Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Erik Mohn

Ambulance Allocation Optimization and Simulation with Incident Urgency and Demand Prediction

Master's thesis in Informatics
Supervisor: Ole Jakob Mengshoel
June 2023

NTNU
Norwegian University of
Science and Technology

Erik Mohn

Ambulance Allocation Optimization and Simulation with Incident Urgency and Demand Prediction

Master's thesis in Informatics
Supervisor: Ole Jakob Mengshoel
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

ABSTRACT

Rapid medical assistance can mean the difference between life and death in many cases. For this reason, it is essential that emergency response vehicles reach the scene of an incident as quickly as possible to provide patients with timely treatment. Unfortunately, recent years have seen a rise in the average response time to incidents in Oslo, which could have severe consequences. As the Emergency Medical Communication Centre in Oslo and Akershus faces limited resources, reducing ambulance response time has become a crucial challenge.

This thesis aims to address this challenge by researching and developing tools to improve the management and utilization of existing resources. The optimization involves strategically allocating ambulances to specific ambulance base stations so that they are more likely to be close to upcoming incidents. Simulations using real-world historic incidents in the area of Oslo and Akershus can evaluate different allocations through the resulting response times. The simulations are used to heuristically find good allocations using artificial intelligence methods, specifically, an evolutionary approach that takes advantage of the genetic algorithm's ability to search in a great number of possible allocations.

The primary contribution of this thesis is the adaptation of the optimization method to consider the different urgencies of incidents. Correctly classifying the acute incidents is important since their response time is critical and resources should prioritize those incidents over less critical ones. Additionally, contributions include the improvement of both the evolutionary optimization algorithm and the accuracy of the simulation. By addressing the important challenge of reducing ambulance response time, this research has the potential to enhance emergency medical services and improve patient outcomes.

SAMMENDRAG

Mange hendelser som krever assistanse fra medisinsk personell er tidskritiske. Det er derfor nødvendig at utrykningskjøretøy når frem til hendelsesstedet så raskt som mulig for å gi pasienten den behandlingen de trenger. De siste årene har den gjennomsnittlige responstiden ved akutte hendelser i Oslo økt, noe som kan ha store konsekvenser. Med begrensede ressurser har Akuttmedisinsk kommunikasjonssentral i Oslo og Akershus en viktig utfordring med det å redusere ambulansens responstid.

Denne masteroppgaven vil forsøke å bistå i denne utfordringen, ved å undersøke og utvikle verktøy som kan forbedre utnyttelsen av eksisterende ressurser. Optimaliseringen gjøres ved å strategisk allokere ambulanser til spesifikke ambulansebasestasjoner slik at de har større sannsynlighet for å være i nærheten av kommende hendelser. Simuleringer ved hjelp av et sett med historiske hendelser i Oslo og Akershus kan evaluere ulike allokeringer gjennom den resulterende gjennomsnittlige responstiden fra simuleringen, som brukes til å heuristisk finne gode allokeringer ved hjelp av metoder innenfor kunstig intelligens. Spesielt brukes en evolusjonær tilnærming som utnytter den genetiske algoritmens evne til å søke i det store antallet mulige allokeringer.

Hovedbidraget til denne oppgaven er tilpasningen av optimaliseringsmetoden til å håndtere alvorsgraden av hendelser. Riktig klassifisering av akutte hendelser er viktig, da responstiden er kritisk, og ressurser bør prioriteres til disse hendelsene fremfor mindre kritiske hendelser. Videre inkluderer bidragene forbedring av både den evolusjonære optimaliseringsalgoritmen og simuleringens nøyaktighet.

PREFACE

This thesis was conducted at the Norwegian University of Science and Technology (NTNU) by Erik Mohn to finalize his master's degree in Informatics at the Department of Computer Science. The project is a continuous project in collaboration with Oslo University Hospital (OUS) that has spanned across two previous iterations involving master students. Several parts of this thesis builds on their work and research, and I thank Magnus Eide Schjølberg and Nicklas Imanuel Paus Bekkevold for their help and advice with the introduction and setup of the project. I also thank OUS for taking time out of their day to give insightful information about the Emergency Medical Communication Center and their operations.

I also want to thank Ole Jakob Mengshoel for his supervisor role with topic ideas, research advice, and support throughout the project period.

Erik Mohn
Trondheim, June 7, 2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

- **ANN** Artificial Neural Network
- **DES** Discrete Event Simulation
- **EMCC** Emergency Medical Communication Center
- **EMS** Emergency Medical Service
- **GA** Genetic Algorithm
- **IMGA** Island Model Genetic Algorithm
- **MEXCLP** Maximum Expected Coverage Location Problem
- **MSE** Mean Squared Error
- **NTNU** Norwegian University of Science and Technology
- **OSM** Open Street Map
- **OUS** Oslo University Hospital
- **UTM** Universal Transverse Mercator

CHAPTER

ONE

INTRODUCTION

This chapter gives an introduction to the motivation of the project, the general problem, and how the research and contributions aims to improve the current solution. Specifically, the research goals and research method of this thesis will be elaborated upon.

1.1 Motivation

Different incidents require varying levels of urgency in medical response, and in this thesis the incidents are categorized into three levels: acute, urgent, and regular. For acute incidents, the national goal in Norway is that ambulances arrive within 12 minutes in 90 percent of the incidents in urban areas and within 25 minutes in rural areas. Unfortunately, this goal has not been met by any of the regions in Norway, and emergency medical response time has been getting worse in recent years (Helsedirektoratet 2022). This thesis focuses on the notion of response time, which is defined as the time from an emergency call being received to when the ambulance arrives at the incident location. A typical emergency response timeline of relevant events can be seen in Figure 1.1.1.

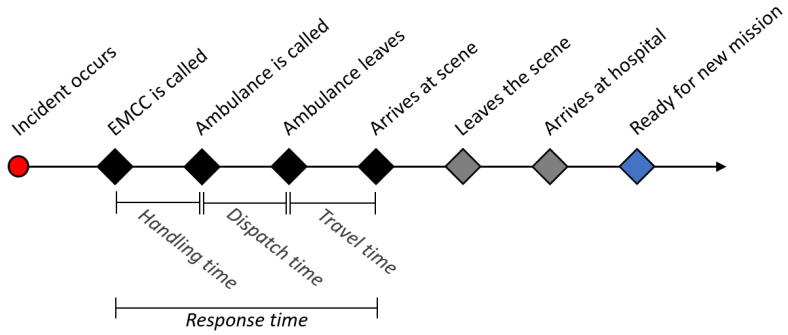


Figure 1.1.1: Emergency response timeline with the response time duration marked in black nodes.

Acquiring more resources like ambulances, base station facilities, and employees is not only expensive, but a long and potentially difficult political process. The Norwegian Board of Health Supervision reported that the service of the Emergency Medical Communication Centre (EMCC) in Oslo was not acceptable, partly because of staffing and workload issues (Helsetilsynet 2022). This makes the EMCC interested in exploring other options to improve their system. An optimized allocation of ambulances can help to utilize the resources more efficiently, which can lower the workload for both the EMCC and the ambulance employees. Reducing response time for acute incidents ultimately saves lives and reduces the probability of lasting injuries from an incident, which evidently is the end goal.

Another reason which the supervision report claimed was a factor in lowering the quality of service of the EMCC, was the inaccuracy in urgency level that the operators assign to an incident when receiving an emergency call. The operators use a triage system to categorize incidents by severity level, but many non-acute incidents are reportedly classified as acute as a safety measure. As a consequence, unnecessary dispatching of ambulances occurs, which results in the waste of valuable resources that could have been more useful in upcoming incidents. The ripple effect of incorrectly assigning urgency levels can be difficult to observe, so researching this phenomenon could give valuable insight into the benefits of improving the triage system.

Finally, an improvement in emergency vehicle response time is not only of interest to the city of Oslo, it is a common need among other parts of the world. Other regions may have different challenges, but any research or contribution is valuable.

1.2 Research Goals

The overarching goal of this thesis is to investigate potential solutions for improving the service provided by emergency response vehicles, with a specific focus on reducing response times, particularly for incidents with acute urgency. The thesis builds upon several theses done on the same domain, outlined in Section 1.3. Among other tools, a simulation of the real system and an optimization model for ambulance allocations were developed in these theses, which are utilized and expanded upon in this work. While the focus is on achieving tangible results, the goal is also to expand knowledge in this area and explore the applicability of bio-inspired artificial methods. To achieve this, the thesis will concentrate on four research goals:

- **Goal 1:** The first goal aims to improve simulation realism to accurately reflect real-world conditions, including travel time calculation, EMCC dispatch behaviour, and incident types. By enhancing the simulation, the optimization solutions and research outcomes become more relevant.

- **Goal 2:** The second goal is to explore different dispatch strategies that could improve the average response time. Choosing which ambulance to dispatch in various situations can be challenging and has significant impacts on response times. One especially interesting strategy that will be examined

is implementing demand prediction to guide the EMCC in selecting the best ambulance to dispatch for each incident.

- **Goal 3:** The third goal is to study the urgency aspect of incidents. This will mainly involve observing the impact on response times from reducing the number of non-acute incidents assigned as acute as a precautionary measure. It will also be interesting to examine this effect under different dispatch strategies.

- **Goal 4:** The fourth goal aims to reduce ambulance response times to incidents by optimizing the allocation of ambulances to base stations. This will involve enhancing the algorithm used to optimize the allocation of ambulances.

1.3 Research Method

The focus of this thesis is to propose and experiment with potential improvements to the simulated emergency medical service (EMS) system itself, as well as the allocation optimization. To ensure the relevance of the proposed improvements, related works will be used as a source of inspiration and validation. The research domain encompasses both the simulation of the EMS system and optimization through bio-inspired artificial intelligence methods.

The research will be done by improving on and using work done in three previous theses. Hermansen (2021) focuses its research on predicting future demand based on historic data of incidents. This data, received from OUS, contains incidents from 2015 to 2019 in the area of Oslo and Akershus. The dataset was also used in the other theses, where Van De Weijer and Owren (2022) continues to focus on demand prediction, while Bekkevold and Schjølberg (2022) focuses their research on allocation optimization using a simulation to evaluate the EMS system. Both the same dataset, and the optimization and simulation system developed in Bekkevold and Schjølberg (2022), will be used for research in this thesis.

1.4 Thesis Structure

This thesis is structured to provide a comprehensive understanding of the proposed solutions and research goals. Chapter 2 will provide the necessary background information on different aspects of the problem. Chapter 3 will focus on the theory of relevant methods that are used. Chapter 4 will focus on the related work done in the literature. The architecture of the proposed solutions and the results of the research goals are studied in Chapter 5, while Chapter 6 concludes with a summary of the thesis, including contributions, limitations and suggestions for further work.

CHAPTER

TWO

BACKGROUND

This chapter will give further knowledge required to understand the problem domain and the solution space. First, an explanation of the EMS of Oslo and Akershus will be presented. The service is the basis for the simulation, and is an integral part of the thesis. Second, the dataset will be explained and analysed. Lastly, the possible solutions for the ambulance allocation problem will be explained.

It is important to note that the dataset used in this research is from 2015 to 2019, which means that this thesis will only present and use the elements of the EMS system during that period. The changes that have been made since the recording of the data have not been taken into account, in order to ensure an accurate simulation and evaluation of the system with potential improvements.

2.1 EMS

The EMS of Oslo and Akershus is the largest EMS provider in Norway, serving a population of 1.5 million people. The EMS of Oslo and Akershus includes various departments and entities, such as the ambulance department and the EMCC, which work together to provide medical assistance to the population. According to a report from 2014, the EMCC received up to 500,000 calls, which is expected to have increased at a similar rate to the reported increase in incidents over the past years. These calls to the EMCC result in approximately 150,000 incident operations every year (OUH 2022a) (OUH 2022b).

2.1.1 EMCC

The EMCC is responsible for taking emergency calls from the public, evaluating what resources are necessary for the incident, and dispatching ambulances or other response vehicles. The EMCC in Oslo and Akershus is located at Ullevål Hospital, and one of their offices is shown in Figure 2.1.1.

Figure 2.1.1: EMCC office environment at Ullevål Hospital (NRK 2022).

2.1.2 Ambulance Department

The ambulance department is responsible for all the ambulances and ambulance personnel required to respond to the incidents in Oslo and Akershus. They have a total of 45 ambulances that are able to respond to all types of incidents in operation during the day, where 29 of them are also operational during the night shift. The department and its ambulances are currently distributed across 15 base stations in 5 different regions.

2.1.3 Triage

When emergency calls are received by the EMCC from the public, the operators answering the call make decisions about the resources that should be deployed. The incidents can range from not requiring any assistance to needing expert medical help to arrive as soon as possible. The urgency of the incident is sorted into three different categories, usually referred to as a triage. In Oslo and Akershus, the EMCC uses a triage system with the following levels:

- **Acute (A):** Immediate dispatch and call-out. It was described to the author that ambulance employees should drop everything in their hands and rush to the ambulance. The ambulance will use sirens and lights for this type of incident.

- **Urgent (H):** Dispatching should have no delay, but no need to run. The ambulance will not use sirens and lights.

- **Regular (V):** No particular urgency. These events are split into planned and unplanned events, and are not prioritized. Planned events include missions like transporting patients between hospitals.

Assigning the appropriate level of urgency to an incident is not always straightforward for EMCC operators, as callers may be unable to provide clear information due to their state of distress or misunderstanding of the situation. As mentioned in Section 1.1, the EMCC is frequently dealing with a large number of calls which causes the operators to make decisions with a limited amount of time, leading to a lack of information being processed and a higher chance of human error. Several additional reasons for assignment inaccuracy are presented in Ivanov et al. (2021). Consequently, operators often err on the side of caution and assign a higher level of urgency than necessary, resulting in what is referred to as 'over-triage'. When the EMCC decides to dispatch an ambulance to an incident that in fact does not require immediate assistance, the ambulance might leave an area where another actual acute incident occurs. However, if the first incident was correctly assigned as a non-acute incident, the EMCC could dispatch an ambulance located further away from the incident, but in an area with lower demand or abundance of ambulances.

2.1.4 Base Stations

The ambulance department controls 45 ambulances during daytime operations and 29 ambulances during the night that are stationed across five areas within the responsibility area of the EMS of Oslo and Akershus. In total, there are 15 base stations where the ambulances and working personnel are stationed when they are inactive between incident operations. The base stations function as a place for the ambulance employees to rest between missions, but also to preserve and maintain the ambulance equipment in garages sheltered from rain and snow. A base station typically hosts two or three ambulances with associated personnel, but the most central stations have a larger capacity. An overview of the different base stations can be seen in Table 2.1.1 which includes their position in the Universal Transverse Mercator (UTM) coordinate format with zone 33.

Name	**Region**	**Easting**	**Northing**
Asker	West	244478	6641283
Bærum	West	248901	6648585
Smestad	West	259127	6652543
Ullevål	Mid	261774	6652003
Sentrum	Mid	262948	6649765
Brobekk	East	267085	6651035
Lørenskog	East	275840	6650643
Nittedal	East	270631	6663254
Aurskog-Høland	East	307577	6642937
Ullensaker	North	286455	6671754
Eidsvoll	North	287187	6692448
Nes	North	304199	6669959
Prinsdal	South	265048	6640259
Northern Follo	South	266827	6627037
Southern Follo	South	259265	6621267

Table 2.1.1: Base stations in Oslo and Akershus.

In addition to the 15 base stations, it was decided to establish a set of standby points to decrease response time in certain areas. Since building and maintaining completely new base stations is expensive, it was deemed more resource efficient to utilize these simple standby points which can easily be moved if it is discovered that they are more useful elsewhere. The dynamic standby points are typically positioned at gas stations which provide necessary personnel facilities in addition to being located close to main road junctions. The 4 additional standby points give a total of 19 stations, and are presented in Table 2.1.2.

Name	Region	Easting	Northing
Bekkestua	West	253295	6650494
Grorud	East	270248	6654139
Skedsmokorset	East	279154	6657789
Ryen	South	265439	6646945

Table 2.1.2: Standby points used in Oslo and Akershus.

2.1.5 Hospitals

Hospitals are significant locations frequently visited by ambulances, both as a destination for transporting patients and as a starting point after completing incident missions. The EMS of Oslo and Akershus is responsible for eleven hospital locations, as listed in Table 2.1.3, some of which are located with a base station.

Name	Region	Easting	Northing
Bærum Hospital	West	248901	6648585
Asker and Bærum emergency ward	West	248901	6648585
Radiumhospitalet	West	257732	6651563
Diakonhjemmet Hospital	West	260024	6652122
Rikshospitalet	West	260789	6653451
Ullevål Hospital	Mid	261774	6652003
Lovisenberg Diaconal Hospital	Mid	262348	6651667
Storgata emergency ward	Mid	262948	6649765
Aker emergency ward	East	265200	6652210
Aker Hospital	East	265200	6652210
Akershus university hospital	East	276381	6650642
Nedre Romerike emergency ward	East	278942	6652867
Ski Hospital	South	266359	6628267
Follo emergency ward	South	266359	6628267

Table 2.1.3: Hospitals in Oslo and Akershus.

All the relevant infrastructure that the EMCC and the ambulances deals with are presented in Figure 2.1.2, which shows the base stations and the hospitals within the EMS of Oslo and Akershus. Generally, the infrastructure is more concentrated around highly populated areas, as one might expect.

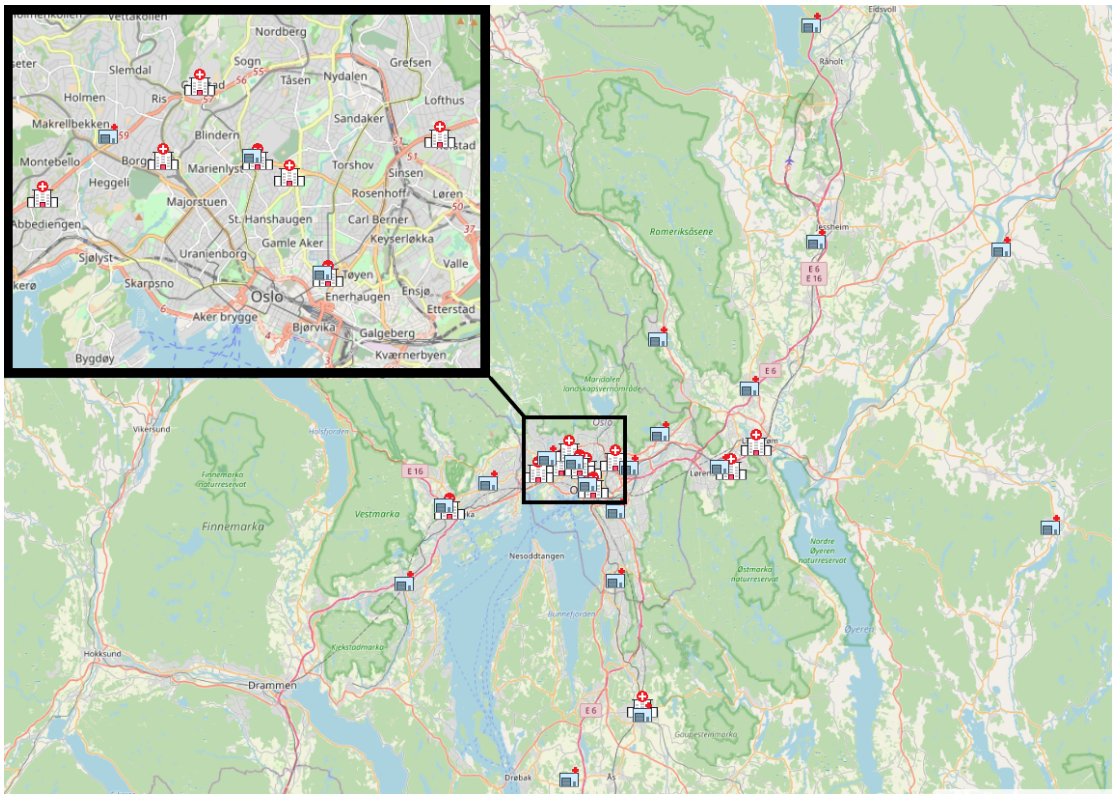


Figure 2.1.2: Map of EMS infrastructure in Oslo and Akershus.

2.2 Datasets

The dataset, referred to as *incidents-original* in this thesis, was obtained from OUS in 2021 and comprises 754 811 incidents mainly from 2015-2018, with some incidents from 2019. The incident locations were anonymized by OUS using a 1km x 1km grid for research purposes. The dataset includes several columns, or features, with the most significant ones presented in Table 2.2.1.

Feature	Description
Urgency	The urgency level that the EMCC assigned the incident
Resource type	The type of vehicle that was dispatched
Call time	Time when the EMCC was called about the incident
Notified time	Time when the ambulance was notified of the incident
Dispatch time	Time when the ambulance left its location
Arrival time	Time when the ambulance arrived at the incident
Departure time	Time when the ambulance left the scene of the incident
Hospital time	Time when the ambulance arrived at the hospital
Available time	Time when the ambulance became available again
X coordinate	UTM-33 easting value for grid coordinate of incident
Y coordinate	UTM-33 northing value for grid coordinate of incident

Table 2.2.1: Incident dataset features with description. All time features are timestamps with both the date and the time.

The dataset has been processed similarly to what has already been done in previous theses, but with minor changes. Optional size reducing steps of processing include:

- **Filter years:** Only keep incidents that happened in years that are complete (2015-2018).
- **Filter regions:** Incidents that are not within the response area of Oslo and Akershus are removed.
- **Filter erroneous timestamps:** Incidents that have obvious timestamp errors are removed.
- **Filter dispatch types:** Do not include incidents that were responded to by a special unit.
- **Filter urgency:** Only include incidents that are either acute or urgent.
- **Aggregate concurrent incidents:** When multiple ambulances respond to an incident, there are two rows. These are merged into one, keeping count of the demand.

In addition to filtering, some unimportant feature columns for the incidents have been removed, while others have been converted to a more useful and understandable format. A couple of alternate versions of the dataset has been created as a result of processing. The main processed version, *incidents-processed*, is used for data analysis as well as the baseline for other versions. The processing steps of *incidents-processed* are outlined in Figure 2.2.1.

The *incidents-processed* dataset contains a large number of incidents over several years, which is not suitable for the simulation since it would take an excessive amount of time to simulate all incidents. For this reason only incidents in week 32 between 7.8.2017 and 14.8.2017 were chosen to be used for most simulations in this thesis. This week of incidents is denoted *incidents-simulation*. A separate version for simulations, *incidents-simulation-33*, was also created for comparison, containing incidents from week 33 in 2017. These versions are not processed any differently than *incidents-processed*, other than the time frame of incidents.

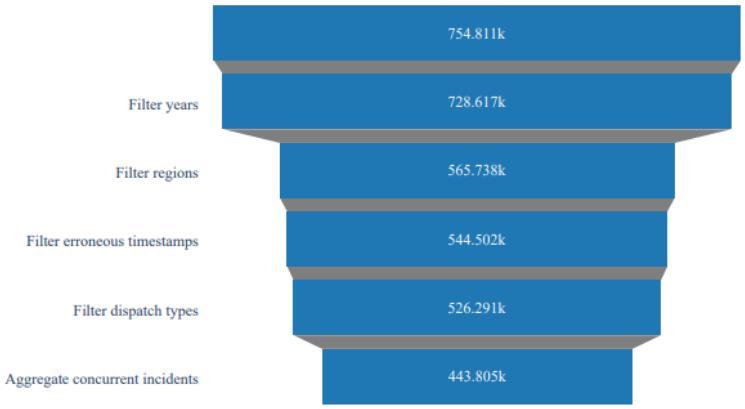


Figure 2.2.1: Processing steps resulting in *incidents-processed*.

Bekkevold and Schjølberg (2022) filtered out regular incidents in the data preparation for the simulation model because the dispatch behavior needed for a realistic simulation was considered outside the scope of their thesis. However, this led to a significant reduction in the number of incidents, which in turn reduced the likelihood of the EMCC facing difficult decisions regarding which ambulance to dispatch since there were likely many available. The reduced number of incidents is deemed unrealistic and reduces the potential to observe improvements from new methods. Therefore, regular incidents have been retained in the *incidents-processed* dataset for analysis and subsequently in the *incidents-simulation* dataset for simulating the system.

