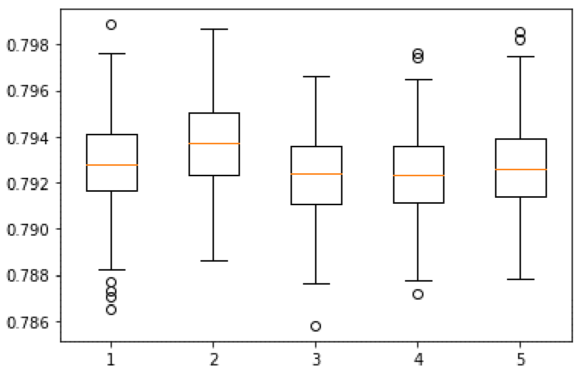
	1	2	3	4	5
Objective	296	299	297	299	303
Median	0.79378	0.79358	0.79390	0.79339	0.79428
Q1	0.79277	0.79253	0.79309	0.79259	0.79340
Q3	0.79472	0.79432	0.79481	0.79447	0.79508
IQR	0.00194	0.00179	0.00173	0.00188	0.00168

ICB 2018-2019 WEEKEND



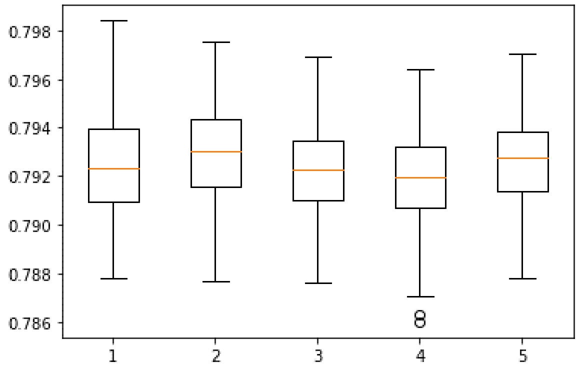
	1	2	3	4	5
Objective	342	343	345	344	345
Median	0.79437	0.79450	0.79427	0.79439	0.79457
Q1	0.79368	0.79383	0.79353	0.79380	0.79398
Q3	0.79505	0.79523	0.79508	0.79505	0.79538
IQR	0.00137	0.00141	0.00155	0.00126	0.00139

ICB 2018-2019 WINTER WEEKEND



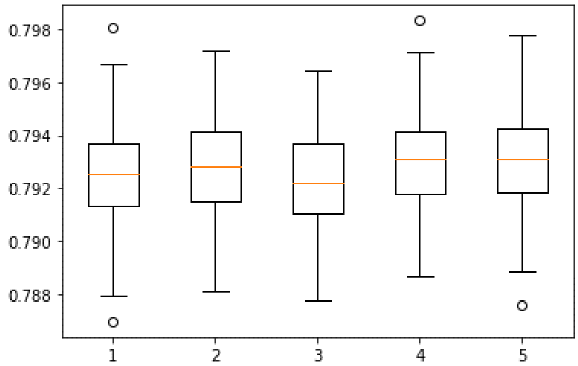
	1	2	3	4	5
Objective	241	243	239	240	240
Median	0.79281	0.79375	0.79239	0.79232	0.79261
Q1	0.79165	0.79233	0.79112	0.79114	0.79142
Q3	0.79411	0.79506	0.79360	0.79362	0.79390
IQR	0.00247	0.00273	0.00249	0.00248	0.00248

ICB 2018-2019 SPRING WEEKEND



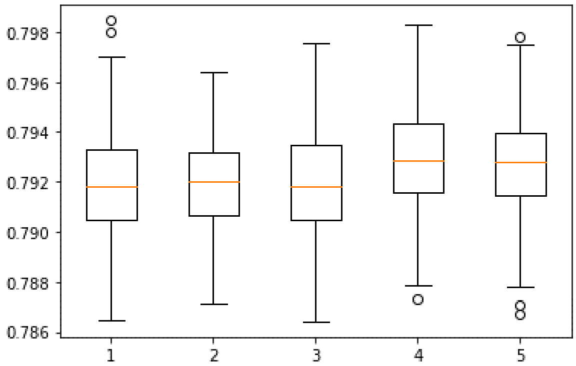
	1	2	3	4	5
Objective	255	254	254	254	256
Median	0.79230	0.79304	0.79226	0.79197	0.79278
Q1	0.79097	0.79160	0.79100	0.79071	0.79139
Q3	0.79395	0.79437	0.79348	0.79319	0.79385
IQR	0.00298	0.00277	0.00248	0.00249	0.00246