An additional version of the dataset, *incidents-processed-predict*, was also created. This version is almost identical to *incidents-processed*, but the regular incidents have in fact been removed. This was done for the purpose of predicting demand, where regular planned incidents would interfere with the prediction. This dataset is therefore essentially what was used as the main dataset in Bekkevold and Schjølberg (2022), which subsequently gave a smaller set of incidents for their simulation dataset *incidents-simulation-B&S*. A summary of all dataset versions is displayed in Table 2.2.2.

Dataset version	**Incidents**	**Description**
incidents-original	754 811	Raw dataset
incidents-processed	443 805	Main processed dataset
incidents-processed-predict	368 068	Regular incidents removed
incidents-simulation	2 005	Only week 32 in 2017
incidents-simulation-33	2 064	Only week 33 in 2017
incidents-simulation-B&S	1 625	Regular incidents removed

Table 2.2.2: Incident dataset versions with number of incidents and description.

2.3 Data Analysis

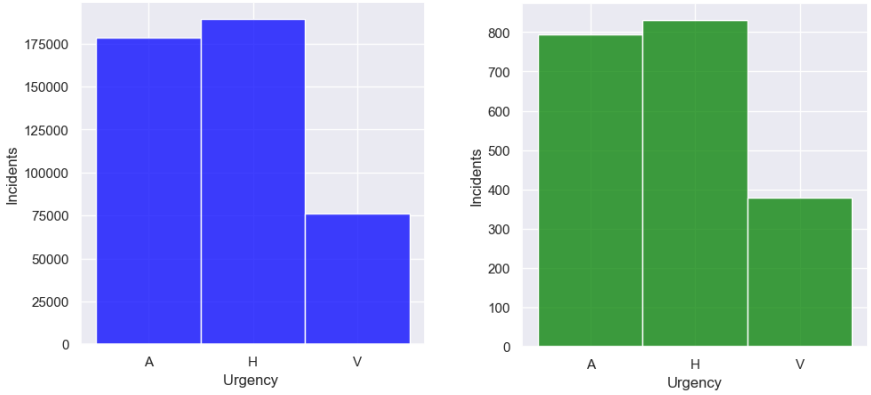
In this thesis it is interesting to analyse the response time, especially in combination with the urgency of the incidents. Additionally, spatial and temporal trends are analyzed in order to understand how to best predict incident demand.

2.3.1 Urgency

The urgency distribution of the incidents in the dataset is shown in Figure 2.3.1, where it can be observed that the distribution for *incidents-simulation* is almost identical to *incidents-processed* which shows that the simulation will not use incidents that differ from the norm in the data.

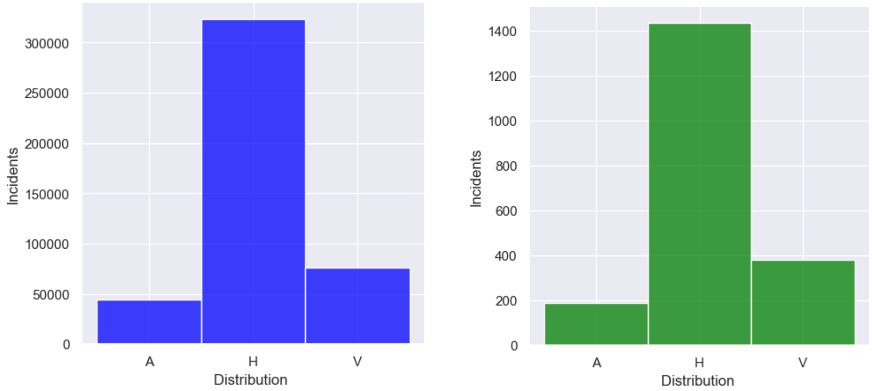
Figure 2.3.1 also reveals that there are approximately the same number of acute incidents as there are urgent incidents. However, as previously mentioned in Section 1.1, many incidents are assigned as acute as a precautionary measure. According to a contact person from OUS, the percentage of acute incidents that were later understood to actually be acute was as low as 20-25%, and the percentage of

incidents where a quick response time would make a difference in the patient's outcome were even lower. Figure 2.3.2 shows the estimated real urgency distribution of the incidents when 75% of the acute incidents are changed to be urgent.



(a) *incidents-processed* **(b)** *incidents-simulation*

Figure 2.3.1: Urgency distribution for acute(A), urgent(H), and regular(V) incidents.



(a) *incidents-processed* **(b)** *incidents-simulation*

Figure 2.3.2: Urgency distribution for acute(A), urgent(H), and regular(V) incidents in *incidents-processed* without over-triage.

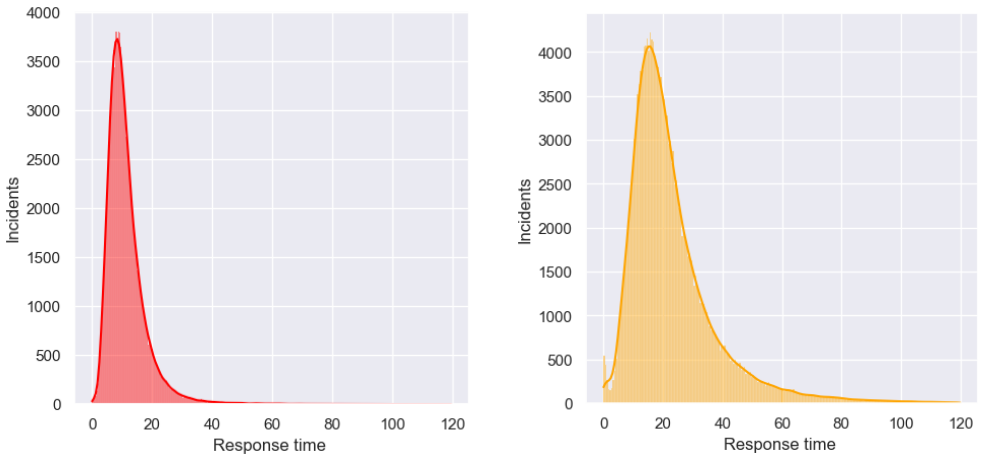
2.3.2 Response Time

The incidents with different urgencies naturally have different response times. Table 2.3.1 presents the average, median, and 90th percentile of response times for acute, urgent, and regular incidents.

	Average	**Median**	**90%-percentile**
Acute	11.57	9.9	18.75
Urgent	24.30	19.5	42.08
Regular	85.58	56.0	190.46

Table 2.3.1: Response time statistics for acute, urgent, and regular incidents.

Analyzing response time for acute and urgent incidents further in Figure 2.3.3, it is evident that the response times of acute incidents are more concentrated than those of urgent incidents. A response time limit of 120 minutes was set so that the outliers would not make the difference between the the graphs of acute and urgent incidents unreadable. The reason for the long response times of outliers remains unknown, but when analyzing one specific incident, there were not an exceptional number of incidents in the same time period causing a long queue. It is therefore reasonable to assume incorrect timestamps as the most likely cause. The longest response times for acute incidents reach up to 30 hours.



(a) Acute incidents **(b)** Urgent incidents

Figure 2.3.3: Histograms with kernel density estimation functions of response time under 120 minutes for acute and urgent incidents in *incidents-processed*.

2.3.3 Temporal and Spatial Trends

In the second goal in section 1.2, an implementation of demand prediction for an improved dispatch strategy was highlighted as promising. To get better insight of how to best predict incident demand, an analysis of both when and where the incidents occur has been done on the *incidents-processed-predict* dataset.

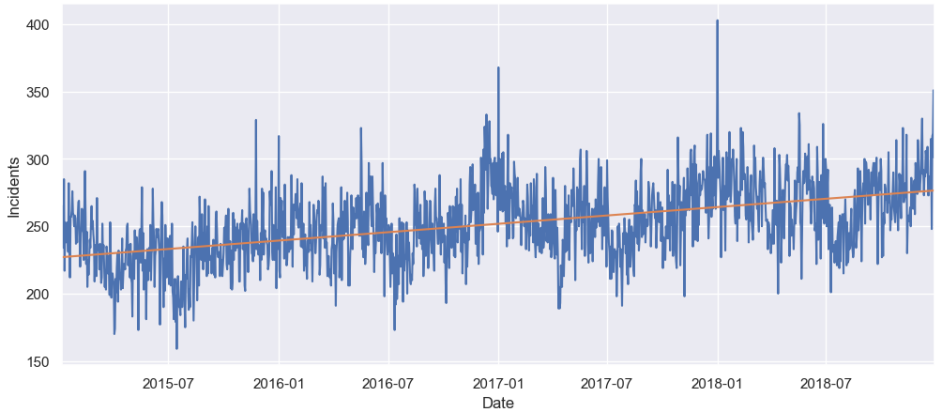


Figure 2.3.4: Total incidents per day from 2015 to 2018 in the *incidents-processed-predict* dataset. The orange line shows the general trend.

Temporal trends can provide valuable insights into patterns of behavior that may affect ambulance response demand. Figure 2.3.4 shows the total number of incidents per day in the whole dataset from 2015 to 2018, indicating that the number of incidents is generally on the rise. This is likely due to population growth.

Figure 2.3.5 shows the average count of incidents per day over a year, which reveals certain temporal patterns. The figure highlights that there are spikes in incident counts on New Year's Eve and Constitution Day on the 17th of May, which are public holidays and often involve celebrations that may lead to a higher number of incidents. Additionally, the figure shows that there are slightly more incidents in the winter months, which could be attributed to various factors such as weather conditions and the holiday season.

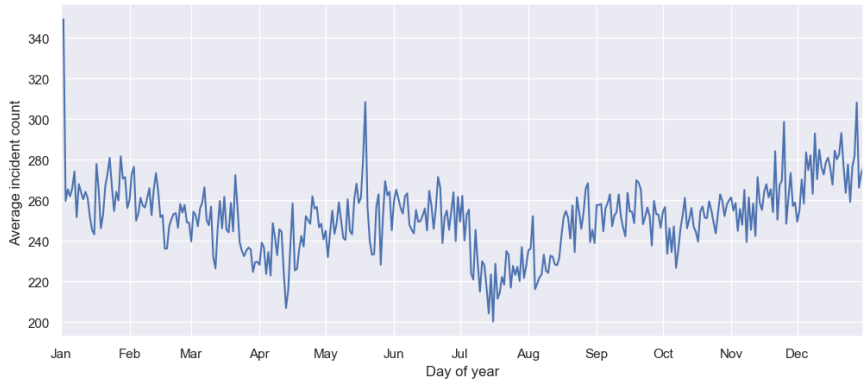


Figure 2.3.5: Average number of incidents per day of the year. The averages are from 2015 to 2018 in the *incidents-processed-predict* dataset.

Figure 2.3.6 displays the average incident counts per hour for each weekday, illustrating the variability of incident occurrences across weekdays and weekends. The weekend days differ from the weekdays by having more incidents at night, which is possibly due to increased social activity during those hours, such as parties and nightlife. Additionally, the weekends have fewer incidents during the daytime, which could be a result of people mostly staying at home during that time. Nevertheless, there is a general trend of a peak in incidents during midday, gradually decreasing towards the evening.

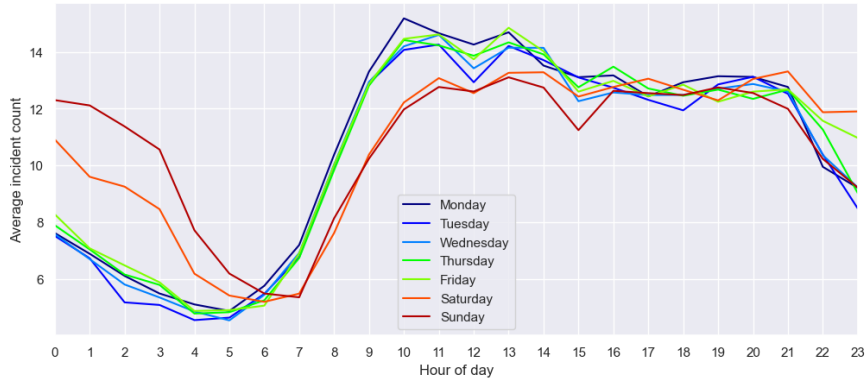
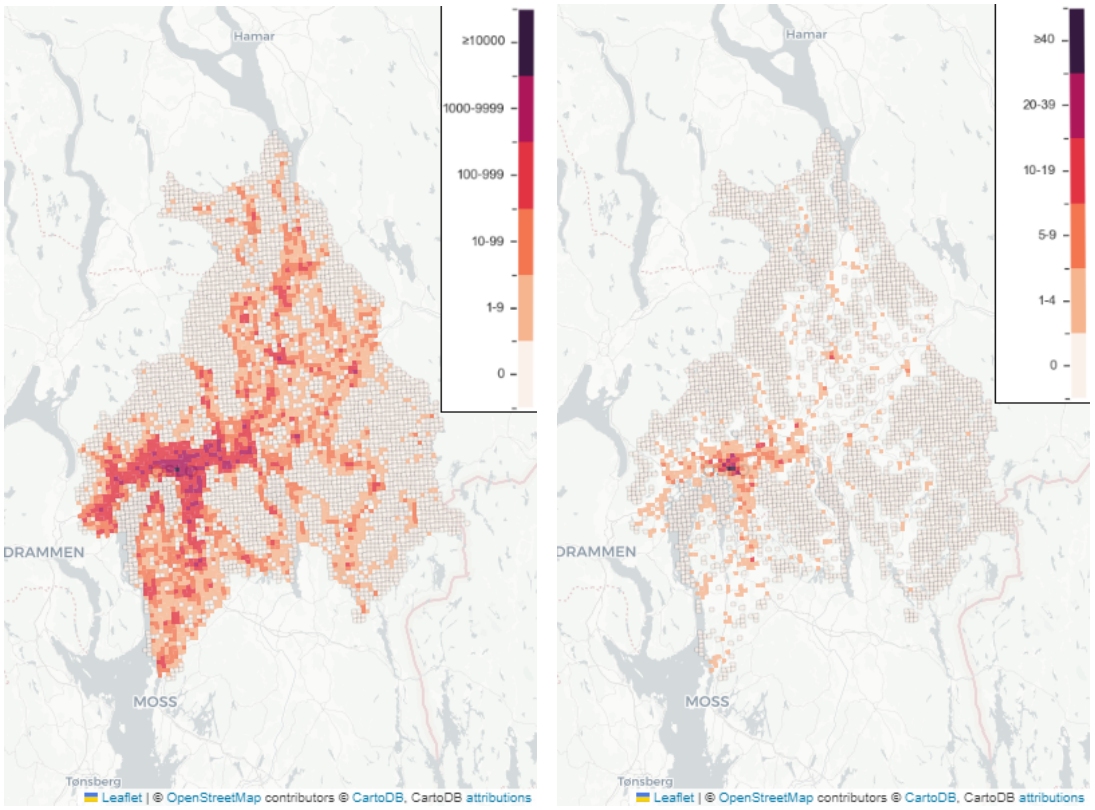


Figure 2.3.6: Average incident counts per hour for each weekday in the *incidents-processed-predict* dataset.

For the demand prediction to be useful in the dispatch decisions, it was necessary to employ a relatively high temporal resolution, which means that predictions are made for a small time period. Given the significant hourly difference in demand during a day shown in Figure 2.3.6, predicting demand per hour might be appropriate. This will however cause the data to become quite sparse.

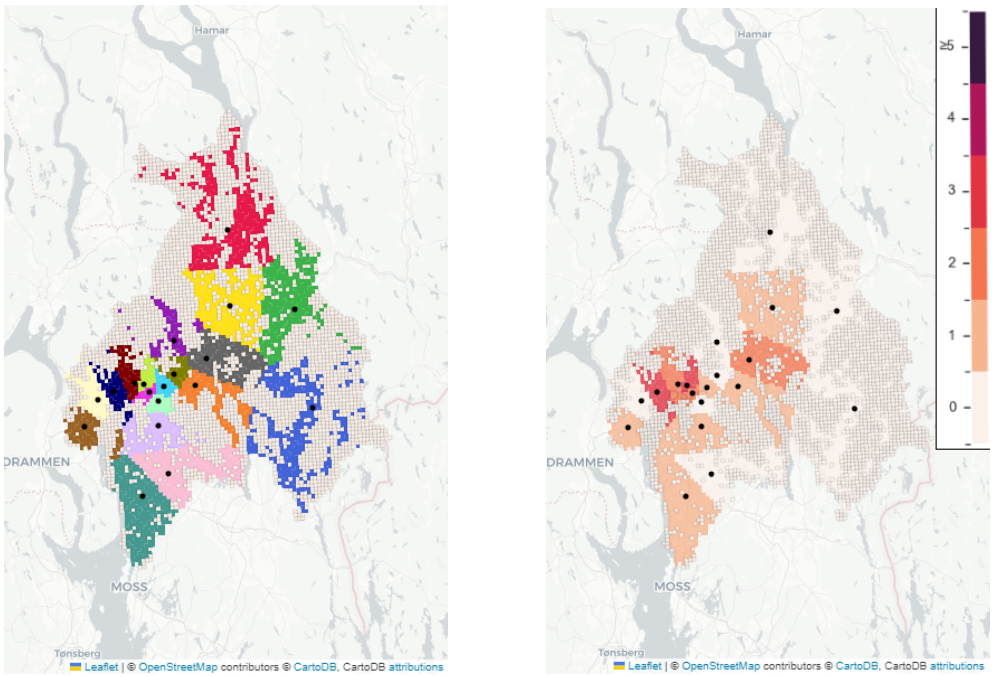
In addition to predicting when incidents will occur, it is useful to know in which areas they might occur as well. Spatial analysis is a powerful tool for understanding trends in incident demand across different geographical areas. As shown in Figure 2.3.7 there have been more incidents in highly populated areas like the center of Oslo. Figure 2.3.7b shows all incidents that occurred in the same time period as *incidents-simulation*, and it is evident that the sparsity of incident data increases when the time period is decreased to a week. The data becomes even sparser if one considers the predicted demand within single grid cells, making it challenging to create meaningful predictions.



(a) All **(b)** Week 32, 2017

Figure 2.3.7: Heatmaps showing total incidents per grid. Grids with no incidents in the entire *incidents-processed-predict* dataset are outlined with a border. Note that the two sub-figures have different color scales.

One possible solution to this sparsity is to group the individual grids into larger spatial areas. Bekkevold and Schjølberg (2022) divided the grids into responsibility areas for all the base stations using K-means clustering, and these areas were deemed appropriate for helping with the sparsity of incident counts. The responsibility areas are displayed in Figure 2.3.8, and a heatmap of incident counts across all responsibility areas within one hour is presented in the Figure 2.3.8b.



(a) Base station responsibility areas **(b)** Incident heatmap

Figure 2.3.8: Base station responsibility areas and heatmap showing total incidents per base station for the hour 11:00-12:00, 11.08.2017.

2.4 Allocation Problem

Optimizing ambulance allocation in order to reduce response time to incidents involves determining the optimal number of ambulances that should be distributed to different base stations, in order to cover areas most effectively. However, this problem quickly becomes challenging when the number of ambulances and base stations increase, due to the vast number of possible solutions.

2.4.1 Representation

To better understand the problem at hand, allocations can be represented in a more visually coherent way. One option which is shown in Equation 2.1, is to represent the ambulances and base stations as symbols in a string. Here, the number of stars between two bars represent the number of ambulances in a base station. Two subsequent bars in the string means that that base station is empty. Other than making sure the sequence of symbols start and end with a bar, the sequence can be reordered in any possible way to create another valid allocation. Since the ambulance department deals with both a day shift and a night shift, a solution will contain one allocation for the day shift and another allocation for the night shift.

$$Allocation = |**|*|***|*||***|*|*...| \tag{2.1}$$

2.4.2 Solution Space

The solution space is the set of all possible solutions to a given problem. The size of the solution space for the ambulance allocation problem is massive, and can be calculated by taking the product of the number of ways in which ambulances can be distributed to each base station. The total number of possible allocations for each shift can be calculated using Formula 2.2. This formula calculates the number of possible allocations of n ambulances to k base stations, taking into account that each base station can have any number of ambulances from 0 to n. The binomial function is used to calculate the number of ways to choose $n-1$ objects out of $n+k-1$ objects, which is equivalent to the number of ways to distribute n indistinguishable objects into k distinguishable containers, allowing for empty containers.

$$N_{solutions} = \binom{n+k-1}{k-1} = \frac{(n+k-1)!}{n!(k-1)!} \tag{2.2}$$

For the day shift, with 45 ambulances, there are approximately 2.59×10^{15} possible allocations. For the night shift, with 29 ambulances, there are approximately 4.57×10^{12} possible allocations. Since a solution is a combination of the two shifts, the total number of solutions is therefore approximately $(2.59 \times 10^{15}) \times (4.57 \times 10^{12}) = 1.18 \times 10^{28}$. Thus, exploring the entire solution space is not feasible, and finding an optimal solution using an exhaustive search algorithm is not practical. This calculation matches the analysis done by Bekkevold and Schjølberg (2022).

2.4.3 Software Tools and Hardware

Most of the software tools used for the implementation in this thesis is continued from the implementation of Bekkevold and Schjølberg (2022).

Java 18 was used for the main components of the implementation, including simulation and optimization. For visualization of the simulation, JavaFX 19 was used, together with Mapjfx (Meisch 2023). Visualization of an optimization method used an interface library called matplotlib4j (Nakamura 2023), which uses the Python library Matplotlib.

Python 3.10 was used for data analysis using the Pandas and Numpy libraries. Python was also utilized for generating most of the graphs and visualizations in this thesis, mainly with Matplotlib. Visualizations for map data was made with tools such as Selenium and Folium. Prediction models was developed with Keras and Statsmodels.

An open source map tool called Open Street Map (OSM 2023) was used to calculate travel times for the simulation, as explained in Section 5.1.3.3.

The implementation code for the thesis can be found in the GitHub repository at https://github.com/erikmoh/ambulance-optimization. Since the dataset provided by OUS contains sensitive data, it is not included in the repository. Most of the implementation depends on this data.

2.4.3.1 Hardware

The system on which the implementation was developed and the results were generated for this thesis had the following specifications:

- Operating System: Windows 10 (64-bit)
- Processor: 11th Gen Intel Core i7-11700K @ 3.60 GHz
- RAM: 32 GB

CHAPTER

THREE

THEORY

This chapter will present methods related to simulation, prediction and optimization, which are all important parts of the thesis. The goal of the chapter is to provide the reader with knowledge and understanding of the methods that are used in the rest of this thesis.

3.1 Simulation

Simulation is an essential tool in evaluating the performance of EMS systems without making any changes to the real-world system. By simulating the system, the response times of the incidents can be calculated, which can be used as an output to evaluate the system's performance. Additionally, simulation allows for experimentation with different system configurations to evaluate the impact of changes, which may not be feasible or ethical to do on a real EMS system with observational studies or controlled experiments.

3.1.1 Discrete Event Simulation

Discrete Event Simulation (DES) is a common method used in simulation where the system is modeled as a sequence of events that occur at specific points in time Ridler, Andrew J. Mason, and Raith 2022. Each event can modify the state of the system, such as when an ambulance is dispatched to an incident or when it arrives at the incident location. To calculate the state of the system at each event, a set of update equations is used. Equation 3.1 shows how the system state changes as a result of the event. In this equation, X_t represents the state of the system at time t, and U_t represents the input at time t. The function f is used to describe how the system state evolves from one time step to the next.

$$X_{t+1} = f(X_t, U_t) \tag{3.1}$$

3.1.2 Continuous Simulation

Continuous simulation is another option where the state of the system changes continuously over time Raczynski 2003. This approach may be used in situations where events are more difficult to predict and model. The system's state is calculated using a set of differential equations, as seen in Equation 3.2. The equation represents the rate of change of the state of the system, where X is the state of the system, U is the input or control, and f is a function describing how the state of the system changes over time.

$$\frac{dX}{dt} = f(X, U) \tag{3.2}$$

In some cases, a combination of both discrete and continuous simulation may be used to model complex systems that have both discrete events and continuous changes.

3.2 Prediction

One part of this thesis revolves around predicting future demand for ambulance services to make better informed decisions about ambulance dispatching. This section will explain relevant machine learning methods that are used for prediction.

3.2.1 Artificial Neural Network

One common method for prediction is the use of neural networks, which are models inspired by the structure of the human brain. The biological brain contains billions of neurons that propagate signals between them in a way that creates complex cognitive processes. Artificial Neural Networks (ANN) aim to mimic this propagation of information in a simplified way. ANNs consist of layers of artificial neurons, and the propagation of information from one neuron to the next is based on the inputs to the neuron. Each incoming connection to a neuron has a weight associated with it, which is multiplied by the activation value of the previous neuron. These weighted values are then summed and passed through an activation function that determines whether the neuron will fire and propagate its output to the next layer of neurons. The weights of the connections between neurons are learned through a training process that involves adjusting the weights to minimize the difference between the predicted output and the target value. This process is often referred to as backpropagation, where the error at the output layer is propagated backward through the network to update the weights. A simple example of an ANN is shown in Figure 3.2.1.

3.2.2 Feature Extraction and Selection

Feature extraction and selection are important steps in preparing data for use in prediction models, including ANNs. Feature extraction is the process of selecting and transforming relevant features from raw data in order to enhance the performance of the model. Giving the prediction model enough features is important to

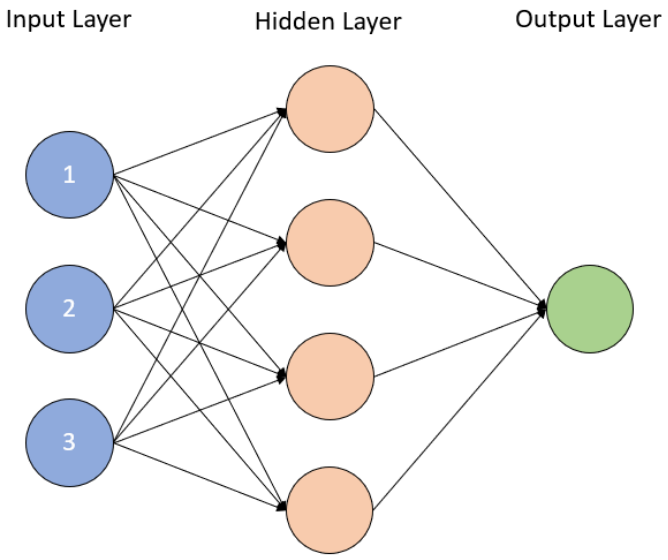


Figure 3.2.1: A simple ANN with three input features, one hidden layer, and one output neuron.

avoid underfitting, which occurs when the model is too simple and fails to capture the underlying patterns in the data.

Feature selection is to identify the most relevant features to use in the model. By reducing the number of input variables the risk of overfitting is reduced, and the model's performance might improve. Overfitting occurs when the model becomes too complex and starts to fit to the noise in the data rather than the underlying patterns. This can lead to poor generalization performance when the model is applied to new, unseen data.

3.2.3 Regression and Classification

Machine learning models can be used for both regression and classification tasks. Regression involves predicting a continuous output variable, while classification involves predicting a categorical output variable. The choice between regression and classification depends on the nature of the problem being addressed. Regression is typically used when the goal is to predict a numerical value, such as the price of a house. On the other hand, classification is used when the goal is to assign a label to a given input, such as classifying emails as spam or not spam.

3.2.4 Evaluation

Evaluation of model performance means to make sure that the model is making accurate predictions on new data. One common approach for evaluating models is to use metrics that quantify how well the model is able to make predictions on a test dataset.

The mean squared error (MSE) is a widely used metric for evaluating the performance of regression models. It quantifies the average of the squared differences between the predicted output values and the actual values in a dataset. Specifi-

cally, for each data point, the difference between the predicted output value and the actual output value is calculated as $y_i - \hat{y}_i$, where y_i is the actual output value and $\hat{y}_i$ is the predicted output value. This difference is then squared as $(y_i - \hat{y}_i)^2$. Finally, the average of all these squared differences is computed to obtain the MSE. The formula for MSE can be expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.3}$$

3.2.5 K-fold Cross-validation

In addition to evaluating a model using predictions on a test dataset, it is common to employ more robust methods such as k-fold cross-validation. In k-fold cross-validation, the data is split into k equally sized subsets, or folds. The model is then trained on $k-1$ folds and evaluated on the remaining fold. This process is repeated k times, with each fold used exactly once for evaluation. The results from the k evaluations are then averaged to provide an estimate of the model's performance on new, unseen data. An example is shown in Figure 3.2.2.

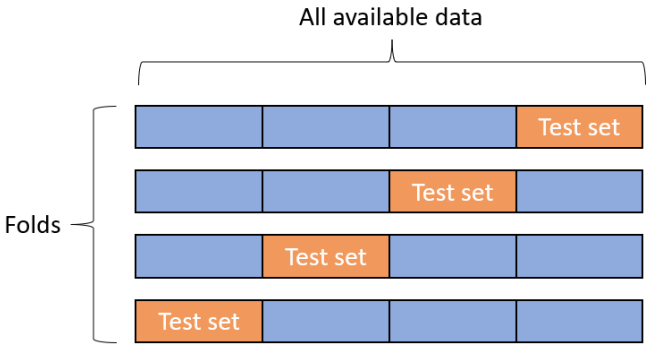


Figure 3.2.2: K-fold cross-validation with k=4.

3.2.6 Poisson Regression

In addition to neural networks for prediction, there are several other statistical and machine learning methods that can be used to model and predict data. One popular statistical method is the Poisson regression model, which is used to model count data, such as the number of accidents, emergency room visits, or insurance claims.

The Poisson regression model estimates the expected count of events based on one or more predictor variables, assuming that the count data follows a Poisson distribution Yang and Berdine 2015. The model uses a logarithmic function to predict the expected count and finds the coefficients that maximize the likelihood of observing the observed count data.

3.3 Optimization

As explained in Section 2.4.2, the solution space of possible allocations is too large to be explored using an exhaustive search algorithm.

Rather than blindly searching through a vast set of solutions, it is often more efficient to use heuristics in the optimization process. Heuristics involve incorporating "rule of thumb" knowledge to guide the search towards better solutions. One such heuristic is the Genetic Algorithm (GA), which is inspired by the process of evolution. This algorithm maintains a population of candidate solutions and evaluates their quality, or fitness, to guide the search process. Based on the evaluation, certain solutions are selected to combine into new solutions, while others are chosen to survive into the next generation. This iterative process continues until a satisfactory solution is found or another termination condition is met.

3.3.1 Parent Selection

Parent selection is a key aspect of the GA, as it determines which solutions will be used to create the next generation. There are various methods for selecting parents, each with its own advantages and disadvantages. One of the commonly used methods is tournament selection, which involves randomly selecting a small subset of solutions from the population and selecting the one with the highest fitness value as a parent. This process is repeated to select a second parent.

One important aspect related to parent selection is the concept of selection pressure. Selection pressure refers to the degree to which fitter individuals are favored in the selection process. Higher selection pressure gives a stronger advantage to individuals with higher fitness values, leading to a more exploitative search, while lower selection pressure allows for greater exploration of the search space. Tournament selection provides a means to adjust the selection pressure by controlling the tournament size. A larger tournament size increases selection pressure, since a high quality individual is more likely to be a part of the tournament.

3.3.2 Genetic Operators

The GA works by combining two solutions to create new solutions with potentially better fitness. This is achieved through two main genetic operators: crossover and mutation.

Crossover involves selecting two candidate solutions and exchanging some of their genetic information to create two new solutions. This allows the algorithm to explore different combinations of good solutions and potentially create better solutions. Mutation, on the other hand, involves randomly changing some part of a single solution. This is done to introduce diversity into the population and to prevent the algorithm from getting stuck in local optima.

3.3.3 Survival Function

The survival function determines which solutions will survive to the next generation. The survival function can be based on a variety of criteria, such as fitness score or diversity. One common approach is to use a combination of fitness and diversity, where solutions with high fitness scores and low similarity to other solutions in the population are more likely to survive.

Another approach is to use elitism, which involves carrying over the best solutions from the current population to the next generation without any changes. This ensures that the best solutions are not lost in the search process and can be further improved upon in future generations.

3.3.4 Diversity

Maintaining diversity in the population is crucial for the GA to work effectively. If all the solutions in the population are very similar, the algorithm will converge to a suboptimal solution. One way to maintain diversity is to use Island Model Genetic Algorithm (IMGA), which utilizes separate subpopulations that evolve independently. This allows for different parts of the search space to be explored simultaneously and can increase the chances of finding a good solution.

Another technique employed to maintain diversity in genetic algorithms is crowding. Crowding is a mechanism utilized in the survivor selection step to preserve the diversity of solutions within the population and prevent premature convergence towards local optima. It consists of pairing the offspring to similar solutions in the population, before selecting one of the solutions in each pair that is carried on into the next generation, based on a replacement approach. Two main approaches are Deterministic and Probabilistic crowding, where deterministic crowding always choose the most fit solution in the pair, while probabilistic crowding chooses a solution with a probability according to the fitness of the solutions in the pair. These methods reduces the number of similar solutions in the population.

3.3.5 Parameter Tuning

Like many optimization algorithms, the performance of the GA depends on the values of several parameters, such as population size, crossover rate, mutation rate, and selection pressure. Proper tuning of these parameters can greatly improve the performance of the algorithm. Several methods can be used for parameter tuning, including grid search and manual tuning. The parameter values could also change during the search process of the GA.

Grid search involves defining a grid of possible parameter values and exhaustively evaluating the algorithm's performance for each combination of parameter values in the grid. It systematically explores all combinations, making it a brute-force approach. Grid search is easy to implement and interpret, but it can be computationally expensive, especially when dealing with a large number of parameters or a wide range of parameter values.

Manual tuning involves iteratively adjusting the values of the parameters of an optimization algorithm based on observation and experimentation. It relies on personal judgment and domain knowledge to select parameter values that are expected to improve the algorithm's performance.

Another way to control a parameter is to automatically modify the value of the parameter during the search process. This could be done either by adapting the value based on the performance or diversity of the population, or to change the parameter value according to the current generation of the GA.

3.3.6 Constraints

In many optimization problems, the search space is limited by constraints that must be satisfied in addition to finding the optimal solution. The GA can be extended to handle such constraints through several methods, such as penalty functions, repair algorithms, and feasibility rules. These methods modify the genetic operators or the fitness function to ensure that the solutions generated by the algorithm satisfy the constraints.

3.3.7 Multi-Objective Optimization

In some cases, optimization problems involve multiple conflicting objectives, and the goal is to find a set of solutions that are optimal with respect to all the objectives. The GA can be extended to handle multi-objective optimization problems through methods such as Pareto optimization and weighted sum methods. These methods aim to find a set of solutions that are optimal with respect to a trade-off between the different objectives.

NSGA-II (Non-dominated Sorting Genetic Algorithm II) is a widely used multi-objective optimization algorithm that extends the basic GA framework to handle problems with multiple conflicting objectives. Developed by Deb et al. (2002), NSGA-II aims to find a set of solutions that represent the optimal trade-off between the different objectives.

One of the key aspects of NSGA-II is its utilization of Pareto dominance, a concept that allows for the comparison of individuals in the population based on their objective values. In NSGA-II, an individual is considered to dominate another if it is better in at least one objective and not worse in any other objective. This dominance relationship forms the foundation for identifying non-dominated solutions, which are the ones that cannot be improved upon in any objective without sacrificing performance in another objective.

To classify individuals into different fronts based on their dominance relationships, NSGA-II applies a non-dominated sorting technique. This sorting process results in a hierarchy of fronts, with the first front containing the non-dominated individuals, followed by subsequent fronts where individuals are dominated by those in the preceding front. By organizing the population into fronts, NSGA-II is able to maintain diversity and establish a ranking mechanism for selection.

In order to ensure a well-distributed set of solutions along the Pareto front, NSGA-II employs the concept of crowding distance. Crowding distance measures the density of solutions surrounding an individual in the objective space. Solutions with larger crowding distances are preferred during the selection process as they contribute to a better coverage of the front. By considering both dominance and crowding distance, NSGA-II strikes a balance between exploring diverse regions of the search space and focusing on promising areas that exhibit high-quality solutions.

CHAPTER

FOUR

RELATED WORK

This chapter will explore and discuss work that has been done in relation to the topics of this thesis. Literature concerning the problem domain of ambulance location and allocation will be discussed first, before exploring the many options that have been presented in work related to optimization.

4.1 Problem Domain and Simulation

This section will present literature related to the simulation and some different available approaches of implementation. It will also cover survivability as an evaluation metric.

4.1.1 Simulation Approaches

In the domain of emergency management systems and emergency medical response, numerous studies have been conducted on ambulance location and allocation. Many of these studies focus on the coverage provided by ambulance allocations. For example, Schmid and Doerner (2010) developed a model that considers time-varying coverage areas. However, coverage-based approaches often struggle to accurately represent important operational factors such as ambulance availability (Zaffar et al. 2016). As a result, simulation methods have been suggested as a more accurate approach (McCormack and Coates 2015). Simulation models improve accuracy and realism, leading to better results (Yue, Marla, and Krishnan 2021; Henderson and A. Mason 2004). While deterministic models have been common in the literature, Beraldi and Bruni (2009) notes the increasing prevalence of probabilistic models, proposing their own stochastic model for emergency service facility location with demand uncertainty.

Almost all of of the literature related to simulating EMS systems employ a DES as simulation approach (Ridler, Andrew J. Mason, and Raith 2022). One example is

presented in Lam et al. (2015), which utilizes a DES to evaluate dynamic allocation plans in Singapore. Their simulation uses historical emergency calls data to model when the calls occur, in addition to response delays and travel times. Interestingly, their approach modeled the pre-dispatch delays and the handover delays from empirical distribution functions based historical data. These delays are in this thesis referred to as handling time, dispatch time, and hospital time. Zaffar et al. (2016) utilized a simplified service completion time, estimating weighted averages of on-scene time, travel time, and drop-off time from the data. Kergosien et al. (2015) mentions that uncertain random delays should not be generated dynamically, but rather generated a priori since variance in the results should be eliminated in order to enable comparisons of changes to the EMS system without the influence of random factors.

Van Barneveld et al. (2018) notes that historical travel times should not be used in the simulation since the observed historical times are largely dependent on the location of the ambulance that responded at the time. They noted that this location is a result of previous incidents, so an estimation of travel times has to be used. Zaffar et al. (2016) calculated travel times using the average speed and the Manhattan distance between specific zones. Other approaches to travel time calculation include modelling a road network with either deterministic or stochastic travel times. As mentioned in Ridler, Andrew J. Mason, and Raith (2022), Andrew James Mason (2013) uses a road network with both deterministic and temporally dynamic travel times. Ridler, Andrew J. Mason, and Raith (2022) incorporated a road network from Open Street Map (OSM), an open-source project for geographic data (OSM 2023), to find all pairs of shortest paths in the network. This network was subsequently simplified by reducing intermediate nodes, and saving the network to be used for all simulations. Their implementation included off-road travel times, which may occur when the incident location is not on a road. Additionally, the travel speed for an ambulance travelling with lights and sirens were set to be 43% faster than regular travel speed.

Ridler, Andrew J. Mason, and Raith (2022) mentions that planned transports of patients may occur in the EMS system, but that most models do not include these regular incidents. Kergosien et al. (2015) included planned incidents by making the simulated EMS system handle both emergency requests and patient transport. They mention that these incidents are generally received dynamically, but in advance so that they can be scheduled. It is further discussed that some EMS systems split the ambulances into two groups where one of the groups is assigned to transporting tasks.

To validate their simulation model, Ridler, Andrew J. Mason, and Raith (2022) implemented a visualization of the simulation which enabled verification that the ambulances follow the right order of operations, and generally that the EMS system behaves as it should. The simulation model was also compared to a validated simulation model which gave almost the exact same response times from the same input data.

4.1.2 Dispatch Behaviour

The dispatch behavior plays a crucial role in ambulance allocation simulations. One simplification commonly used is assuming that an ambulance is dispatched only from its base station. However, it is also common to relax this restriction and allow dispatch while the ambulance is returning to the base station (Zaffar et al. 2016; McCormack and Coates 2015). Similarly to this relaxation, ambulances can be considered available for dispatch in other situations. Ridler, Andrew J. Mason, and Raith (2022) mentions that a common form for this is redispatch, where an ambulance assigned to a low-priority incident can be redispatched to a new high-priority incident nearby. In this thesis this operation is referred to as reassigning. They also mention a second situation which is when the ambulance assigned to a call can be changed because some new ambulance that is closer to the call has just become available. A third situation is presented in Van Barneveld et al. (2018), which is about ambulances that are currently dropping off a patient at a hospital. In this situation it is evident that the ambulance will become available in the near future. Their results show that considering these ambulances as available has no significant impact on response times. This operation is referred to as queuing in this thesis.

Other more complex dispatching decisions are further expanded upon in Ridler, Andrew J. Mason, and Raith (2022), who explains that some EMS operators do not always dispatch the closest ambulance to low priority calls. In some situations it can be beneficial to dispatch an ambulance that is further away in order to keep the available ambulances in a preferable state. This desirable state refers to the preservation of area coverage. Van Barneveld et al. (2018) proposes the notion of coverage being a reflection of preparedness of the system to respond to future calls. They then present one common method for increasing the coverage, called redeployment, which consists of relocating an idle ambulance to an area that in turn provides the highest expected coverage. This method is closely related to the maximum expected coverage location problem (MEXCLP) (Daskin 1983). Coverage can be estimated from several factors, such as the number of available ambulances within a certain area, and the expected demand for the area.

Another approach of utilizing incident priority and future demand for dispatch strategies is presented in Bandara, Mayorga, and Albert (2012), who proposes a solution that incorporates a Markov Decision Process. This decision process enhances dispatching by taking two levels of incident urgency into account.

Stochastic programming formulations have also been suggested, considering uncertainty about future emergency demand over two stages (Chun Peng 2020). One stage considers the probability of covering the demand while minimizing the cost, and the second stage employs probabilistic constraints that enables control of the degradation of coverage.

4.1.3 Demand Prediction

Predicting future demand can consist of both spatial and temporal trends. The resolution of these trends are highly dependent on the domain, and the formal

definition of what counts as a low or high resolution is nonexistent, as explained in Van De Weijer and Owren (2022). For the case of predicting ambulance demand, H. Huang et al. (2019) presents a Poisson Neural Network for predicting daily demand of the city of Ningbo in China, assuming that the data follows a Poisson distribution. Another method for demand prediction is proposed in Chen et al. (2016), which uses ANNs to, among other combinations of resolution, predict demand for 3-hour periods in 3km x 3km areas. These methods utilizes input features such as weather, year, and weekday, but also extra generated features such as weekend and season which can help the methods learn the trends that the features are related to.

In addition to ANNs, statistical methods for predicting demand has also been researched. Lam et al. (2015) presents a geographical information system to analyze the spatio-temporal heterogeneity of emergency call volumes. This system utilizes a statistic, which essentially is a z-score that can facilitate the statistical test for identifying statistically significant hot and cold spots. Another statistical method is presented in Zhou and Matteson (2015), who use a spatio-temporal kernel density estimation to predict hourly demand in Toronto in Canada.

4.1.4 Urgency

When the EMCC operators receive an emergency call, they have to decide on which urgency in the triage the incident belongs to. This task can be challenging, as explained in Ivanov et al. (2021) who also proposed using both Natural Language Processing and Machine Learning methods to improve assigning accuracy.

4.1.5 Survivability

Several studies suggest that survivability is a better evaluation metric than response time or coverage alone (Zaffar et al. 2016; Bandara, Mayorga, and Albert 2012). Erkut, Ingolfsson, and Erdoğan (2008) introduced survival functions based on response time, providing the probability of survival for specific incident types, particularly cardiac arrests. They argue that response time is important for other acute incidents as well, and that the probability of recovery decreases gradually over time, even though these incidents are likely to have different survival functions. This work has been expanded upon by various authors, leading to the proposal of different measures for assessing survivability.

Knight, Harper, and Smith (2012) proposed slightly different survival functions as those presented in Erkut, Ingolfsson, and Erdoğan (2008), even though they based their functions partially on the same research. They also proposed a heterogeneous approach, using different survival functions for different urgencies. The survival functions for the lower-priority incidents are step-functions, based on the respective response time goals. These survival functions have different associated weights, in order to appropriate the consequences of response time for different urgencies.

4.2 Optimization

Both coverage and response time optimization problems have been explored using various techniques. Metaheuristic techniques such as the GA have been commonly employed. For instance, Aytug and Saydam (2002) implemented two versions of the GA to solve a MEXCLP. Simulated annealing and Tabu search have also been suggested as optimization techniques (McCormack and Coates 2015). Additionally, particle swarm optimization has been investigated for ambulance distribution (Zhang et al. 2022).

4.2.1 GA

In the realm of optimization algorithms, one key aspect that significantly influences their effectiveness is the representation of solutions. A well-designed representation ensures the attainment of feasible solutions without the requirement for continuous feasibility checks. McCormack and Coates (2015) adopts an encoded string of genes as the chosen representation for solutions, for optimizing locations of potential base stations and the ratio of emergency vehicle types at each base station in addition to the ambulance allocation. This optimization problem requires a more advanced representation than what is needed for the optimization problem in this thesis.

Diversity preservation in GAs are important for their performance. One method of preserving diversity is the IMGA as presented in Whitley, Rana, and Heckendorn (1998). Gozali and Fujimura (2019) proposes a further improvement by a localized strategy for the migration procedure often seen in IMGAs. Chang, W.-H. Huang, and Ting (2010) mentions that the applications of GAs for solving combinatorial problems are often faced with early convergence, and proposes a dynamic diversity control method to counter this.

Another method for preserving diversity is presented in Mengshoel, Galán, and de Dios (2014) who explored an adaptive generalized crowding for GAs. In addition to having a scaling factor to influence the replacement rule, they proposed an approach where this scaling factor is adapted according to the diversity of the population. They also explored a self adaptive method where the scaling factor is a part of the representation of the solution, and in turn is a part of the search process.

4.2.2 Multi-Objective Optimization

Multi-Objective Optimization is a commonly utilized method when faced with multiple objectives. One application for this is presented in Olivos and Caceres (2022), optimizing mean response time, maximum response time, and uncovered demand of the EMS system of Antofagasta in Chile. They incorporate a Pareto set of efficient solutions.

Knight, Harper, and Smith (2012) mentions that a variety of survival functions for different urgencies are in essence a multi-objective optimization when weighting of the survival functions is applied.

CHAPTER

FIVE

METHOD

This chapter presents the research conducted for the thesis, focusing on the architecture changes and experiments performed. The structure of the chapter follows the research goals outlined in Section 1.2. It begins with improvements to the simulation, followed by the exploration of different dispatch strategies, including demand prediction. The chapter then examines the impact of more accurate urgency assignment and concludes with enhancements to the optimization method. An overview of how the simulation and the optimization method interacts is presented in Figure 5.0.1. For all the simulations done in this chapter, the *incidents-simulation* dataset outlined in Section 2.2 is used.

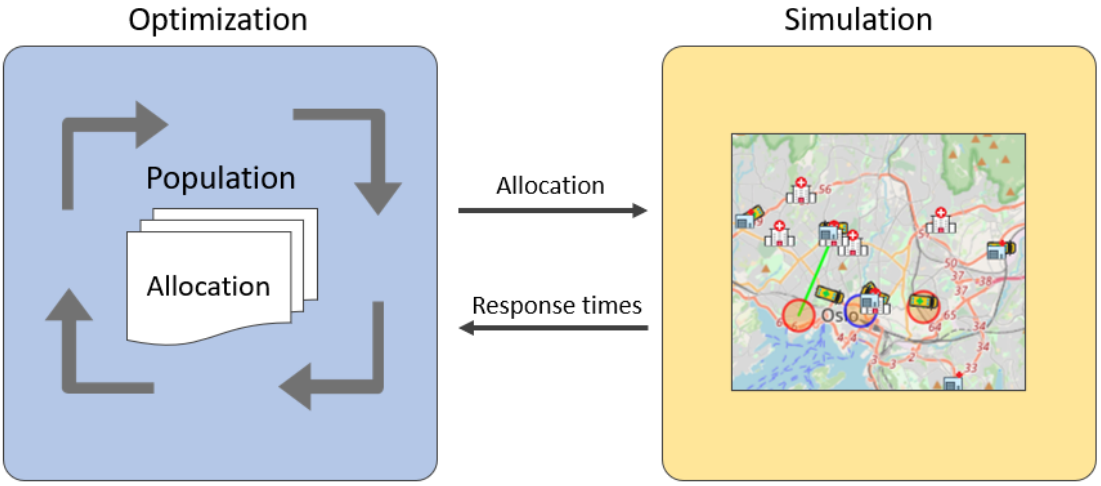


Figure 5.0.1: Overview of the optimization and simulation interaction.