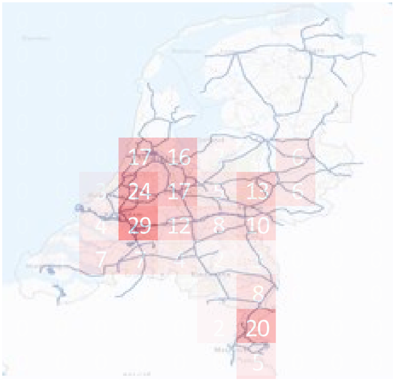
ICB 2018-2019 SUMMER WEEKEND



	1	2	3	4	5
Objective	245	242	245	244	246
Median	0.79253	0.79280	0.79221	0.79309	0.79309
Q1	0.79133	0.79152	0.79106	0.79181	0.79186
Q3	0.79368	0.79413	0.79366	0.79416	0.79425
IQR	0.00235	0.00262	0.00260	0.00235	0.00239

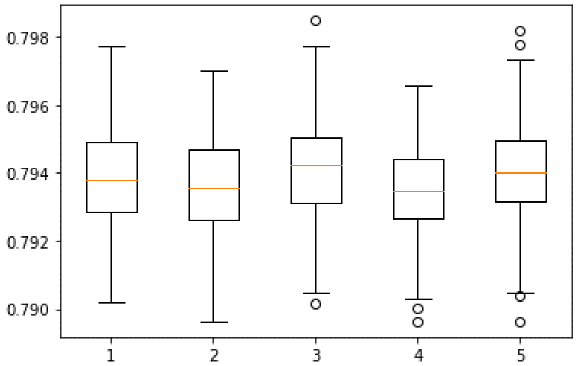
ICB 2018-2019 AUTUMN WEEKEND





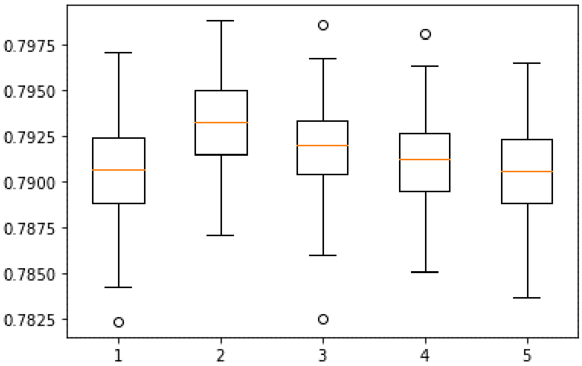
	1	2	3	4	5
Objective	238	235	233	234	234
Median	0.79186	0.79203	0.79186	0.79285	0.79279
Q1	0.79051	0.79066	0.79049	0.79160	0.79144
Q3	0.79329	0.79316	0.79347	0.79432	0.79394
IQR	0.00278	0.00250	0.00297	0.00272	0.00250

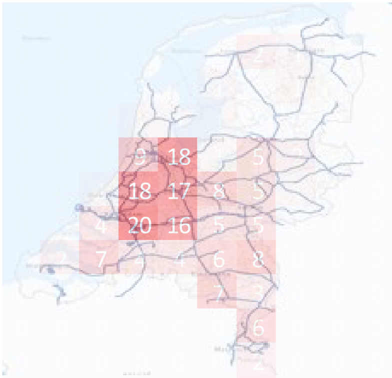
ICB 2020 WEEKEND



	1	2	3	4	5
Objective	279	281	281	282	279
Median	0.79378	0.79355	0.79423	0.79349	0.79402
Q1	0.79285	0.79264	0.79315	0.79268	0.79316
Q3	0.79489	0.79470	0.79503	0.79441	0.79498
IQR	0.00204	0.00206	0.00188	0.00174	0.00182