5.1 Goal 1: Improve Simulation Realism

In the study conducted by Bekkevold and Schjølberg (2022), the developed simulation demonstrated a comprehensive and reasonably accurate representation. Notably, the timing of events and the response times in the simulation was similar to what is reflected in *incidents-original*. While it is challenging to capture every intricate detail of the system, there are still improvements that can be made.

5.1.1 Discrete Event Simulation (DES)

DES was chosen in the existing simulation, using a sequence of events that occur at specific points in time and changing the state based on update equations for each event. To simulate the continuous nature of ambulance movement, an adaptation was made to fit the movement into an event based simulation by updating an ambulance's location at 5 minute intervals. This adaptation was only used when an ambulance was travelling back to its base station so that it could respond to a new nearby incident. In the implementation used in this thesis however, all ambulance movement utilizes this location update adaptation. Although this very slightly increases the time it takes to run a simulation, it creates a more realistic representation of the system which enables the exploration of different dispatch strategies presented in Section 5.2.

Pseudo code for the main parts of the updated simulation is presented in Algorithm 1. This algorithm includes the update equations for each state, but many details are not included in the pseudo code since the simulation is too complex.

5.1.2 Regular Incidents

In contrast to the simulation developed by Bekkevold and Schjølberg (2022), this simulation includes regular incidents that are treated differently from urgent and acute incidents in some situations. The regular incidents in the dataset are either planned or unplanned events, but the behaviour of planned incidents is not included in the simulation, as with most simulations in the literature Ridler, Andrew J. Mason, and Raith 2022.

5.1.3 Response Time

The primary objective of the simulation is to evaluate the system with different allocations using the resulting response times. It is therefore important that the response times of incidents are accurately simulated and calculated. The evaluation only considers response times for acute and urgent incidents, since regular incidents are not time-critical and are often planned events. As seen in Figure 1.1.1, the response time t_R is a sum of handling time t_H, dispatch time t_D, and travel time t_T, expressed by Equation 5.1. These times are explained in 5.1.3.1, 5.1.3.2, and 5.1.3.3 respectively. In Algorithm 1, t_R is referred to as *event.duration* on Line 13, which determines when the Scene Arrival event should happen.

$$t_R = t_H + t_D + t_T \tag{5.1}$$

Algorithm 1 Discrete event simulation

Input: allocations X, configuration parameters θ
Output: list of response times $r = (r_0, r_1, ..., r_{\max})$
function SIMULATE(X, θ)
    $r \leftarrow (), C \leftarrow \emptyset, Q \leftarrow$ initializeEventQueue()
    ambulances $\leftarrow$ initializeAmbulances(X)
    while Q is not empty **do**
        event $\leftarrow Q$.pop(), $t \leftarrow$ event.time
        ambulances $\leftarrow$ setCurrentShift(t)
        switch event **do**
            case NewCall
                ambulances $\leftarrow$ dispatch(event)
                if |ambulances| > 0 **then**
                    Q.add(SceneArrival($t +$ event.duration, event))
                else
                    C.add(event)
                end if
            case AbortIncident
                for each ambulance $\in$ event.ambulances **do**
                    ambulance.flagAsAvailableOrFinishShift()
                end for
                CheckQueue(C)
            case SceneArrival
                append event.responseTime onto the end of r
                Q.add(SceneDeparture($t +$ event.duration, event))
            case SceneDeparture
                for each ambulance $\in$ event.ambulances **do**
                    if ambulance.isTransport **then**
                        $t_A \leftarrow$ event.duration
                        Q.add(HospitalDeparture($t + t_d$, ambulance))
                    else
                        ambulance.flagAsAvailableOrFinishShift()
                        CheckQueue(C)
                    end if
                end for
            case HospitalDeparture
                event.ambulance.flagAsAvailableOrFinishShift()
                CheckQueue(C)
            case LocationUpdate
                event.ambulance.updateLocation()
                if event.ambulance.isNotAtDestination() **then**
                    Q.add(LocationUpdate($t + \Delta t$, event.ambulance))
                end if
    end while
    return list of response times $r = (r_0, r_1, ..., r_{\max})$
end function

5.1.3.1 Handling Time

Handling time is the duration between when the EMCC gets called about an incident to when they call the ambulance that is dispatched. This time is used by the EMCC operators to listen to the caller describe the situation to get an understanding of the urgency of the incident among other information that might need to be relayed to the ambulance personnel. Additionally, the EMCC have to decide on which ambulance to dispatch, if there are any available.

Handling time is highly situational, as it depends whether the EMCC is overloaded with many incident calls, the location and number of currently available ambulances, and if there are other incidents or events that needs to be prioritized. In a simulation which is made for experimenting with changes of the allocation, the status of the available ambulances might be completely different from what it actually was. So although Notified time is present in the dataset, as shown in Table 2.2.1, it was decided to not utilize this time to determine the handling time for each incident. Instead, the median handling times corresponding to the different incident urgencies was used, similarly to what was done in Lam et al. (2015) and Zaffar et al. (2016). Table 5.1.1 shows that the median handling time t_{H_M} for regular incidents is very long, probably because other incidents were prioritized at that time, which might not be the case in the simulated state. Since response times of regular incidents are not used in the evaluation, it was decided to assume instant handling time for regular incidents.

Urgency	**Acute**	**Urgent**	**Regular**
t_{H_M}	2m 5s	6m 5s	36m 26s

Table 5.1.1: Historic median handling times t_{H_M} for incidents in *incidents-processed*.

If there are no available ambulances when the EMCC receives a call about an incident, it is put into a call-queue which is serviced when an ambulance becomes available. In such cases, the total handling time is the max duration of the queue time and the historic median time, since it is assumed that the EMCC operators can do the necessary dispatching work while waiting for an ambulance to become available. The simulated handling time t_H is expressed by the following equation where t_{H_M} is the historic median time and t_Q is the queue time:

$$t_H = \max(t_{H_M}, t_Q) \tag{5.2}$$

5.1.3.2 Dispatch Time

Dispatch time is the time it takes from when an ambulance is notified about an incident to when it starts to move from its current location. In the study conducted by Bekkevold and Schjølberg (2022), the developed simulation used historic dispatch times for each incident, utilizing the Dispatch time presented in Table 2.2.1. Unfortunately, the dispatch times vary a lot, even between incidents with the same urgency, which creates similar problems as with the handling time since the cause of the variation is unknown. An ambulance could for example

already be on the road when notified of an incident, or it could be parked at the base station with the ambulance personnel needing time to get equipped. This makes it unreasonable to use the historic dispatch time as part of the simulation.

The new simulation uses the median dispatch times corresponding to the different incident urgencies shown in Table 5.1.2. These times are only used in situations when the ambulance is located at a base station, so short dispatch times are not included in the calculation of the median because it is assumed that the short times were for ambulances already on the road. When such an ambulance is dispatched, it is assumed a dispatch time of 60 seconds for the ambulance personnel to process and plan the response mission. The dispatch time is presented in the following equation where t_{D_M} is the historic median time:

$$t_D = \begin{cases} t_{D_M}, & \text{ambulance is at base station} \\ 60, & \text{otherwise} \end{cases} \tag{5.3}$$

Urgency	**Acute**	**Urgent**	**Regular**
t_{D_M}	1m 28s	2m 2s	3m 43s

Table 5.1.2: Historic median dispatch times t_D for incidents in *incidents-processed* with dispatch time longer than one minute.

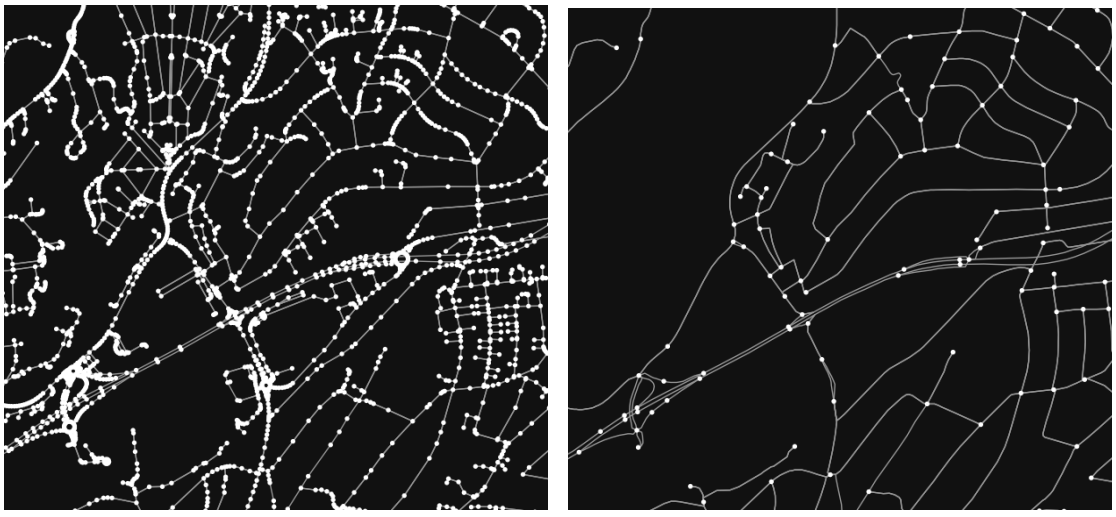
5.1.3.3 Travel Time

Travel time is usually the most influential part of the response time, and is the time it takes for an ambulance to reach the scene of the incident from its current location. In addition to being a part of the response time calculation, travel times are used to simulate how the ambulances travel in all other parts of the system, for example when an ambulance returns to its base station. It is therefore critical to have an accurate and robust calculation of travel time.

Bekkevold and Schjølberg (2022) implemented a calculation using a third party software called Ferd, created by Norkart (2023). This implementation consists of an origin-destination distance matrix between grids created with this software, which could be efficiently used in the simulation. There was however some distances that were lacking, so some assumptions had to be made to create a complete matrix. Additionally, this implementation did not give any information about the route that the ambulance takes, which can be very useful, for example for dispatching available ambulances that are returning to their base station. The general lack of control with the travel time implementation motivates a new method for simulation.

The main goal of the simulation is to evaluate the system by analyzing the resulting response times under various allocation scenarios. The optimization method, further described in Section 5.4, employs a GA to explore a substantial number of allocations. Consequently, the simulation model shown in the overview in Figure 5.0.1 must be run many times to evaluate all the allocations explored during this process, which necessitates fast travel time calculation. For each simulation run

using *incidents-processed-simulation*, the travel time between various points are needed approximately 100,000 times. A pre-calculated distance matrix, similar to the previous implementation, and argued for in Van Barneveld et al. (2018), is therefore preferable. For more control, a distance matrix was created using OSM, similar to what was done in Ridler, Andrew J. Mason, and Raith (2022). OSM provides geographic data of roads and junctions in a directed network, which includes information like type of road and speed limits. With this network, which can be seen in Figure 5.1.1a, it is possible to retrieve the fastest path between two coordinate points. The speed limits for the different types of roads was adjusted to more accurately represent real travel times. However, the fact that ambulances responding to acute incidents often exceed the speed limit is not accounted for, which creates an important limitation.



(a) All edges and nodes **(b)** Simplified

Figure 5.1.1: OSM network of roads near Ullevål Stadion in Oslo.

The network is a fully connected graph, so all coordinates are valid to use as both origin and destination. Initially nodes in the graph are positioned at all points where a road changes direction, such that a curvy road is split into several edges. To speed up the calculations a few simplifications of the graph was done, mainly removing intermediate nodes and other information not needed for the calculations. The simplified network is shown in 5.1.1b. As mentioned in Section 2.2, the dataset of incidents are distributed on a grid of coordinates, which in most cases does not have the same coordinate as any node in the OSM graph. A mapping of grid coordinates to the corresponding closest node was therefore created. The matrix was then built by using OSM to calculate the travel time between all combinations from one grid coordinate to another.

In addition to saving the travel time between two coordinates, the new implementation made it possible to save the route of the fastest path. To avoid an unnecessarily large matrix, the path coordinates were only saved for every 5 minutes of travelling along the path. This time interval is what is currently used for updating the ambulance positions in the simulation. This new combined matrix

which includes both travel time and the route coordinates required some adjustment in the simulation, but it replaced an implementation which caused some inaccuracies with the calculated path. The main part of the *matrix* creation is presented in Algorithm 2.

Algorithm 2 Travel Time Path Matrix

```
1: Input: OSM graph G, grid ids I, grid to nearest node map M
2: Output: dictionary of origin-destination pairs matrix
3: function FindPaths(G, I, M)
4:     matrix ← ∅
5:     for each grid_a ∈ I do
6:         for each grid_b ∈ I do
7:             if grid_a = grid_b then
8:                 matrix[grid_a][grid_b] ← {"time" : 60, "route" : []}
9:             end if
10:            node_a ← M.get(grid_a)
11:            node_b ← M.get(grid_b)
12:            routeNodes ← G.shortestPath(node_a, node_b)
13:            time, routeGrids ← getRouteInfo(routeNodes)
14:            matrix[grid_a][grid_b] ← {"time" : time, "route" : routeGrids}
15:        end for
16:    end for
17:    return dictionary of origin-destination pairs matrix
18: end function
```

With this travel time matrix, the travel time t_T for an ambulance a to an incident i can be expressed as the following equation:

$$t_T = matrix[grid_a][grid_i].time \tag{5.4}$$

5.1.4 Abort Incident Event

Two of the features shown in Table 2.2.1 are Arrival time and Departure time, which informs when and how long an ambulance was present at the incident. However, some of the incidents lack these timestamps, which leads to an assumption that the response to these incident was aborted. This could happen when the EMCC receives additional information that deems medical assistance unnecessary. In such cases, all dispatched ambulances, which may have already started moving towards the incident, are set to available and to return to their base station.

Although it is not shown in Algorithm 1, this event is created in the New Call event if the relevant timestamps are detected missing from the incident. The Available time shown in Table 2.2.1 will always be present however, which is used to determine when the Abort Incident event should happen, which notifies the dispatched ambulances to abort the incident. The Abort Incident event is shown on Line 17 in Algorithm 1.

5.1.5 Scene Events

When Arrival time and Departure time are present for an incident, the Scene Arrival event is created using the response time explained in Section 5.1.3. The duration that an ambulance was present at the incident is then used to create the Scene Departure event, shown on Line 24 in Algorithm 1. If the incident required patients to be transported to a hospital, a Hospital Departure event is created. Otherwise, the ambulances are set as available and told to return to their base station.

5.1.6 Hospital Time

Hospital time is the duration it takes from when an ambulance arrives at a hospital to when it is ready to leave. This time is used to offload the patient at the hospital and potentially assist the hospital personnel.

The historical hospital time for each incident could be used, since both Hospital time and Available time is present in the dataset as shown in Table 2.2.1. In contrast to handling time and dispatch time, it is assumed that hospital time is not dependent on the state of the EMS system in a significant way. The historical hospital time is therefore used.

To determine the time until an ambulance becomes available after offloading a patient t_A, the travel time to the hospital t_{T_H} and the hospital time t_P is used, as shown in Equation 5.5. This time is used when creating the Hospital Departure event on Line 35 in Algorithm 1.

$$t_A = t_{T_H} + t_P \tag{5.5}$$

5.1.7 Simulation Accuracy

The simulation proposed in this thesis focuses on being a realistic representation of how the EMS system would react to incidents, in different states. Since the state of the simulation is rarely the same as it was for the real EMS system at the time, historic handling times and dispatch times have been replaced with simulated times. This has caused the resulting response times from the simulation to deviate further from the historic response times. Therefore, a comparison between the two is less informative of the simulation accuracy. However, figure 5.1.2 shows the historic response times compared to the simulated response times, which still contains some correlation. The figure has been limited to only show response times below 100 minutes to enable comparison of the two sets of response times.

The simulated response times was created by a simulation that was run on an allocation called *PopulationProportionate*. This allocation was created in Bekkevold and Schjølberg (2022), using the population of the different base station areas to determine how many ambulances should be at each station. Although not an optimal allocation, as will be shown in Section 5.4, the allocation was deemed a viable and realistic allocation for doing experiments with.

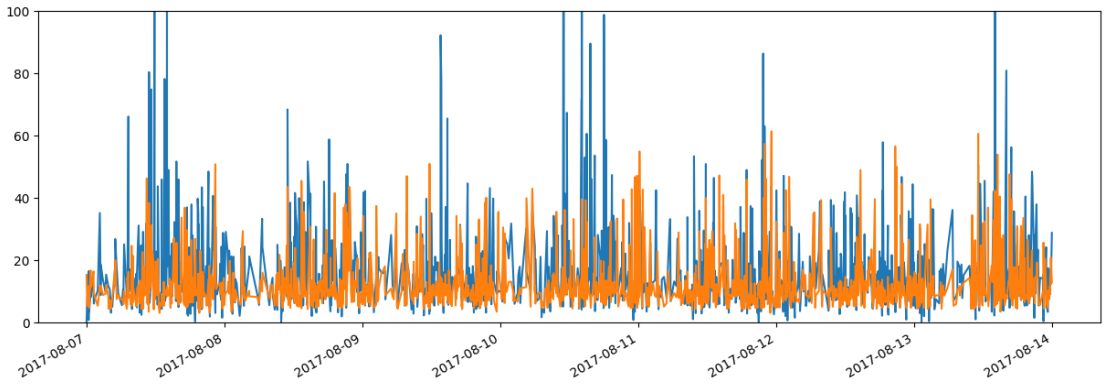


Figure 5.1.2: Historic (blue) and simulated (orange) response times in minutes for acute and urgent incidents in *incidents-simulation*.

In addition to observing simulated response times, a visualization of the simulation was developed by Bekkevold and Schjølberg (2022), similarly to Ridler, Andrew J. Mason, and Raith (2022), for further validation that the simulated EMS system behaves correctly. This visualization has been improved to enable easier debugging and in general give a more clear representation of the system. The visualization shows where incidents occur and how the ambulances move between base stations, incident locations, and hospitals when responding to the incidents.

5.2 Goal 2: Explore Dispatch Strategies

When the EMCC receives a call regarding an incident, they are faced with the decision of which ambulance to dispatch. Typically, the closest available ambulance in terms of travel time to the incident is dispatched. This strategy will in this thesis be called *Fastest*. However, there are situations where this may not be the optimal strategy Bandara, Mayorga, and Albert 2012. This section will first present two new dispatching enhancements that increases the number of ambulances available for dispatch, before exploring three coverage-based dispatch strategies for choosing which ambulance to dispatch. These enhancements and strategies will then be evaluated by the response times from running simulations. All simulations in this section is run on the *PopulationProportionate* allocation.

5.2.1 Dispatching Enhancements

In the simulation presented in Bekkevold and Schjølberg (2022), an enhancement was implemented to better represent the options that the real EMCC has. This enhancement involves making ambulances that travel back to their base station available for dispatching, as present in Zaffar et al. (2016) and McCormack and Coates (2015). For this to be possible in the simulation, the location of the ambulances has to be updated along the way back to the base station. As mentioned in 5.1.1, the simulation in this thesis updates ambulance position while travelling to any location, which was done in order to enable the enhancements presented in this section.

5.2.1.1 Reassigning Ambulances

The first enhancement is to be able to dispatch ambulances that are already on their way to another incident, which is presented in Ridler, Andrew J. Mason, and Raith (2022). The idea being that if an acute incident is called in it should have higher priority, so an ambulance that is on its way to a less urgent incident can be reassigned if it is the closest ambulance. Another available ambulance will then be dispatched for the less urgent incident. This strategy is somewhat complex since there are two ambulances and two incidents involved in the operation, and the response time calculation has to use the right pair of ambulance and incident. Some limitations to when an ambulance can be reassigned are presented:

- Only reassign ambulance if its current incident is of lower urgency than the new one
- Do not reassign ambulance if it has arrived at its current incident
- Only reassign ambulance if it is the only ambulance responding to its current incident (for simplification of the implementation)

Figure 5.2.1 shows a situation where the *Reassigning* enhancement is utilized. Here, the ambulance is travelling to incident 2, which is a regular incident, when a new acute incident occurs nearby to the ambulance. Since this ambulance is the closest ambulance to incident 1, and none of the limitations are in effect, the ambulance is reassigned from incident 2 to incident 1.

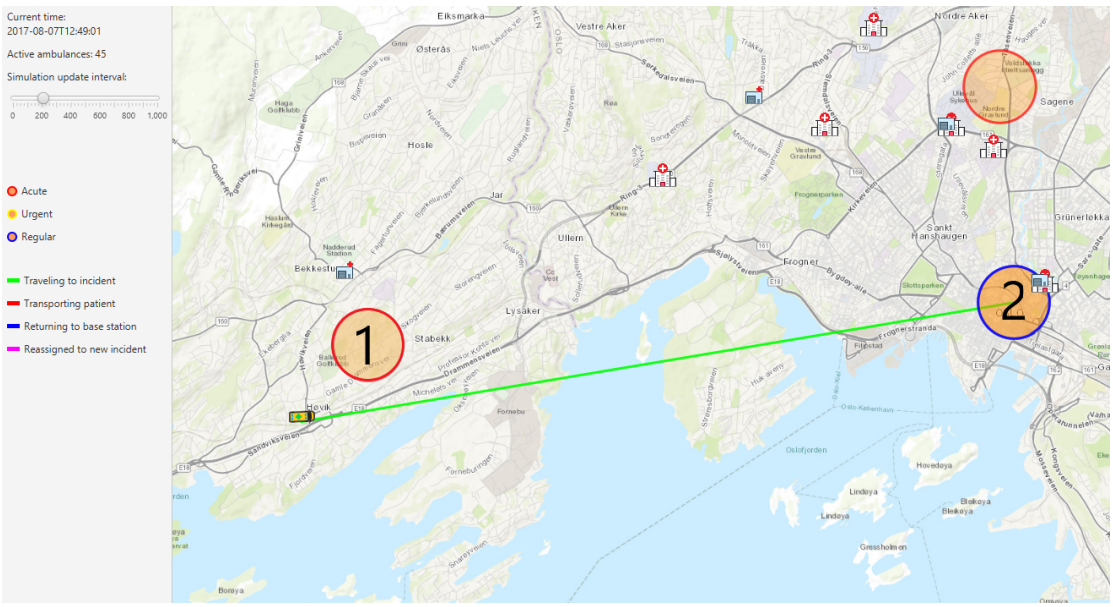


Figure 5.2.1: A situation where the *Reassigning* enhancement is utilized. Circles mark incidents, where incident 1 is the new acute urgency that the ambulance will be reassigned to, and incident 2 is the regular urgency that the ambulance is currently travelling to. The unnumbered incident is not a part of the reassigning procedure in this case.

5.2.1.2 Queuing Incidents

The second enhancement is the introduction of a queuing mechanism for ambulances currently transporting a patient to a hospital, or currently at the hospital. The *Queuing* enhancement enables these ambulances to be available for dispatching by allowing each ambulance to have a queue with a capacity of one incident in addition to the ongoing patient transport. When a new incident occurs, it can be added to this queue. Once an ambulance completes its current patient transport, it can respond to the incident that was added to its queue. Unlike the *Reassigning* enhancement, there are no limitations based on urgency, so all incidents can be queued. This enhancement is used in Van Barneveld et al. (2018).

When a new incident occurs, an ambulance eligible for queuing must reach the new incident before any other available ambulance in order to be selected. This means that t_A as shown in Equation 5.5 and the subsequent travel time t_T to the new incident, must result in the shortest response time T_{R_Q} compared to the response time T_R of other available ambulances. This is represented by the following equation:

$$t_{R_Q} = t_A + t_T \tag{5.6}$$

The *Queuing* enhancement stops the possibility of an ambulance being dispatched to an incident far away, when a much closer ambulance is almost done with its current incident. This situation is shown in Figure 5.2.2, where the ambulance is transporting a patient to the hospital when a new incident occurs nearby.

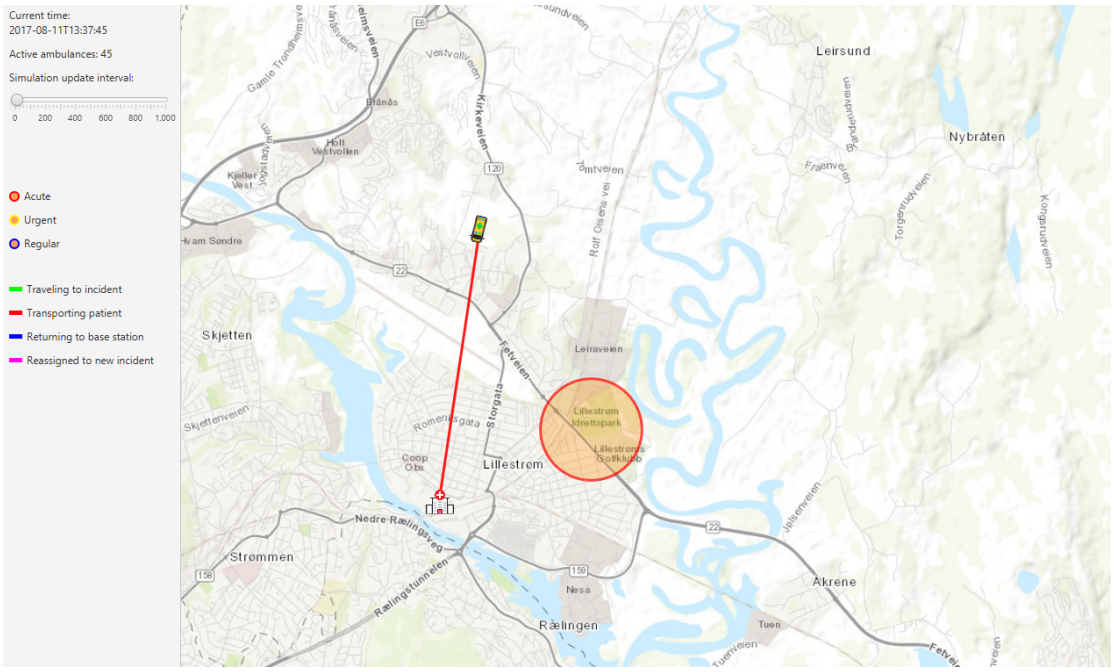
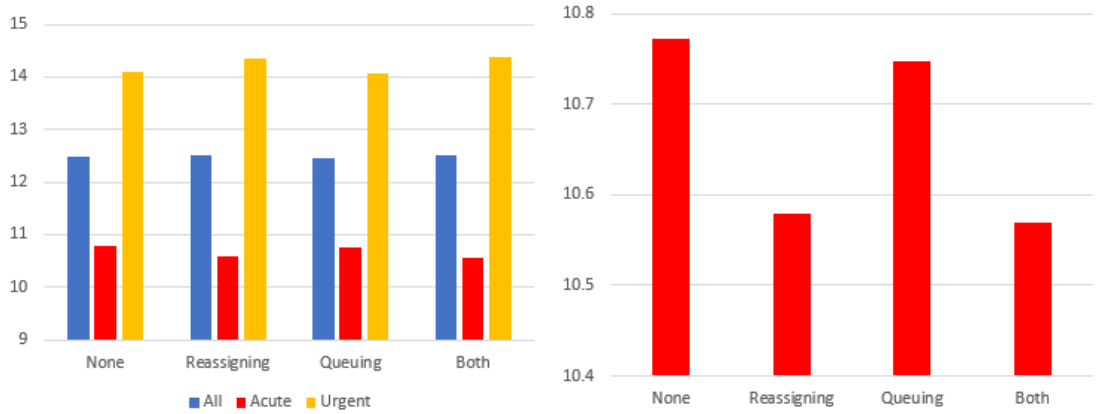


Figure 5.2.2: A situation where the *Queuing* enhancement is utilized. The ambulance is transporting a patient to the hospital indicated by the red line, and the circle marks the incident that the ambulance will respond to after offloading the patient.

5.2.1.3 Response Times

Both the *Reassigning* and the *Queuing* enhancements reflects how the real EMCC operates, and contribute to improving response times of acute incident. To observe the resulting effect that these enhancements have on response times, four simulations were run. The result is displayed in Figure 5.2.3, which shows how the enhancements impact the average response time of all, acute, and urgent incidents. The simulations used the *Fastest* dispatch strategy.

The results in Figure 5.2.3a show that reassigning ambulances improves response times of acute incidents, but response times of urgent incidents increases. This is because the *Reassigning* enhancement prioritizes acute incidents. Queuing incidents does however cause slightly shorter response times for both acute and urgent incidents. The combination of *Both* enhancements is deemed best since it has quickest average response time to acute incidents, as shown in Figure 5.2.3b, which should be given precedence. This combination was therefore decided to be used as the standard dispatch enhancement procedure, giving an average acute response time improvement of about 12 seconds.



(a) All incidents **(b)** Acute incidents

Figure 5.2.3: Average response times in minutes for each dispatch enhancement.

5.2.2 Coverage-Based Dispatch

Dispatching an ambulance to an incident reduces the coverage in the area previously occupied by that ambulance. As a result, response times to new incidents in that area might become longer. In such cases, it may be more effective to dispatch an ambulance from an area with greater coverage provided by other available ambulances, as explained in Ridler, Andrew J. Mason, and Raith (2022). This strategy ensures a more balanced distribution of resources and potentially shorter travel times to subsequent incidents.

In this thesis, coverage is a term to describe how many available ambulances are located within an area. Available ambulances are ambulances not currently assigned to an incident, or ambulances available because of the dispatch enhancements presented in Section 5.2.1.

As mentioned in Section 1.2, response times for acute incidents is a focus for the thesis. None of the three proposed strategies presented in Section 5.2.2.1, 5.2.2.2, and 5.2.2.3 will therefore be utilized in the case of acute incidents, since these incidents should be prioritized for fastest dispatch regardless of coverage status. To use the distinction between urgencies, a coverage importance value c_I is set to 0 for acute incidents, 1 for urgent incidents, and 2 for regular incidents. This will give a weight to coverage in relation to the response time, which will value faster response time the more urgent an incident is. For acute incidents, the closest ambulance will always be dispatched.

The dispatch strategies will use a penalty based on the coverage in the area of the ambulance a. This penalty $c_P(a)$ will together with $c_I(i)$ and response time $t_R(a, i)$ from the ambulance a to incident i, form a dispatch cost $d_C(a, i)$ that is used to sort the ambulances to find the best ones to dispatch. This calculation is shown in Equation 5.7, where a lower d_C indicates a more favorable ambulance to dispatch. These strategies incentivizes dispatching of ambulances from an area that is covered by other available ambulances.

$$d_C(a, i) = t_R(a, i) + c_P(a) \times c_I(i) \tag{5.7}$$

A problem with this coverage penalty c_P is determining its value, which affects the balance between the importance of coverage versus response time. An excessively large coverage penalty c_P will lead to more dispatching of ambulances far from the incident since it is more important to preserve coverage for future incidents, which can increase average response time. A small penalty leads to the coverage element being insignificant, and there might not be any available ambulances close to future incidents.

A gradual penalty depending on how many available ambulances A_a there are in an area was deemed advantageous. Additionally, only the number of available ambulances remaining A_r will be considered, which is the available ambulances when subtracting the incident demand i_d as shown in Equation 5.8. A simplification that only considers three cases of A_r was implemented: 0, 1, and 2 or more. All of the three dispatch strategies presented in Section 5.2.2.1, 5.2.2.2, and 5.2.2.3 have different *penalty values* to calculate the coverage penalty c_P depending on A_r.

$$A_r = max(0, A_a - i_d) \tag{5.8}$$

The penalty values in each strategy were found through experimentation, running hundreds of simulations with different sets of values and observing which values gives the best average acute response time. Since the simulations are only run using the *PopulationProportionate* allocation, the penalty values are most likely not optimal with other allocations. This is also the case for other incident sets than the *incidents-simulation* dataset. This method of finding penalty values could therefore be improved to increase generalization, but for the purpose of comparing the strategies the method is deemed satisfactory.

5.2.2.1 Base Station Coverage

One way to dispatch with hopes of not losing coverage is to give a coverage penalty c_P to ambulances that are assigned to a base station with few other available ambulances A_a also assigned to this base station. The calculation of the coverage penalty c_P for the *Base Station* strategy is presented in Algorithm 3.

Algorithm 3 Coverage Penalty Base Station

```
1: Input: Ambulance a, incident i, map of ambulances in base stations M
2: Output: Coverage penalty c_P
3: function CoveragePenalty(a, i, M)
4:     A_a ← countAvailable(M.get(a.BaseStation())
5:     A_r ← max(0, A_a − i.demand)
6:     c_P ← 0
7:     switch A_r do
8:         case 0
9:             c_P ← 1510
10:        case 1
11:            c_P ← 60
12:        case ≥ 2
13:            c_P ← 0
14:    return c_P
15: end function
```

One factor which can make this penalty a misrepresentation of coverage of an area, is that the available ambulances might not be located inside the base station responsibility area at the time. However, since the ambulances are available they are either travelling back to the base station, or they are they will start travelling quite soon. The only occasion when this is not the case, is when an ambulance is available because it can be reassigned.

5.2.2.2 Nearby Coverage

A second approach to coverage-based dispatching, is to give a coverage penalty c_P to the ambulance considered for dispatch a if it is located far from other available ambulances A_a. This penalty works similar to the penalty for the *Base Station* strategy, but instead of counting available ambulances for each base station, it counts available ambulances that are nearby in a specific range to the ambulance a considered for dispatch.

In contrast to base station coverage, the *Nearby* strategy has to take into account the position of all other available ambulances at the time. This made this strategy computationally slow, which is an important factor for optimization as discussed in 5.1.3.3. The range which determined whether an ambulance was nearby or not, was set to 7 minutes (7×60 seconds) of travel time, after experimenting the same way as with the penalty values. The calculation of the coverage penalty is presented in Algorithm 4.

Algorithm 4 Coverage Penalty Nearby

Input: Ambulance a, incident i, list of available ambulances V
Output: Coverage penalty c_P
function CoveragePenalty(a, i, V)
    $A_a \leftarrow 0$
    for each *ambulance* in V **do**
        $distance \leftarrow ambulance.timeTo(a)$
        if $distance < 7.0 \times 60$ **then**
            $A_a \leftarrow A_a + 1$
        end if
    end for
    $A_r \leftarrow \max(0, A_a - i.\text{demand})$
    $c_P \leftarrow 0$
    switch A_r **do**
        case 0
            $c_P \leftarrow 1590$
        case 1
            $c_P \leftarrow 55$
        case ≥ 2
            $c_P \leftarrow 0$
    return c_P
end function

5.2.2.3 Predicted Demand Coverage

General ambulance coverage can be an advantageous strategy, but giving equal importance to covering all the different areas might not be the best strategy. As seen in Section 2.3.3, some areas have a higher number of incidents than others. This leads to the idea of giving a higher coverage importance to these areas, since a prediction can be made that it is more likely that incidents occur there than in other areas. By considering future demand and the current coverage of the area, the EMCC can make informed decisions to optimize ambulance dispatch and improve overall emergency response efficiency.

As discussed in Section 2.3.3, the sparsity of the data in which predictions could be based on led to the generalization of grouping number of incidents in grids into base station areas. Although Van De Weijer and Owren (2022) explored using different areas for prediction, it was deemed best to use base station areas for implementation convenience. Additionally, since ambulances will eventually return to their base station, the base station areas can be beneficial compared to randomly placed areas.

Only acute and urgent incidents should be considered for predictions, so the *incidents-processed-predict* dataset was used for training the model. The incidents in the same period as *incidents-simulation* was extracted since this period is used for simulations. Since the data has important temporal trends, k-fold cross-validation was not chosen for validation as the temporal dependencies in the data can cause each fold to have significantly different characteristics, leading to

largely varying performance across the folds. Instead, the month of august, 2018 was extracted from the training data as validation data. The different models created during development was evaluated using the mean MSE across 10 training runs on the same validation data. The best performing model from experimenting with model configuration and input features was then chosen to make predictions in the period of *incidents-simulation* for use in the *Predicted Demand* dispatch strategy.

To enhance the prediction model's ability to understand the temporal and spatial incident demand, additional preprocessing was performed on the *incidents-processed-predict* dataset. The dataset represents single incidents, but an aggregation was done which involved counting the number of incidents that occurred within each hour and base station area. As a result, the dataset was modified to have one row for every hour and base station combination, with the corresponding count of incidents during that hour. This alteration of the dataset is only used for demand predictions. Figure 5.2.4 shows that the rows with 0 incidents are in majority.

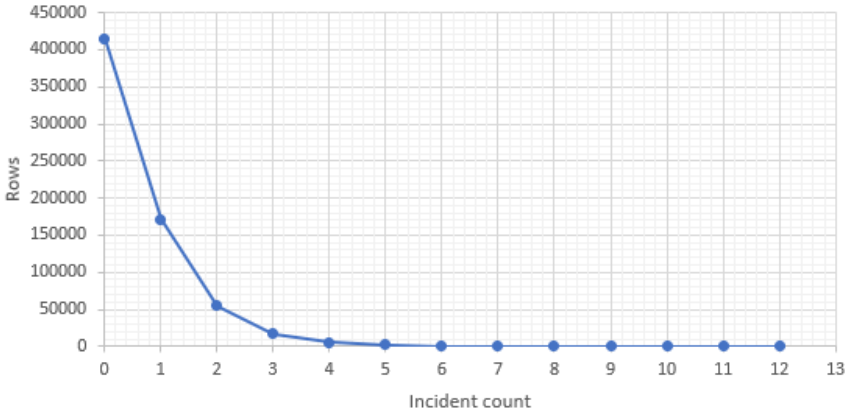


Figure 5.2.4: Number of rows with the different incident counts in the altered version of *incidents-processed-predict*.

The spatial feature represents the base station's ID, which corresponds to its responsible area. This information was obtained by extracting the coordinates of each incident and identifying the base station area to which it belongs. The temporal features were derived from the call time of the incidents. These features include Year, Month, Day, Week, Weekday, and Hour. In addition to these default features, feature extraction was done to help make the distinction between certain temporal aspects clearer for the model. Inspiration was taken from Chen et al. (2016), who introduced the Season feature and the Weekend feature. As discussed in 2.3.3, these trends are significant. The Season feature determines in which season it is, while the Weekend feature determines whether it is in a weekend or not. In addition to adding the Season and the Weekend features, a third feature was implemented. The Daytime feature categorizes the incident count into one of six 4-hour intervals, representing different parts of the day. The Daytime feature was introduced to help the model generalize across different time periods rather than focusing too much on specific hours.

Table 5.2.1 shows the mean MSE when training the model with different feature sets. Other feature sets were also experimented with, but the best feature set was found to be all of the features, except for Daytime. As seen in Section 2.3.3, many of the temporal features are important.

Feature Set	**MSE**
Default Features	0.5508
+ Season	0.5505
+ Season, Weekend	**0.5503**
+ Season, Weekend, Daytime	0.5507

Table 5.2.1: MSE for different feature sets.

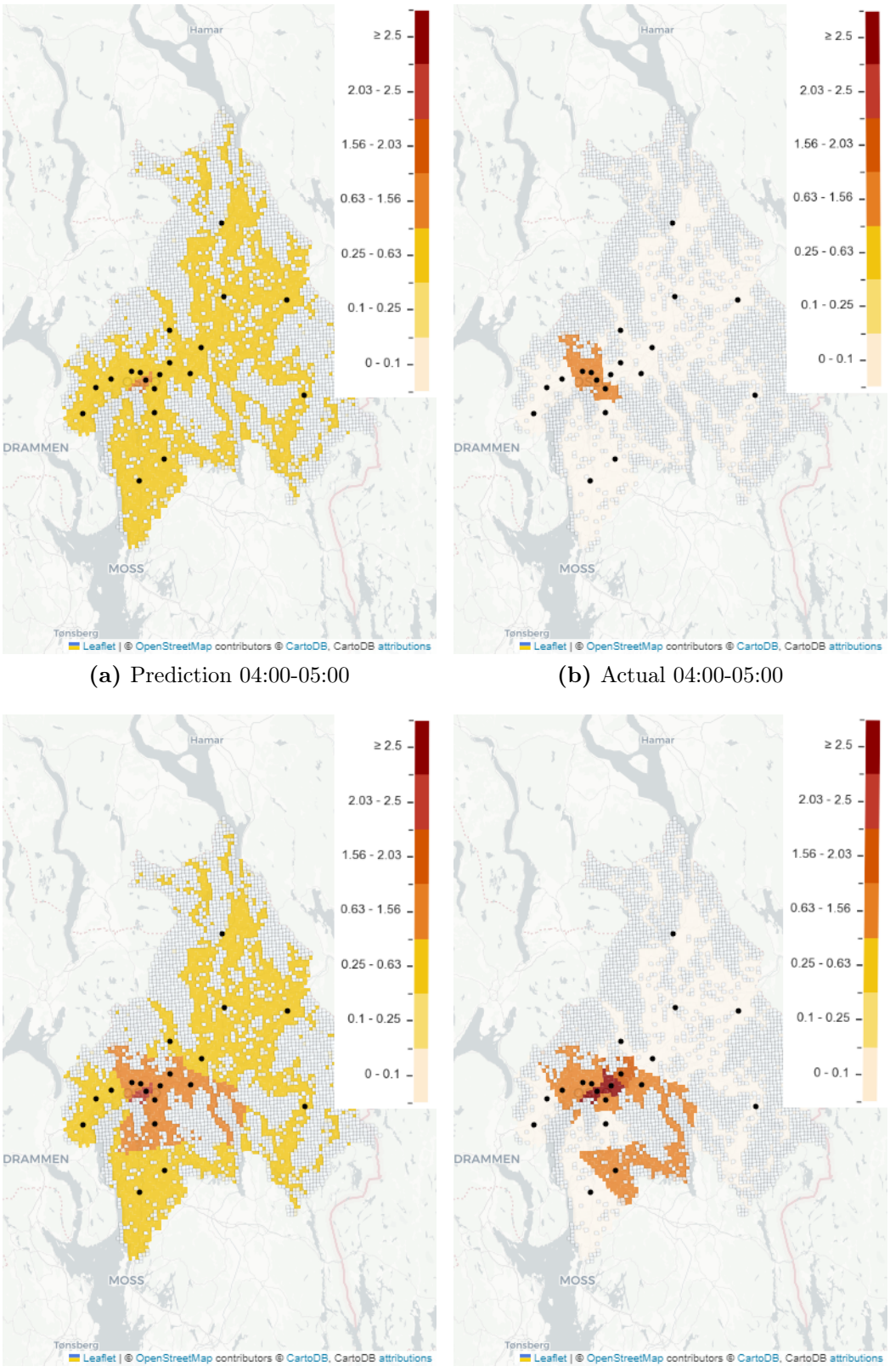
The model that was developed was inspired by one of the models presented in Van De Weijer and Owren (2022). Specifically, a regression model using a neural network was implemented, using temporal and spatial features as input, and incident count as output. The model consists of two hidden layers with 64 and 32 nodes for each layer, using the *Swish* activation function, and dropout layers between each. The *Adam* optimizer was chosen with a learning rate of 0.0005. Early stopping was utilized with a patience of 5 to reduce overfitting, using MSE on the validation data as monitoring metric. The model's configuration values were found through experimentation with different values, some of which are shown together with their respective result in Table 5.2.2.

Hidden layers	**MSE**
64	0.5524
64, 32	**0.5503**
128, 32, 8	0.5529
128, 128, 64, 8	0.5522

Table 5.2.2: MSE for different hidden layer configurations.

Instead of using a neural network, a statistical method was also experimented with. This was done to see whether these types of methods could compete with the neural network and be viable for predictions on such sparse data. Figure 5.2.4 shows that the data has a Poisson distribution, similarly to the emergency medical data in H. Huang et al. (2019), so a Poisson regression model was tested. The model achieved an MSE of 0.7462, which shows that it is able to capture some of the temporal and spatial features. However, it seems it may not be able to capture the more complex patterns, so the statistical methods were discarded from further consideration.

Since the neural network model is a regression model, it returns the count as a float value. Figure 5.2.5 shows the predicted demand for all base station areas in a given hour, compared to the actual count. The color range values in the figure is exponentially distributed between a predicted demand value of 0.1 and 2.5, in order to better see the difference between predictions. Evidently, the prediction is not completely accurate, but the main spatial and temporal trends are found which can be valuable for the dispatch strategy.



(a) Prediction 04:00-05:00

(b) Actual 04:00-05:00

(c) Prediction 19:00-20:00

(d) Actual 19:00-20:00

Figure 5.2.5: Predicted and actual incident counts for the different base station areas, for two different hours on 11.08.2017.

The predicted incident counts for each hour was then used in the *Predicted Demand* strategy. This was done by increasing the penalty based on how many incidents are predicted to occur in the next hour. The predicted demand was converted to an appropriate penalty scale with a static factor of 185. This value was optimized similar to what was done for the penalty values. The penalty values was also optimized specifically for this strategy. The coverage penalty calculation is presented in Algorithm 5.

Algorithm 5 Coverage Penalty Predicted Demand

Input: Ambulance a, incident i, map of ambulances in base stations M, predictions P
Output: Coverage penalty
function CoveragePenalty(a, i, M)
    $A_a \leftarrow$ countAvailable($M.get(A.\text{BaseStation}())$
    $A_r \leftarrow \max(0, A_a - i.\text{demand})$
    $c_P \leftarrow 0$
    switch A_r **do**
        case 0
            $c_P \leftarrow 1920$
        case 1
            $c_P \leftarrow 280$
        case ≥ 2
            $c_P \leftarrow 0$
    $predictedDemand \leftarrow P.get(A.\text{BaseStation}()).get(I.\text{time})$
    $c_P \leftarrow c_P + predictedDemand \times 185$
    return c_P
end function

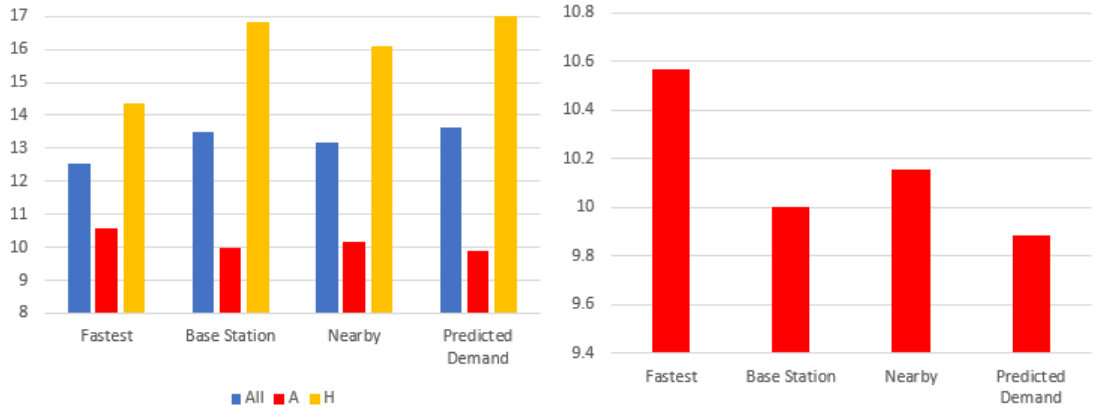
5.2.2.4 Response Times

Figure 5.2.6 shows how the three new dispatch strategies performed in terms of average response time compared to the *Fastest* strategy and to each other. Figure 5.2.6b shows that the *Predicted Demand* strategy is able to reach the lowest average acute response time. As seen in Figure 5.2.6a, the response times of urgent incidents increase when using these strategies however, which is an effect of prioritization of acute incidents. As presented in the strategy algorithms and in Table 5.2.3, the optimal penalty values are all quite high when considering that they represent seconds in travel time. This causes future coverage to be valued higher than response time of urgent incidents. Interestingly, the optimal penalty values for the *Predicted Demand* strategy are higher than the others, possibly because the information from the predictions is valuable.