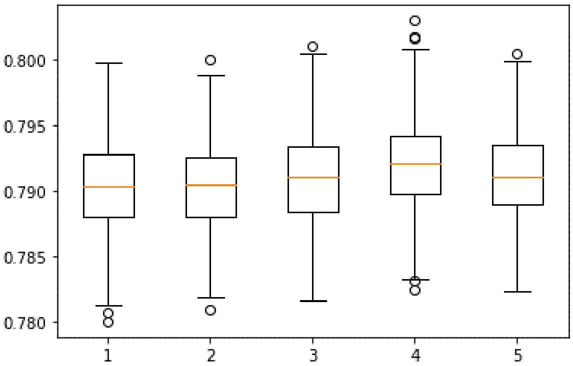
ICB 2020 WINTER WEEKEND





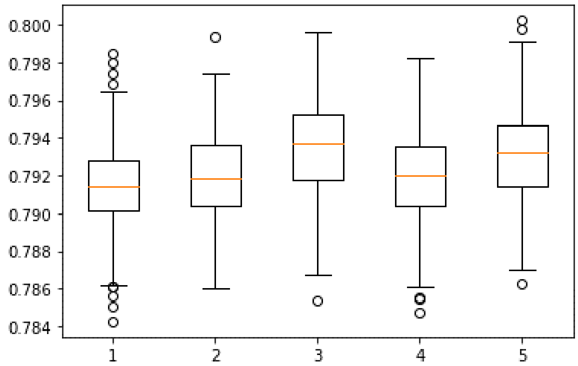
	1	2	3	4	5
Objective	194	197	192	196	194
Median	0.79070	0.79329	0.79197	0.79125	0.79056
Q1	0.78885	0.79149	0.79039	0.78947	0.78889
Q3	0.79241	0.79498	0.79335	0.79263	0.79232
IQR	0.00356	0.00349	0.00295	0.00316	0.00344

ICB 2020 SPRING WEEKEND



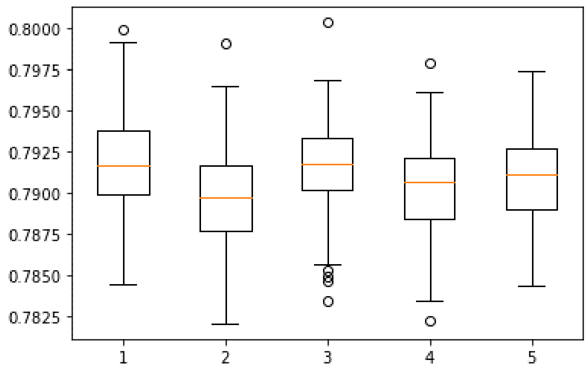
	1	2	3	4	5
Objective	153	153	155	155	151
Median	0.79041	0.79051	0.79101	0.79209	0.79111
Q1	0.78807	0.78811	0.78834	0.78981	0.78897
Q3	0.79282	0.79255	0.79335	0.79425	0.79350
IQR	0.00475	0.00444	0.00502	0.00444	0.00454

ICB 2020 SUMMER WEEKEND



	1	2	3	4	5
Objective	195	189	192	190	190
Median	0.79147	0.79181	0.79367	0.79200	0.79324
Q1	0.79013	0.79037	0.79178	0.79036	0.79146
Q3	0.79279	0.79362	0.79523	0.79352	0.79466
IQR	0.00266	0.00326	0.00345	0.00315	0.00320

ICB 2020 AUTUMN WEEKEND



	1	2	3	4	5
Objective	189	188	192	190	190
Median	0.79175	0.78973	0.79184	0.79066	0.79113
Q1	0.78994	0.78769	0.79024	0.78845	0.78904
Q3	0.79383	0.79169	0.79337	0.79215	0.79275
IQR	0.00389	0.00400	0.00312	0.00370	0.00371