A_r	**Base Station**	**Nearby**	**Predicted Demand**
0	1510	1590	1920
1	60	55	280

Table 5.2.3: Optimized penalty values for the coverage penalty c_P of the different dispatch strategies.

The average acute response time improvement is about 41 seconds, which is a significant improvement when considering the number of incidents in the dataset. A small average improvement can save lives. In any case, both the enhancement results in Figure 5.2.3 and the strategy results in Figure 5.2.6 shows that the simulation is able to model changes and give information about their effects.



(a) All incidents **(b)** Acute incidents

Figure 5.2.6: Results of each dispatch strategy in terms of average response time (a) and average acute response time (b).

5.3 Goal 3: Incident Urgency

The third research goal of this thesis is to explore the urgency aspect of incidents. This section will mainly explore the impact of improving the accuracy of assigning urgency to incidents. This will be done by running simulations with and without the improved accuracy, and observing the average response times of acute and urgent incidents. The effect that different dispatch enhancements and dispatch strategies has when the accuracy is improved will also be explored. The simulations will as in Section 5.2 also be run using the *PopulationProportionate* allocation.

5.3.1 Preset Urgency

As mentioned in Section 1.1, non-acute incidents are often assigned as acute as a precautionary measure which can lead to inefficient use of resources. In an ideal world, the EMCC operators would perfectly assign urgencies to incidents in a way that enables optimal efficiency in deployment of resources. Although this is not currently a realistic goal, this section will research the effect of having a better urgency assignment procedure. Investing in research on urgency assignment procedures like the machine learning and natural language processing methods in Ivanov et al. (2021), could give valuable results in terms of resource management and response time.

To simulate the urgency assigning accuracy, the actual urgency for all incidents would be useful to know. Unfortunately, this information is not present in the dataset provided by OUH. Instead of using real historic data about the incidents, a synthetic modification of the *incidents-simulation* dataset was created. In this

synthetic dataset, called *incidents-simulation-corrected*, a percentage of the acute incidents have been corrected to be urgent. As expanded upon in Section 2.3.1, 75% of the acute incidents should be changed to create the best case scenario where the urgency assignment is perfect. The set of 75% incidents that are corrected are randomly distributed in *incidents-simulation-corrected*, but the same incidents are always corrected in every simulation to enable comparison. It is assumed that no incidents were assigned a lower urgency than what it actually was, and no incidents are corrected from urgent to regular since this occurrence is assumed to be rare, seeing as many of the regular incidents are planned. A confusion matrix for the relation between assigned and actual urgency is shown in Figure 5.3.1.

		Actual	
		Acute	**Non-acute**
Assigned	**Acute**	25%	75%
	Non-acute	0%	100%

Figure 5.3.1: Assumed incident urgency confusion matrix showing the percentage relation between assigned and actual urgency.

5.3.1.1 Benchmark

As a benchmark, two simulations were run on *incidents-simulation* and *incidents-simulation-corrected* without using any of the new dispatch enhancements or dispatch strategies presented in Section 5.2. The comparison of the resulting average response times can be viewed Figure 5.3.2.

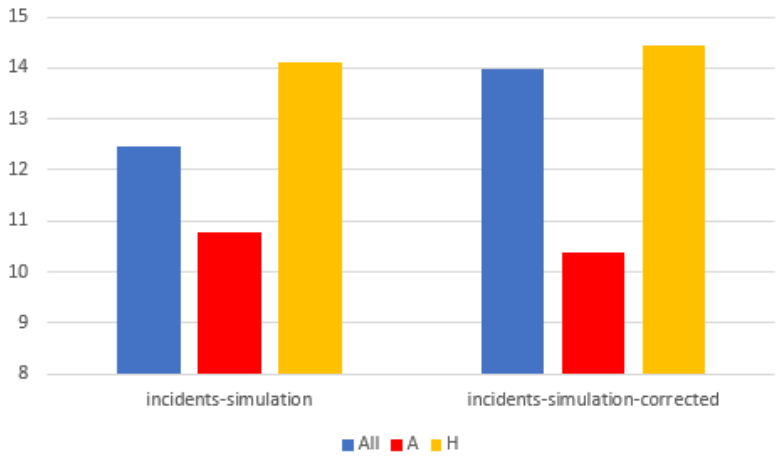


Figure 5.3.2: Average response times in minutes from simulations with no new dispatch enhancements and strategies, with and without urgencies correctly classified as acute (A) or urgent(H).

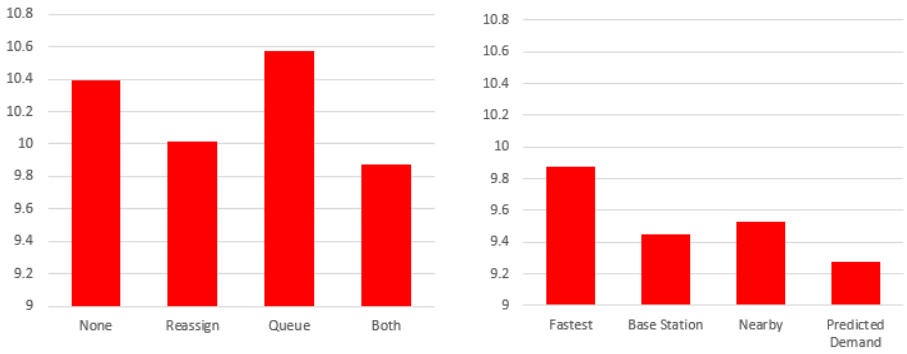
Even though no prioritization is being done in terms of dispatching, the simulation still uses historic median times for handling time and dispatch time. The median times for urgent incidents are slower than acute incidents as discussed in Section 5.1.3.1 and Section 5.1.3.2. Subsequently, the increase in overall response time for *incidents-simulation-corrected* compared to *incidents-simulation* is expected since there are many more urgent incidents. The variance in average acute and urgent response times is unknown, but it could possibly be a cause of randomly correcting urgencies of incidents that are located far from any base station or ambulance.

5.3.1.2 Dispatch Enhancements and Strategies

With the *incidents-simulation-corrected* dataset, simulations were run with the different dispatch enhancements and strategies to research the effect they have in a more ideal world of correct incident urgency diagnosis. The experiment results can be seen in Figure 5.3.3, where Figure 5.3.3a uses the *Fastest* dispatch strategy, and Figure 5.3.3b uses *Both* dispatch enhancements. Only the acute response times are included in the results since it is the most important metric, in order to better differentiate between the strategies. The penalty values for the dispatch strategies were optimized again specifically for this set of synthetic incidents, and are presented in Table 5.3.1.

A_r	**Base Station**	**Nearby**	**Predicted Demand**
0	1315	1780	1590
1	310	255	20

Table 5.3.1: Optimized penalty values for the coverage penalty c_P of the different dispatch strategies on *incidents-simulation-corrected*.



(a) Dispatch enhancements **(b)** Dispatch strategies

Figure 5.3.3: Average response time for acute incidents in minutes when simulating on *incidents-simulation-corrected* using new dispatch enhancements and strategies.

The results in Figure 5.3.3 show that the dispatch enhancements and dispatch strategies contribute to better response time for the reduced number of acute incidents. When most of the incidents are urgent, it gives the EMCC more options to preserve coverage so that an ambulance is more likely to be nearby and ready for new acute incidents.

For *incidents-simulation*, the average acute response time went from 10m 46s to 9m 53s when utilizing the *Predicted Demand* strategy, which is a difference of 53 seconds. In comparison, average acute response time for *incidents-simulation-corrected* went from 10m 23s to 9m 16s when utilizing the Predicted Demand strategy, which is a difference of 67 seconds. The improvement on *incidents-simulation-corrected* is significantly bigger, which further motivates efforts to improve the accuracy of urgency assignment. Additionally, it is assumed that as average response times decrease, it becomes increasingly more challenging to further reduce them, as there is a natural lower limit.

5.3.2 Survivability

For the previous experiments, the simulated EMS system has been evaluated using incident response times, especially response times of acute incidents. As mentioned in Section 5.2.2.4, the dispatch strategies prioritizes acute incidents, which in addition to the *Reassigning* enhancement, increases urgent response times. Although acute incidents should be prioritized, urgent incidents should not be completely excluded from evaluation of the EMS system.

One method to include urgent incidents in the evaluation, is to utilize survival functions. In the domain of ambulance location problems, a survival function is a function that gives a measure of the survivability of the patient based on the response time to the incident. Since acute incidents require medical care in a more time-critical manner than less urgent incidents, a heterogeneous approach of using two different survival functions for acute and urgent incidents was implemented. It is natural to give more weight to slow response time to acute incidents than to urgent incidents.

Detailed information about each incident would enable the use of more fine tuned survival functions fit for the specific incident illnesses. Unfortunately, the dataset from OUH does not contain this information. An option for survivability approximation is to have more general survival functions. Equation 5.9 shows a general survival function presented in Bekkevold and Schjølberg (2022), which followed the work done by Amorim, Ferreira, and Couto (2019). Here the c_k coefficient influences the starting point or initial level of survival probability for minimal response times, while m_k determines the rate of change in survival probability with respect to the response time variable r.

$$s_k(r) = (1 + e^{c_k + m_k * r})^{-1} \tag{5.9}$$

From this general survival function, two urgency specific survival functions are suggested. Equation 5.10 shows the survival function used for acute incidents, which in Knight, Harper, and Smith (2012) is meant specifically for cardiac arrests. Although not all acute incidents are cardiac arrests or similarly time-critical incidents, these coefficients are considered satisfactory for representing the importance of fast response times for all acute incidents. Equation 5.11 is used for urgent incidents, which in contrast to the acute survival function does not use coefficients

approximated from data. The urgent survival function is rather an attempt to create a reasonably realistic survival function by relaxing the acute survival function. For this thesis, the survivability works more as a scoring system than an actual representation of probability of survival. The behaviour of the survivability score related to response time for acute and urgent incidents are presented in Figure 5.3.4.

$$s_a(r) = (1 + e^{-0.26+0.139r})^{-1} \tag{5.10}$$

$$s_u(r) = (1 + e^{-4+0.05r})^{-1} \tag{5.11}$$

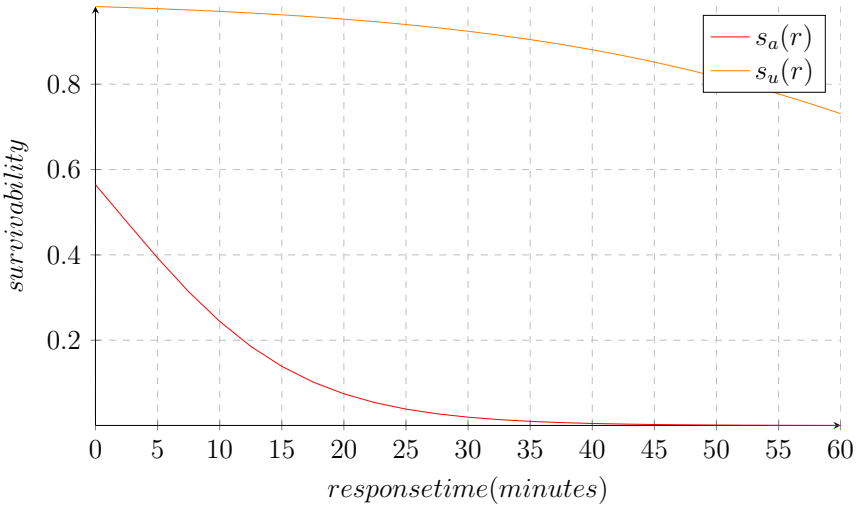


Figure 5.3.4: Survival functions for acute and urgent incidents. Acute incident survivability drops instantly when response time increases.

The dispatch strategies presented in Section 5.2 can be evaluated using survivability as metric of performance instead of just the average acute response time. Figure 5.3.5 shows the dispatch strategies now evaluated with the survival functions presented above. The *Predicted Demand* strategy performs marginally better than the other strategies.

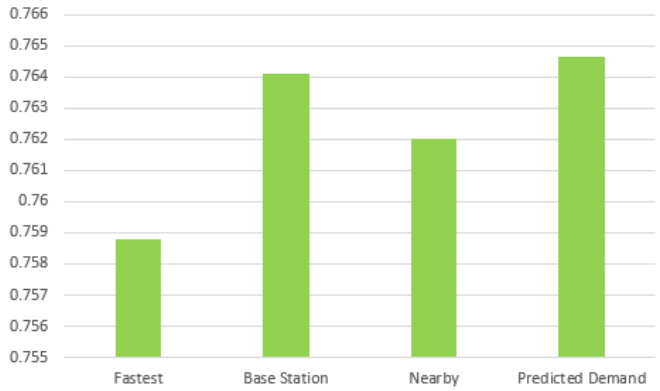


Figure 5.3.5: Average survivability of incidents when simulating on *incidents-simulation* using different dispatch strategies.

The penalty values for the dispatch strategies have here been optimized to increase survivability instead of average acute response time. This causes the optimal penalty values to be lower, since survivability is affected by response times to both acute and urgent incidents. Lower penalty values means that urgent incidents are more likely to be responded to by the closest ambulance. The penalty values are presented in Table 5.3.2.

remainingAvailable	**Base Station**	**Nearby**	**Predicted Demand**
0	1050	690	720
1	50	210	220

Table 5.3.2: Optimized penalty values for the dispatch strategies when optimizing for survivability.

One important thing to note is that the survivability is not linear, so even if average acute response time increases, the survivability of acute incidents might also increase. Additionally, both acute and urgent incidents are considered which makes the relation between average response time and survivability more complicated.

5.4 Goal 4: Optimization

The fourth research goal of this thesis is to reduce ambulance response times to incidents by optimizing the allocation of ambulances. An optimal allocation makes sure that high-activity areas are covered by enough ambulances to quickly respond to incidents. This section will present the GA developed in Bekkevold and Schjølberg (2022) and the improvements done in this thesis to this optimization method. A multi-objective approach will also be explored, trying to optimize both acute and urgent response times. Finally, the results of the improvements will be presented and discussed.

To compare the different optimization methods and improvements, it was decided to set a total maximum optimization time of 40 minutes. This will enable comparisons of methods that are computationally different in terms of running time. To mitigate the impact of randomness in the stochastic methods, the allocated 40-minute running time is divided into 10 separate optimization runs, each lasting 4 minutes. This division ensures that no single method is evaluated more favorably than another due to random factors alone. By conducting multiple runs for each method, the influence of chance is minimized, allowing for a more reliable and robust comparison between the methods.

For all optimization methods in this section, *Both* dispatch enhancements are enabled and the *Demand Prediction* dispatch strategy is used.

5.4.1 GA

Bekkevold and Schjølberg (2022) presented a GA as an optimization method for ambulance allocation. They also explored using a local search algorithm called Stochastic Local Search, and a hybrid GA called Memetic Algorithm. However,

the results obtained from these alternative approaches were not as promising as those achieved with the GA. Consequently, further development of Stochastic Local Search and Memetic Algorithm was not pursued in this thesis. Pseudo code for a generic GA is presented in Algorithm 6.

Algorithm 6 Genetic Algorithm

```
Initialize population
Evaluate fitness of solutions
while Termination condition not met do
    Select parents for reproduction
    Do crossover to create offspring
    Mutate offspring
    Evaluate fitness of new solutions
    Select solutions for next generation
end while
return Best solution
```

The GA and the configurations presented in Bekkevold and Schjølberg (2022) is used as a baseline method for new methods and improvements. This *BaselineGA* optimization method was developed to find good allocations for a slightly different simulation, since changes has since been done to the simulation as described in Section 5.1. Importantly, it also uses total average response time as fitness function for optimization. The configuration of the *BaselineGA* method is presented in Table 5.4.1. The total number of generations in this configuration shows approximately how many generations the method reaches in 4 minutes of run time.

Parameter	**Value**
Fitness	Response Time
Initializer	Random
Elite Size	4
Generations	1600
Population Size	30
Tournament Size	5
Crossover Probability	0.2
Mutation Probability	0.05

Table 5.4.1: Configuration parameters for the *BaselineGA* optimization method.

The optimization results of the *BaselineGA* method is presented in Table 5.4.2. The Best row in all the result tables in this section refers to the run with the best fitness, while the Average row refers to the average of the 10 different optimization runs. As mentioned in 5.3.2, the relation between survivability and average response times is not linear, which can be observed in some of the result tables. The relation is even more obscured since some of the results are averages of several runs.

Figure 5.4.1 presents the fitness values per generation for the best run, showcasing the progression of the algorithm over time. This graph shows that the algorithm quickly finds good solutions, before slowly trying to reach optimal solutions.

	Response Time			Survivability
	All	Acute	Urgent	
Best	11m 56s	9m 41s	14m 5s	0.7680
Average	**12m 0s**	9m 40s	14m 16s	0.7680

Table 5.4.2: Optimization results from running the *BaselineGA* method.

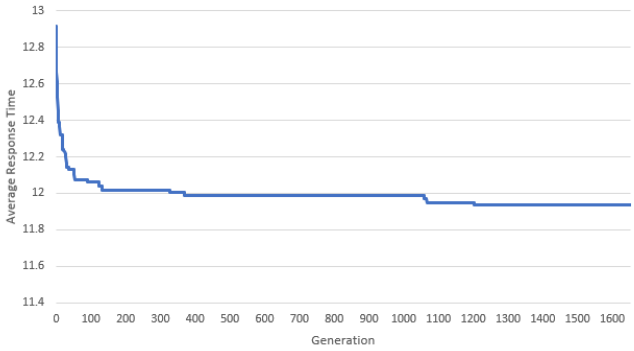


Figure 5.4.1: The best fitness value (average response time in minutes) in the population for each generation of the best run of the *BaselineGA* method.

5.4.1.1 Fitness Function

The fitness function is a crucial component of the GA, responsible for evaluating each allocation solution in the population. For ambulance allocation optimization, the fitness function is based on the response times returned from the simulation of the EMS system when a specific allocation is used, as mentioned in Section 5.4. Bekkevold and Schjølberg (2022) used the average response time of all incidents as fitness for the solutions.

Since the average response time of all incidents is used as a fitness function, the acute and urgent response times are valued the same. Some allocations might be better suited to respond to acute incidents, while others are better for urgent incidents. Therefore, it is considered more interesting to use a fitness function that gives more weight to acute response times. The survivability score presented in Section 5.3.2 does indeed provide such a weighting, so it was decided to use this as fitness function for the rest of the optimization methods in this section. Table 5.4.3 shows the results of running the *BaselineGA* method, but now using the survival functions for fitness when optimizing. This optimization method is referred to as the *SurvivabilityGA* method.

	Response Time			Survivability
	All	Acute	Urgent	
Best	12m 12s	9m 28s	14m 49s	0.7736
Average	12m 18s	9m 34s	14m 54s	**0.7727**

Table 5.4.3: Optimization results from running the *SurvivabilityGA* method.

5.4.1.2 Genotype

The genotype serves as a representation of the solution, facilitating the implementation of essential genetic operators during the search process. In this thesis, the genotype is comprised of two allocations: one for the daytime shift and another for the nighttime shift. Each allocation is represented as a list of numbers, where each number corresponds to a specific ambulance and its value signifies the ID of the base station to which that ambulance is assigned. Equation 5.12 defines one shift allocation A with n number of ambulances available for that shift. The set of base station IDs is denoted as B, and a_i represents ambulance a with ambulance ID i which can have any value in B.

$$A = (a_1, a_2, \ldots, a_n), \quad a_i \in B \tag{5.12}$$

The complete genotype for the solution consists of the daytime shift allocation A_{day} and the nighttime shift allocation A_{night}. For the daytime shift n is 45 while for the nighttime shift n is 29, as mentioned in Section 2.1.

Three different approaches for generating allocations were chosen from the ones presented in Bekkevold and Schjølberg (2022) to experiment with in terms of initialization. Instead of relying on a single approach, a proposed strategy is to randomly select an approach per allocation in the initial population. This was done to try to include solutions that are considered favorable starting points for the algorithm, but still creating a diverse range of initial solutions. This initialization approach is called *Mix*, and consists of the following approaches:

- ***Random***: Ambulances are randomly assigned to base stations. Selected with a probability of 80%.
- ***PopulationProportionate***: Ambulances are assigned to base stations so that the number of ambulances per base station correlate to the population of the base station area. Selected with a probability of 10%.
- ***UniformRandom***: Ambulances are assigned to base stations so that they are evenly distributed across all base stations, with the rest of the ambulances assigned randomly. Selected with a probability of 10%.

Solutions generated from the three different approaches exhibit distinct characteristics, which might offer a broader exploration of the problem space. This diversity can be advantageous for the GA, as it increases the chances of discovering promising solutions early. The *MixGA* optimization method uses the *Mix* initialization instead of only the *Random* initialization in the *Baseline* method. The result of the *MixGA* method is presented in Table 5.4.4, which shows that the *Mix* initializer performs similarly to the *Random* initializer, but is slightly worse. The difference can still be a factor of the stochastic nature of the optimization method, but one reason can be that the *PopulationProportionate* and *UniformRandom* methods are not good starting points and only decrease the diversity of the starting population. The *Random* initializer was therefore chosen for the implementation in this thesis and for the rest of the optimization methods in this section.

	Response Time			Survivability
	All	Acute	Urgent	
Best	12m 6s	9m 24s	14m 41s	0.7741
Average	12m 13s	9m 35s	14m 49s	**0.7726**

Table 5.4.4: Optimization results from running the *MixGA* method.

5.4.1.3 Parent Selection

A main aspect of the GA is to explore new solutions based on current good solutions. The process of selecting which current solutions to modify is called parent selection. Tournament selection was implemented in this thesis, which enables adjustment of selection pressure. The parameter for adjusting selection pressure is discussed further in Section 5.4.1.6.

5.4.1.4 Genetic Operators

The exploration of new solutions is done through genetic operators which alter the genotypes of solutions to create new ones. The first genetic operator is crossover which combines two solutions or parents to make offspring. Since the genotype consists of two allocations, the crossover operation combines the dayshift allocations and the nightshift allocations separately. Figure 5.4.2 shows an example of a crossover operation on two different allocations. The crossover operation splits the allocations at the same random crossover point and swaps the last part of both allocations.

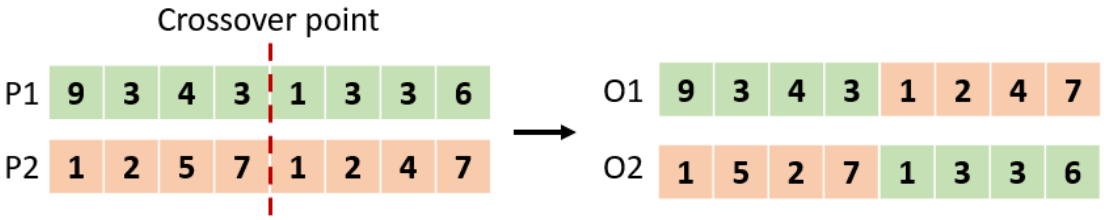


Figure 5.4.2: Example of crossover operation creating dayshift allocations for offspring O1 and O2 from combining the dayshift allocations of parents P1 and P2. The number of dayshift ambulances is only 8 in this example.

The base station IDs are not sorted in the allocations so the crossover operation will likely not keep much of the information about how many ambulances are assigned to a base station. Figure 5.4.2 shows that base station 3 has assigned four ambulances in the allocation for P1 and that the P2 allocation has two assigned ambulances for base station 1, 2, and 7. Neither of the offspring O1 and O2 contains the information that base station 2 and 7 had more assigned ambulances than average in the parent allocations, and the especially high number of ambulances assigned to base station 3 in P1 is not kept either. This shows that the crossover operation for this problem is quite destructive, meaning that it explores solutions far from the parents, instead of exploiting the found knowledge. Subsequently, a low probability of doing crossover is deemed favorable.

Since the parent solutions might be good solutions, it could be advantageous to search the solutions that are similar to the parents. A slightly altered approach to the crossover operation is therefore proposed, which is to sort the base station IDs in the allocations. The idea being that sorted allocations will be able to preserve more of the information gathered in the parents. Figure 5.4.3 shows that this approach would achieve this exploitative goal for this specific case. However, the results of using this *SortedGA* method shows that this approach is not an improvement. The reason for this could be that sorting the allocations reduce the diversity of the population. Non-sorted crossover is therefore implemented in this thesis and utilized for the rest of the optimization methods in this section. The results are presented in Table 5.4.5.

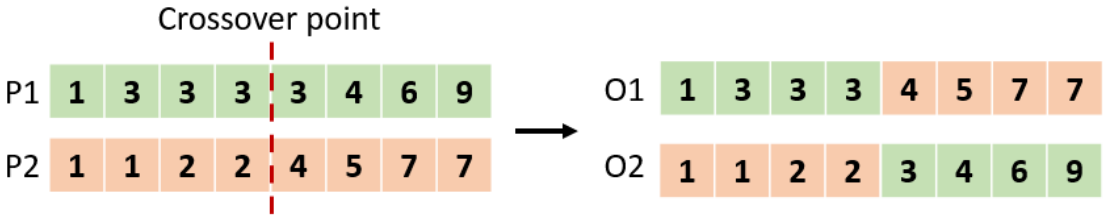


Figure 5.4.3: The same example as Figure 5.4.2, but with the allocations sorted before the crossover operation is done.

Another alternate approach was initially thought of, which was to represent an allocation as a list of numbers for each base station, where each value is the number of ambulances at that base station. However, this approach faces problems when doing genetic operations since the total number of ambulances need to be static.

	Response Time			Survivability
	All	Acute	Urgent	
Best	12m 19s	9m 35s	14m 55s	0.7736
Average	12m 16s	9m 33s	14m 51s	**0.7726**

Table 5.4.5: Optimization results from running the *SortedGA* method.

After crossover is done, the offspring solutions will have a chance of being mutated. This genetic operation will go through every gene and mutate it with a certain probability, which can result in multiple genes in the genotype being mutated. A gene in this genotype is an ambulance, so an ambulance has the chance of being randomly assigned to a new base station. Mutation is usually considered as an explorative operation since it is completely random and does not utilize the knowledge gained during optimization. However, one mutation is a much smaller change to the allocation than the crossover operation, so it effectively searches the neighbourhood of the parents, which might contain good solutions or even better solutions than the parents. Mutation rate is discussed in Section 5.4.1.6.

5.4.1.5 Survival Function

Elitism was selected as the survival function for the GA because it serves two crucial purposes. Firstly, it allows for the preservation of the best solutions in

the population from one generation to the next, ensuring that the most promising individuals continue to contribute to the overall evolution. Secondly, it helps maintain a diverse population, preventing the GA from quickly converging to local optima. The balance between exploration and exploitation can be adjusted by changing the size of the elite population. The elite size is discussed further in Section 5.4.1.6.

5.4.1.6 Parameter Tuning

The optimization method has several parameters that can be tuned to improve performance. Several of the parameters are dependent on each other, so a form of grid search would be optimal. However, since the stochastic nature of the method requires multiple runs to give a fair comparison of the method with different parameters, the process of experimenting is tedious. It was therefore decided to do manual tuning, guided by observing the progress of the population in terms of best fitness, average fitness, and diversity.

The population size is an important parameter that was tuned in the optimization method. The *Baseline* method used a population size of 30 individuals. However, through observation and experimentation, it was found that significantly increasing the population size to 200 individuals improved the algorithm's performance. With a larger population, there was a higher diversity of solutions, allowing for better exploration of the search space. It also helped to mitigate the risk of premature convergence and provided a larger pool of potential parents for the genetic operations. The change of population size has a large effect on the other parameters of the algorithm, and reduced the number of generations to 250 for the 4-minute run.

As mentioned in Section 5.4.1.3, tournament selection was chosen for parent selection. The parameter for tuning the selection pressure of this method is the tournament size. The *Baseline* method used a tournament size of 5, or 16.7% of the population. After increasing the population size and experimenting with both a higher percentage and a lower percentage, it was found that a tournament size of 6, or 3%, was appropriate. It seemed like the diversity of the population was reduced too much because of the selection pressure with a large tournament size, so the algorithm was not able to explore other parts of the search space.

Since the population characteristics will change as the algorithm moves to new generations, some parameters might not be optimal during the whole run. An example is when the crossover probability was experimentally set to 0.7 in order to increase exploration of the search space. It was observed that later generations in the algorithm contained a moderately strong elite population, but offspring created from the crossover operation was very different so the good solutions were not exploited. An idea is therefore to decrease the crossover probability with new generations, such that the algorithm has a higher chance of exploring in the early generations, before the best solutions are exploited in the later generations. The crossover probability was therefore set to linearly decrease from 0.8 to 0.1 across the first 200 generations, before staying static for the rest.

Another parameter that was tuned, is the mutation probability. Since the probability is used on each gene and a genotype consists of 45+29 = 74 genes, the average number of genes being mutated in a solution will be 1 with a mutation probability of 1/74 = 0.014. The *Baseline* method has a mutation probability of 0.05, which results in an average of 3.7 genes being mutated. Mutation is in this implementation of the algorithm used as a local search operation, so a lower mutation probability was considered advantageous. The same dynamic parameter strategy for crossover was chosen for mutation, where the mutation probability is set to decrease from 0.05 to 0.014 across the first 200 generations. This slowly changes the algorithm from focusing on exploration in the early generations to focus on exploitation in later generations.

As mentioned in 5.4.1.5, elitism was chosen as survival function for the population. Different elite sizes were experimented with, including having an elite size equal to population size to see if accelerated convergence would lead to a bad local optima. The results of the experimentation indicated that this was the case. The best elite size found was 10, keeping a balance between exploration exploitation.

The complete list of tuned parameters is presented in Table 5.4.6, and the results of this *TunedGA* optimization method is shown in Table 5.4.7. These tuned parameters gave a significant improvement, and was subsequently chosen to be used as a basis for the rest of the optimization methods in this section.

Parameter	**Value**
Fitness	Survivability
Initializer	Random
Elite Size	10
Generations	250
Population Size	200
Tournament Size	6
Crossover Probability	0.8-0.1
Mutation Probability	0.05-0.014

Table 5.4.6: Configuration parameters for the *TunedGA* optimization method.

	Response Time			Survivability
	All	A	H	
Best	12m 17s	9m 30s	14m 55s	0.7741
Average	12m 14s	9m 31s	14m 50s	**0.7733**

Table 5.4.7: Optimization results from running the *TunedGA* method.

5.4.2 Diversity

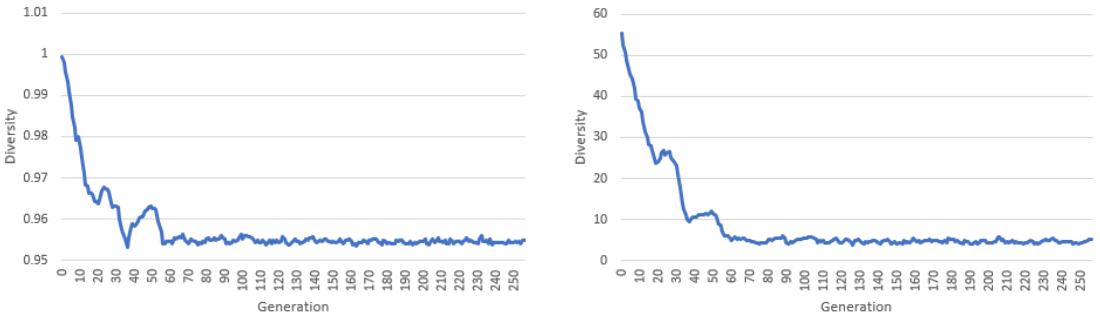
There are different ways to measure diversity. Bekkevold and Schjølberg (2022) implemented the Shannon entropy formula which assumes a probabilistic interpretation of diversity.

An alternative method was also implemented to measure diversity in terms of the phenotype rather than the genotype. This approach utilizes pairwise distance as a

metric for quantifying the differences between two solutions based on the number of ambulances assigned to each base station. The diversity of the population is then determined by calculating the average pairwise distances between all solutions, as expressed in Equation 5.13. In this equation A_i is the allocation of individual i, and N is population size. Since the genotype is not sorted by base station, there are many genotypes that result in the same phenotype, which can make this diversity measurement inaccurate. However, the probability that this happens is low, and the diversity values seemed a good indication of diversity when investigating the population closely at different generations. This diversity measure is therefore used.

$$\text{Diversity} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \text{distance}(A_i, A_j) \tag{5.13}$$

The two diversity measures can be observed in Figure 5.4.4, which shows that they behave similarly as the population evolves. The figure also shows that the diversity is quickly reduced before staying at a low value for most of the generations.



(a) Shannon entropy **(b)** Distance diversity

Figure 5.4.4: Diversity measures for the population for each generation during optimization using the *TunedGA* method.

Lack of diversity in the population can lead to premature convergence and hinder the effectiveness of a GA. When the population becomes too homogeneous, the algorithm may converge to a suboptimal solution or get stuck in a local optima. As seen in the Table 5.4.7, the average survivability of all runs is significantly lower than the survivability from the best run, when considering the range of observed survivability scores in this section. This indicates that the algorithm often converges too quickly to a local optima which is difficult to move away from as the population evolves. In order to explore more of the search space, diversity in the population is critical.

An attempt to maintain diversity was implemented, where offspring would only be added to the population if they are different from their parents. This is checked after both the crossover and the mutation operations are done. Since the tournament selection procedure picks parents that are of high quality, the exclusion of offspring identical to its parent will reduce the speed of convergence, but diversity

will be better maintained. The number of generations increased to around 300 when utilizing this method. The results of this *DistinctGA* method is presented in Table 5.4.8 which shows that the method increased performance significantly.

	Response Time			Survivability
	All	A	H	
Best	12m 9s	9m 28s	14m 43s	0.7751
Average	12m 11s	9m 28s	14m 47s	**0.7739**

Table 5.4.8: Optimization results when running the *DistinctGA* method.

In addition to discarding offspring equal to their parents, an adaptive crowding technique was also experimented with, similarly to the technique presented in Mengshoel, Galán, and de Dios (2014). This method was implemented by pairing offspring with the most similar parent based on the distance metric used in Section 5.13, before selecting the survivor based on the fitness of the solutions in the pair. Generalized crowding was implemented to enable the application of a diverse range of selective pressures through the crowding factor. By incorporating diversity-adaptive control of the scaling factor, the selective pressure of the crowding technique is adjusted according to the current population diversity. The following equation represents how the scaling factor ϕ is adapted to the diversity at generation g as presented in Mengshoel, Galán, and de Dios (2014):

$$\phi(g) = \frac{\Delta(g)}{\Delta(1)} \tag{5.14}$$

As shown in Figure 5.4.4b, the diversity quickly decreases to a value below 10. In order to maintain a lower selective pressure for more generations at the start of the search, the diversity of the first generation $\Delta(1)$ was set to 20, and not 50 which is closer to the usual diversity at generation 1. This causes the crowding selection to have a higher probability of selecting the worst solution in each pair during the first couple of generations. The results of this *CrowdingGA* method is shown in Table 5.4.9 where it can be observed that it performed slightly worse than the *DistinctGA* method. This might be an effect of the crowding factor not being appropriately tuned based on the diversity. Subsequently, the *DistinctGA* optimization method was chosen as a basis for the rest of the methods in this section.

	Response Time			Survivability
	All	A	H	
Best	12m 12s	9m 30s	14m 47s	0.7742
Average	12m 14s	9m 30s	14m 50s	**0.7734**

Table 5.4.9: Optimization results when running the *CrowdingGA* method.

Another approach for maintaining diversity is the use of an IMGA, as presented in Whitley, Rana, and Heckendorn (1998). One way of implementing this strategy is to have multiple separate subpopulations that evolves independently, before

combining the islands into one. Islands promote diversity by maintaining different sets of solutions, facilitating exploration across various regions of the search space. For implementation convenience, the islands were created and evolved sequentially before combining them, since parallel functionality was already used for creating new solutions and running all the simulations required to evaluate them. Unfortunately, this eliminated the possibility of migration of solutions between islands during the evolution process.

It was decided to use 3 islands that could evolve for 90 generations each, before combining the populations into one island. This combined population is reduced to be of the same size as the islands, and is evolved for the rest of the available optimization time for the run, which is typically around 150 generations. The population was reduced to be of size 150, in order to enable more generations for the islands. Additionally, the tournament size was increased to 10, while the crossover and mutation rates now decrease across the first 100 generations. This was done to make the islands converge quicker, so that at least one of the islands will contain good solutions when the islands are combined. The results of this *IMGA* optimization method is presented in Table 5.4.10 which shows that the islands do not lead to an improved survivability score compared to the *DistinctGA* method. The *IMGA* method showed promise earlier on in the development phase, indicating that the parameters might not be tuned correctly.

	Response Time			Survivability
	All	A	H	
Best	12m 4s	9m 24s	14m 38s	0.7748
Average	12m 12s	9m 27s	14m 50s	**0.7738**

Table 5.4.10: Optimization results when running the *IMGA* method.

5.4.3 Constraints

In the real-world system considered in this thesis, there are practical constraints that could be considered. Firstly, the process of creating new base stations is expensive and challenging. As a result, the optimization approach is constrained by only focusing on allocating ambulances to the existing base stations, without considering the option of introducing new ones.

Another constraint is that the base stations have a capacity in terms of how many ambulances that are able to be stationed there. Both because of garage space and because the station need to have enough facilities for the number of ambulance personnel. According to a contact person from OUS, most base stations presented in 2.1.1 have a capacity of about 4 ambulances while Lørenskog, Sentrum, and Ullevål have a capacity of about 10 ambulances each. The standby points are assumed to have a capacity of only 2 ambulances. It could be interesting to observe how the optimization method performs when these constraints are implemented.

It was decided to implement the constraints in the optimization method by giving solutions that break the constraints a penalty to its fitness value. This encourages

the allocations of the solutions to conform to the capacity of the base stations. Another option would be to limit which solutions are possible, but this requires invalid solutions to be corrected which can be an expensive operation.

As shown in the results presented in Table 5.4.11, the inclusion of the constraints in the *ConstrainedGA* optimization method has a noticeable impact on the performance of the algorithm. In Section 5.4.2, the *DistinctGA* method was employed to achieve a diverse set of solutions. However, it was observed that the best allocation obtained through this approach exceeded the capacity of three base stations by a total of six ambulances. Interestingly, all three base stations are standby points, which encourages more research on additional standby points. The number of ambulances exceeding capacity from the *DistinctGA* method suggests that the decrease in performance seen with the *Constrained GA* method could be attributed to the fact that very few or none of the high-quality solutions conform to the constraints. Another possibility is that the penalties assigned to invalid solutions may hinder the search in the vicinity of valid solutions. Even when a valid solution exists nearby, the penalty associated with the invalid region of the search space could prevent the algorithm from further exploring that area.

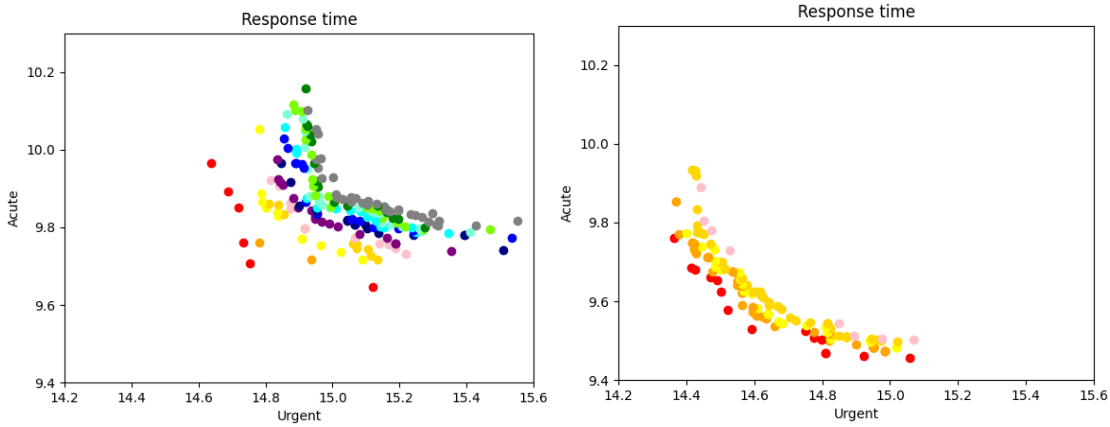
	Response Time			Survivability
	All	A	H	
Best	12m 20s	9m 33s	15m 0s	0.7728
Average	12m 14s	9m 33s	14m 47s	**0.7721**

Table 5.4.11: Optimization results when running the *ConstrainedGA* method.

5.4.4 Multi-Objective Optimization

For evaluating the system, both average response times and survivability have been used. Survivability was introduced to deal with urgency and that acute and urgent incidents should be weighted differently. Because of the dispatch enhancements and strategies in Section 5.2, response time to acute incidents and response time to urgent incidents are influenced by each other in a conflicting manner. When acute response times decrease the urgent response times usually increase. Since there are two potentially conflicting objectives, a multi-objective evolutionary algorithm is proposed.

NSGA-II was chosen as the multi-objective evolutionary algorithm due to its widespread usage and effectiveness, but mainly for its ability to maintain a diverse set of solutions as elaborated upon in Section 3.3.7. The two objectives are average acute response time and average urgent response time. A solution in this implementation will therefore effectively dominate another if it has shorter response time for both acute and urgent incidents, since equal average response time is rare for two different allocations. Figure 5.4.5 shows the different fronts in the population at two stages of the optimization, and how the solutions move towards shorter response times.



(a) Generation 10 **(b)** Generation 50

Figure 5.4.5: Population fronts at two stages of the multi-objective optimization method. The response times are averages in minutes. The Pareto front consists of the solutions colored in red.

The best solution in the population is picked from the Pareto front. Since none of the solutions in the Pareto front can be considered better than the others in terms of the two objectives, the solution with the highest survivability is chosen. All the improving strategies and parameters implemented for the GA in Section 5.4.1 are kept for this *NSGA-II* optimization method, which means that the *DistinctGA* method was used. The results of the *NSGA-II* method are presented in Table 5.4.12, which shows that it on average reaches both the lowest average acute response time and the average urgent response time.

	Response Time			Survivability
	All	A	H	
Best	12m 1s	9m 22s	14m 33s	0.7744
Average	12m 3s	**9m 25s**	**14m 34s**	0.7724

Table 5.4.12: Optimization results when running the *NSGA-II* method.

5.4.5 Results

The results from all the different approaches to the optimization method and their parameter settings show that there are improvements to be found. The most limiting factor is believed to be maintaining diversity of the population shown in Section 5.4.2. This caused most of the optimization runs to converge to a local optima, when the best run showed that there was a significantly better solutions to be found. The *DistinctGA* method was able to improve performance significantly because of the increased diversity. Another factor which made optimizing difficult is that the crossover operation seems to not be effective due to the representation of the solutions.

An overview of the performance of most explored versions of the optimization methods is presented in Figure 5.4.6. This figure shows the survivability of the best allocation found in each run of each optimization version in a box plot. The box plot is a graphical representation that displays the distribution of a dataset by showing the median, quartiles, and any outliers or extreme values. The figure does not show the *Baseline* method that was optimized on response time, since it is unfair to compare it using survivability as fitness.

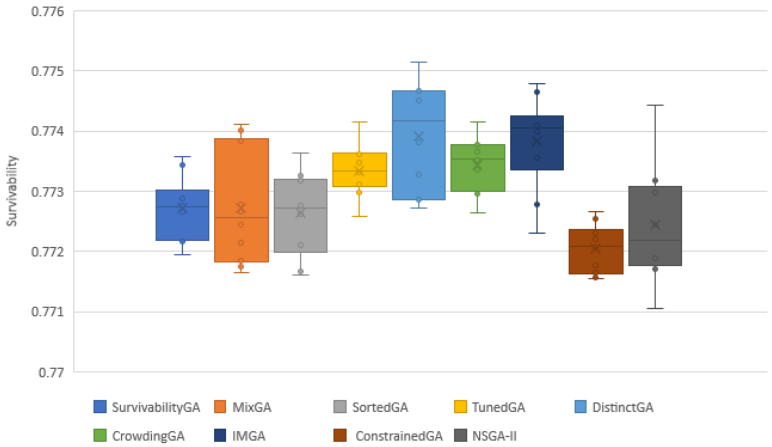


Figure 5.4.6: Box plot of all explored optimization methods using survivability as fitness.

5.4.5.1 Response Time

The best version of the optimization method is considered to be the *DistinctGA* method. This version performed well in terms of survivability, but it is not clear how large the improvement is, and this metric might not be the best metric for performance. Therefore, a comparison to the *Baseline* method is made by optimizing the average response time using the *DistinctGA* method. This comparison is displayed in Figure 5.4.7, which shows that the *DistinctGA* method is slightly more consistent. The difference in lowest response time is only of about 2 seconds, which indicates that around 12 minutes (714 seconds) average response time is close to the lowest possible for the time period of *incidents-simulation*.

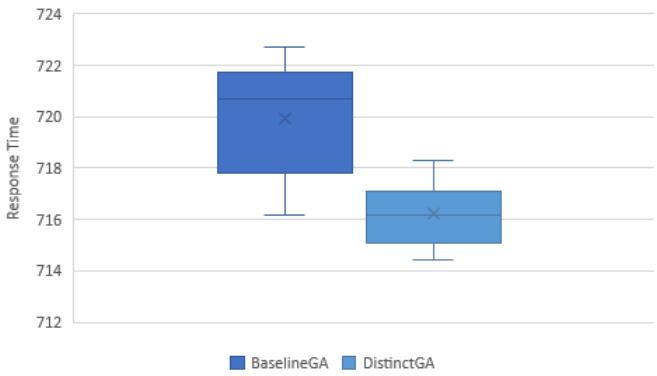
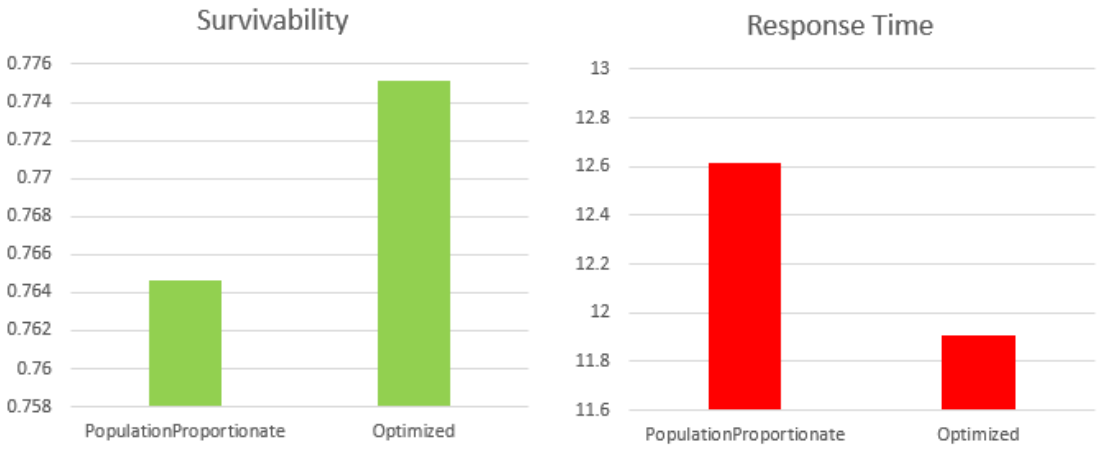


Figure 5.4.7: Box plot comparing the *Baseline* method and the *DistinctGA* method in terms of average response time in seconds.

The *NSGA-II* method seemed to also perform well, since it achieved the lowest average acute and urgent response times of all methods. It is however difficult to compare it to the other methods using survivability or total average response time, since it did not use these metrics as the fitness function for optimization.

5.4.5.2 Allocation Comparison

In Section 5.2 and Section 2.3.1 the *PopulationProportionate* allocation was used to evaluate the impact of different strategies and situations. The purpose of the optimization method is to find a better allocation than *PopulationProportionate* to increase survivability and reduce response times. Figure 5.4.8a shows the performance of the best allocation found with the *DistinctGA* method in terms of survivability, compared to *PopulationProportionate*. The improvement of the optimized allocation is a difference of 0.01, which may not sound like a substantial improvement, but when considering the range of survivability scores, it is significant. Figure 5.4.8b shows a response time comparison of the best allocation found from the *DistinctGA* method when optimizing the response time. The improvement of the optimized allocation is 43 seconds in average response time, which is a more clear improvement.



(a) Survivability **(b)** Response Time

Figure 5.4.8: Comparison of average survivability and response time for the *PopulationProportionate* allocation and the *DistinctGA* allocations.

Figure 5.4.9 and Figure 5.4.10 presents the ambulance count for each base station for the different allocations, which shows that the three allocations are not too dissimilar.

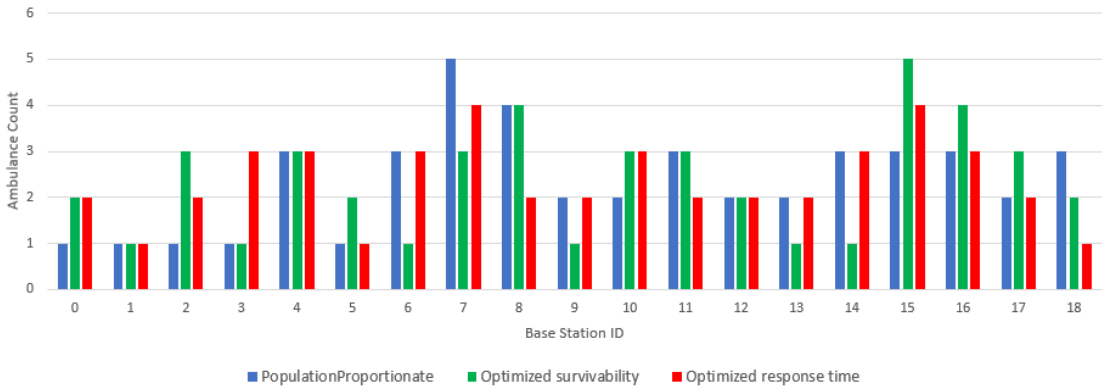


Figure 5.4.9: Comparison of the different allocations for the daytime shift.

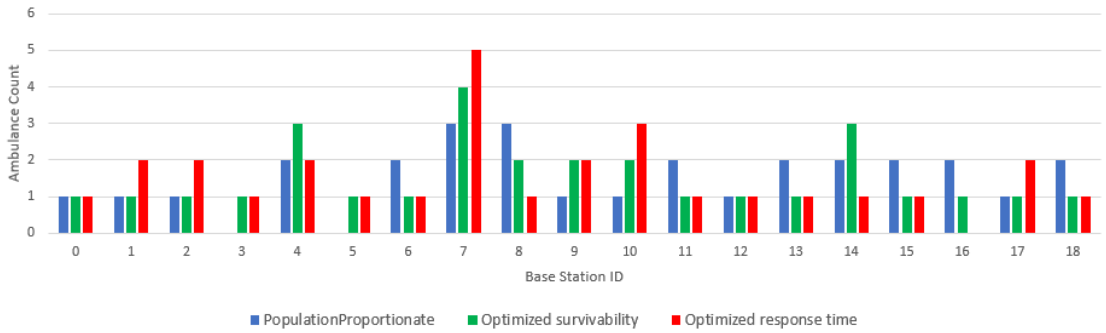


Figure 5.4.10: Comparison of the different allocations for the nighttime shift.

5.4.5.3 Generalization

As mentioned in Section 5.4.5.2, the optimized allocations are quite similar to the *PopulationProportionate* allocation even though the differences in response time and survivability are significant. Small changes in the allocation cause large fluctuations in the results. This indicates that the performance metrics of average response time and survivability are very specific to the set of incidents in *incidents-simulation*.

Figure 5.4.11 shows the results of using the optimized allocations versus the *PopulationProportionate* allocation on the *incidents-simulation-33* set of incidents instead. Even though this dataset is only the next week compared to *incidents-simulation* for which the results are shown in 5.4.8, the improvement of the optimized allocations has reduced, showing the lack of generalization that these allocations have. The optimized allocation achieved an improvements of 0.005 in survivability, and 21 seconds of average response time, which is about 50% of the improvement presented in Section 5.4.5.2.

Bekkevold and Schjølberg (2022) explored simulating over a longer time period than a week and observed that the *PopulationProportionate* allocation performed better than the optimized allocations in terms of average response time when simulating a whole year. A similar result is expected for the optimized allocations on the simulation in this thesis.

(a) Survivability **(b)** Response Time

Figure 5.4.11: Comparison of average survivability and response time for the *PopulationProportionate* allocation and the optimized allocations when simulating on *incidents-simulation-33*. This figure is in comparison to Figure 5.4.8.

CHAPTER

SIX

CONCLUSION

In this thesis, the focus has been on the ambulance allocation problem within the EMS system of Oslo and Akershus. Throughout the previous chapters, various research goals were identified and explored. This chapter provides a summary of the research goals, presenting the main contributions made throughout the thesis. Furthermore, limitations of the research are acknowledged, and suggestions for future work are proposed.

6.1 Contributions

This section provides a summary of the extent to which the thesis achieved its goals and contributed to the knowledge in the research areas.

6.1.1 Goal 1: Improve Simulation Realism

The goal of achieving a higher simulation realism compared to the simulation developed by Bekkevold and Schjølberg (2022) was pursued by expanding the simulated EMS system. The main goal was to enhance the accuracy of how ambulances move in different situations to more closely resemble the patterns observed in the real EMS system. Firstly, the travel time calculations was updated with the implementation of OSM to include the quickest path between two points, which enabled the location of travelling ambulances to be better simulated. Secondly, median times for handling time and dispatch time were implemented instead of the historic times for each incident recorded in the dataset. This improved the simulation to behave similarly to how the real EMS system would behave, since the state of the simulation most likely differ from the historic state at a certain point in time, where the cause of the historic times do not apply for the simulation.

In addition to ambulance movement, a significant improvement was the inclusion of regular incidents. The number of incidents that the system has to respond

to, impacts the average availability of the ambulances, which subsequently makes the central task of dispatching more difficult. The simulated behaviour of the EMCC in terms of dispatching was also improved by the implementation of the reassigning and queuing operations, which are options that the real EMCC has.

The simulation has been developed to not consider certain historic timings since the causes are unknown, in order to improve simulation realism. This has caused the response times of the simulation to be distanced from historic response times, and especially from the outlier response times. It is therefore difficult to evaluate how accurate the simulation is. However, careful observation of the system has been done for many different states, to verify that the system behaves reasonably.

6.1.2 Goal 2: Explore Dispatch Strategies

The second goal was to explore different dispatch strategies that could improve the average response time. The dispatch enhancements of reassigning and queuing ambulance were included in this goal in addition to being a part of the simulation realism in Goal 1, since their effects are interesting to research in relation to the dispatch strategies.

The inclusion of the enhancements improved the average response time of acute incidents, since the reassigning operation prioritizes acute incidents in the dispatching procedure. The balance between reducing response times to both acute and urgent incidents, and in what degree acute incidents should be valued above urgent incidents proved challenging. The coverage-based dispatch strategies showed promise in that they further reduced response time to acute incidents.

The most interesting strategy attempted to use predictions of future demand to influence dispatching decisions. The results of this strategy only showed a small improvement compared to the other coverage based strategies, but further research is needed. The sparsity of the incident data made accurate predictions challenging.

6.1.3 Goal 3: Incident Urgency

The goal of studying the urgency aspect of incidents was firstly pursued by observing the impact on response times from improving the accuracy of urgency assigned to incidents by the EMCC. This effect was most interesting when examined with the different dispatch enhancements and strategies. The reduced number of acute incidents allowed for the dispatch enhancements and strategies to have more of an impact since the EMCC have more options for preserving coverage. The improvement of implementing the dispatch enhancements and strategies when incidents are accurately assigned as acute was 67 seconds in average acute response time, while the improvement when incidents are over-triaged was 53 seconds. This result indicates that reducing over-triaged can be advantageous in terms of resource management, and subsequently response time to critical incidents.

To explore the balance of acute and urgent incidents in relation to prioritization and response time, survivability of the patient was researched. This was done with survival functions that give a survivability score based on the response time to the

incident and its urgency. The survivability of all incidents when different dispatch strategies were applied showed that the survivability is an effective measure of performance. The optimization results also show that survivability considers response time of urgent incidents in addition to response time of acute incidents in the evaluation of the EMS system.

6.1.4 Goal 4: Optimization

The fourth goal was to reduce response times to incidents by optimizing the allocation of ambulances to base stations. This was pursued by exploring different strategies of improving the optimization method. Finding near-optimal solutions proved challenging because of the solution representation being sub-optimal for the crossover operation, but mainly because of diversity preservation. Tuning the parameters of the GA, including dynamic parameters for crossover, mutation and tournament selection, gave a small improvement in performance. The biggest contribution was the strategies implemented for diversity preservation, which gave significantly better allocations.

In addition to the GA, a multi-objective optimization approach was implemented. This approach enabled the optimization of two potentially conflicting objectives, namely acute response time and urgent response time. The results of this method show that it is able to find allocations with lower average response time for both objectives than the allocations found by the islands strategy. This is an encouraging result which motivates further research on multi-objective optimization in the context of acute and urgent response times.

The different allocations found were quite similar, but fluctuated significantly in terms of survivability score and response times. This indicates that the allocations might be overfitted to the incidents in the simulation. Researching the performance of the allocations on a different set of incidents revealed that the optimized allocations, which initially exhibited notable improvements, did not maintain the same level of superiority compared to the non-optimized allocation

6.2 Limitations

This section discusses some of the major limitations related to each research goal.

6.2.1 Goal 1: Improve Simulation Realism

The accuracy of the OSM paths and travel times was not extensively researched. Some random samples showed that although most travel times are close to what you would get when using Google Directions, they seem to be shorter in urban areas since the OSM calculation does not consider traffic. As described in Section 5.1.3.3, the travel time is only calculated from the speed limit and length of the roads of the path. Adjustment of the speed limits was done to get travel times more accurate in urban areas, but this increased the travel times in rural areas. Additionally, the fact that ambulances travel faster to acute incidents is

not considered. The off-road travel times mentioned in 4.1.1 were not included either.

The median times for handling time and dispatch time completely removes the variety of these times that are observed in the real EMS system. Situations where these times are significantly longer or shorter might be important to simulate.

Even though the regular incidents were included to more accurately simulate the number of incidents that the EMCC has to handle, the implemented handling of the regular incidents in the simulation is not accurate. The regular incidents are simulated the same way as urgent incidents, and will only get deprioritized through the reassigning operation. In the real EMS system, the EMCC will often delay these incidents if there is high demand at the time. Additionally, the planned regular incidents are not separated from unplanned regular incidents in the simulation.

The lack of an extensive evaluation of the simulation accuracy reduced the trust that can be placed in the simulation.

6.2.2 Goal 2: Explore Dispatch Strategies

A major limitation of the dispatch strategies is the custom penalty values used to balance the importance of coverage, both depending on how many available ambulances there are and depending on the predicted demand. The penalty values were optimized to improve the result of the strategies on the same set of incidents that was used to compare the strategies. They also depend on the specific allocation used in the simulation. Since the strategies have not been evaluated on different allocations and sets of incidents, the results might not be accurate.

The training of the prediction model used incidents more recent than the incidents in the simulation. This could have given the model information about trends that would otherwise not be possible.

6.2.3 Goal 3: Incident Urgency

The particular synthetic set of incidents utilized to investigate a scenario with reduced over-triaged incidents may yield results that are not representative of the broader range of possible synthetic sets with the same over-triage rate. The various selections of which incidents to correct, could give vastly different outcomes from the simulation. This would also change the effect that the dispatch strategies has on the synthetic dataset.

The lack of research done to calculate the survivability of patients reduces its relevancy. The survival function for acute incidents only uses research related to cardiac arrests, and not other types of acute incidents. The survival function for the urgent incidents have not been created from research, but rather created to work in relation to the importance of acute incidents.

6.2.4 Goal 4: Optimization

The optimization method is stochastic, so the results have a significant variance. This could be a result of the lack of diversity, but the number of runs made to estimate the performance of the methods might also be too small. The number of runs was limited by the amount of time it would take to the test all the variations of the optimization methods.

6.3 Future Work

This section will suggest some ideas and areas for future work in relation to this thesis. The topics are all related to the research goals presented in Section 1.2.

6.3.1 Goal 1: Improve Simulation Realism

A potential change is to vary the travel time of an ambulance depending on what type of incident it is responding to. The OSM method as a whole could potentially be improved, researching the optimal speed limits for travel time accuracy and other possibilities enabled by OSM. Lastly, completely new methods could be researched, for example the Google Direction API or a machine learning method.

To introduce more variability of the median handling time and the median dispatching time, the inclusion of small variances around the median values can be explored. This can help capture the diverse range of scenarios observed in the real EMS system. Additionally, the regular incidents can be simulated more realistically, by incorporating similar methods as those proposed in Kergosien et al. (2015). This will make the simulation handle a more diverse set of situations which are present in the real world.

Conducting a more extensive evaluation of the simulation's accuracy can provide valuable insights and enhance trust in the simulation results. Instead of evaluating accuracy by observing the system at a small set of specific situations, a statistical analysis could be done to evaluate the overall performance and behavior of the simulation.

6.3.2 Goal 2: Explore Dispatch Strategies

One idea for future work is to optimize the penalty values using techniques that provide values that achieve better results across different sets of incidents and allocations. The incorporation of a more extensive search method like a genetic algorithm could be researched.

To address the challenge of accurate predictions due to sparse incident data, research can be done on training the prediction model to predict demand for a longer time period. The random factor of incidents occurring within one hour instead of another made such detailed predictions difficult. It is easier for the prediction model to capture the more general patterns in the data. The predictions would give a less detailed view of future demand, but this might not be damaging for the dispatch strategy. Additionally, considering additional factors such as weather

conditions or other relevant variables in the prediction model could lead to more accurate future demand predictions.

Investigating the possibility of implementing an MDP similar to the one discussed in 4.1.2 can be an interesting approach. An MDP framework can provide a systematic and decision-driven approach to dispatching decisions, considering various factors such as incident urgency, ambulance availability, and predicted demand. This dispatch strategy could be compared to the ones presented in this thesis.

6.3.3 Goal 3: Incident Urgency

Since the survival functions have limitations in terms of their accuracy, getting illness specific coefficients would be interesting. This would only be a benefit if detailed information was available in the dataset of incidents, however. Asking for professional insight on the balance between response time and survivability for different urgencies can be more achievable.

Obtaining more data and a more detailed dataset would be good for the possibility of creating more realistic survival functions. Additionally, information about the actual urgency of the incidents in the dataset would make the limiting synthetic dataset obsolete.

6.3.4 Goal 4: Optimization

Further work in optimizing the genetic algorithm can focus on enhancing diversity within the population. Several strategies can be employed to improve diversity and prevent premature convergence. One approach is to add self-adaptive parameters that evolve as part of the solution representation, including the crowding factor. Another potential improvement could be to enable migration between the islands in the islands method. Other approaches include novelty search, adaptive population size, and fitness sharing. Additionally, more extensive parameter tuning for the different methods could give significant improvements.

In addition to these specific improvements, other optimization methods, such as swarm intelligence algorithms, can also be explored. Techniques like particle swarm optimization or ant colony optimization offer alternative approaches to solving optimization problems that can be better suited for the problem of optimizing ambulance allocations.

REFERENCES

Amorim, Marco, Sara Ferreira, and António Couto (2019). "How do traffic and demand daily changes define urban emergency medical service (uEMS) strategic decisions?: A robust survival model". In: *Journal of Transport and Health* 12, pp. 60–74. ISSN: 2214-1405. DOI: https://doi.org/10.1016/j.jth.2018.12.001.

Aytug, Haldun and Cem Saydam (2002). "Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study". In: *European Journal of Operational Research* 141.3, pp. 480–494. ISSN: 0377-2217. DOI: https://doi.org/10.1016/S0377-2217(01)00260-0.

Bandara, Damitha, Maria Mayorga, and Laura Albert (Aug. 2012). "Optimal dispatching strategies for emergency vehicles to increase patient survivability". In: *Int. J. of Operational Research* 15, pp. 195–214. DOI: 10.1504/IJOR.2012.048867.

Bekkevold, Nicklas Imanuel Paus and Magnus Eide Schjølberg (2022). "Simulation and Optimization of Emergency Medical Services in Oslo and Akershus".

Beraldi, P. and M.E. Bruni (2009). "A probabilistic model applied to emergency service vehicle location". In: *European Journal of Operational Research* 196.1, pp. 323–331. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2008.02.027.

Chang, Pei-Chann, Wei-Hsiu Huang, and Ching-Jung Ting (2010). "Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems". In: *Expert Systems with Applications* 37.3, pp. 1863–1878. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2009.07.066.

Chen, Albert Y. et al. (2016). "Demand Forecast Using Data Analytics for the Preallocation of Ambulances". In: *IEEE Journal of Biomedical and Health Informatics* 20.4, pp. 1178–1187. DOI: 10.1109/JBHI.2015.2443799.

Chun Peng Erick Delage, Jinlin Li (2020). "Probabilistic Envelope Constrained Multiperiod Stochastic Emergency Medical Services Location Model and Decomposition Scheme". In: *Transportation Science*, pp. 1471–1494. DOI: 10.1287/trsc.2019.0947.

Daskin, Mark S. (1983). "A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution". In: *Transportation Science 17*, pp. 48–70. DOI: https://doi.org/10.1287/trsc.17.1.48.

Deb, K. et al. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. DOI: 10.1109/4235.996017.

Erkut, Erhan, Armann Ingolfsson, and Güneş Erdoğan (2008). "Ambulance location for maximum survival". In: *Naval Research Logistics (NRL)* 55.1, pp. 42–58. DOI: https://doi.org/10.1002/nav.20267.

Gozali, Alfian Akbar and Shigeru Fujimura (2019). "Localized Island Model Genetic Algorithm in Population Diversity Preservation". In: *Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*. Atlantis Press, pp. 122–128. ISBN: 978-94-6252-689-1. DOI: 10.2991/icoiese-18.2019.22.

Helsedirektoratet (2022). *AMK - Tid fra AMK varsles til ambulansebil er på hendelsessted [EMCC - Time from EMCC is notified until the ambulance is at the scene]*. URL: https://www.helsedirektoratet.no/statistikk/kvalitetsindikatorer/akuttmedisinske-tjenester-utenfor-sykehus/tid-fra-amk-varsles-til-ambulanse-er-pa-hendelsessted.

Helsetilsynet (2022). *Rapport fra tilsyn med Akuttmedisinsk kommunikasjonssentral i Oslo [Report from the supervision of the EMCC in Oslo]*. URL: https://www.helsetilsynet.no/tilsyn/tilsynsrapporter/oslo-og-viken/2022/akuttmedisinsk-kommunikasjonssentral-i-oslo-2022/.

Henderson, S.G. and A.J. Mason (2004). "Ambulance service planning: Simulation and data visualisation". In: *Operations Research and Health Care: A Handbook of Methods and Applications* 70, pp. 77–102. DOI: 10.1007/1-4020-8066-2_4.

Hermansen, Anna Haugsbø (2021). "Machine Learning for Spatio-Temporal Forecasting of Ambulance Demand".

Huang, Hongyun et al. (2019). "Forecasting Emergency Calls With a Poisson Neural Network-Based Assemble Model". In: *IEEE Access* 7, pp. 18061–18069. DOI: 10.1109/ACCESS.2019.2896887.

Ivanov, Oleksandr et al. (2021). "Improving ED Emergency Severity Index Acuity Assignment Using Machine Learning and Clinical Natural Language Processing". In: *Journal of Emergency Nursing* 47.2, 265–278.e7. ISSN: 0099-1767. DOI: https://doi.org/10.1016/j.jen.2020.11.001.

Kergosien, Y. et al. (2015). "A generic and flexible simulation-based analysis tool for EMS management". In: *International Journal of Production Research* 53.24, pp. 7299–7316. DOI: 10.1080/00207543.2015.1037405.

Knight, V.A., P.R. Harper, and L. Smith (2012). "Ambulance allocation for maximal survival with heterogeneous outcome measures". In: *Omega* 40.6. Special Issue on Forecasting in Management Science, pp. 918–926. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2012.02.003.

Lam, Sean Shao Wei et al. (2015). "Dynamic ambulance reallocation for the reduction of ambulance response times using system status management". In: *The American Journal of Emergency Medicine* 33.2, pp. 159–166. ISSN: 0735-6757. DOI: https://doi.org/10.1016/j.ajem.2014.10.044.

Mason, Andrew James (2013). "Simulation and Real-Time Optimised Relocation for Improving Ambulance Operations". In: *Handbook of Healthcare Operations Management: Methods and Applications*. Ed. by Brian T. Denton. New York, NY: Springer New York, pp. 289–317. DOI: 10.1007/978-1-4614-5885-2_11.

McCormack, Richard and Graham Coates (2015). "A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival". In: *European Journal of Operational Research* 247.1, pp. 294–309. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2015.05.040.

Meisch, Peter-Josef (2023). *mapjfx*. URL: https://www.sothawo.com/projects/mapjfx/.

Mengshoel, Ole J., Severino F. Galán, and Antonio de Dios (2014). "Adaptive generalized crowding for genetic algorithms". In: *Information Sciences* 258, pp. 140–159. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.08.056.

Nakamura, Sho (2023). *Matplotlib for java: A simple graph plot library for java, scala and kotlin with powerful python matplotlib*. URL: https://github.com/sh0nk/matplotlib4j.

Norkart (2023). *Få full oversikt med Norges mest komplette database for geografisk informasjon]*. URL: https://www.norkart.no/planoggeodata/.

NRK (2022). *AMK-krisen: 70 har sluttet de siste årene [The EMCC crisis: 70 have quit in recent years]*. URL: https://www.nrk.no/osloogviken/mange-ansatte-ved-amk-sentralen-i-oslo-sluttet-etter-hoyt-arbeidspress-og-darlig-ledelse-1.16154742.

Olivos, Carlos and Hernan Caceres (2022). "Multi-objective optimization of ambulance location in Antofagasta, Chile". In: *Transport 37*, pp. 177–189. DOI: https://doi.org/10.3846/transport.2022.17073.

OSM (2023). *OpenStreetMap provides map data for thousands of websites, mobile apps, and hardware devices*. URL: https://www.openstreetmap.org/about.

OUH (2022a). *Akuttmedisinsk kommunikasjonssentral (AMK) [Emergency Medical Communication Center]*. URL: https://oslo-universitetssykehus.no/avdelinger/prehospital-klinikk/akuttmedisinsk-kommunikasjonssentral-amk.

– (2022b). *Ambulanseavdelingen [The ambulance service division]*. URL: https://oslo-universitetssykehus.no/avdelinger/prehospital-klinikk/ambulanseavdelingen.

Raczynski, Stanislaw (2003). "Continuous Simulation". In: *Encyclopedia of Information Systems*. Ed. by Hossein Bidgoli. New York: Elsevier, pp. 267–286. ISBN: 978-0-12-227240-0. DOI: https://doi.org/10.1016/B0-12-227240-4/00018-6.

Ridler, Samuel, Andrew J. Mason, and Andrea Raith (2022). "A simulation and optimisation package for emergency medical services". In: *European Journal of Operational Research* 298.3, pp. 1101–1113. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.07.038.

Schmid, Verena and Karl F. Doerner (2010). "Ambulance location and relocation problems with time-dependent travel times". In: *European Journal of Operational Research* 207.3, pp. 1293–1303. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2010.06.033.

Van Barneveld, Thije et al. (2018). "Real-time ambulance relocation: Assessing real-time redeployment strategies for ambulance relocation". In: *Socio-Economic Planning Sciences* 62, pp. 129–142. ISSN: 0038-0121. DOI: https://doi.org/10.1016/j.seps.2017.11.001.

Van De Weijer, Erling and Odd Andre Owren (2022). "Forecasting Ambulance Demand in Oslo and Akershus".

Whitley, Darrell, Soraya Rana, and Robert Heckendorn (Dec. 1998). "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence". In: *Journal of Computing and Information Technology* 7.

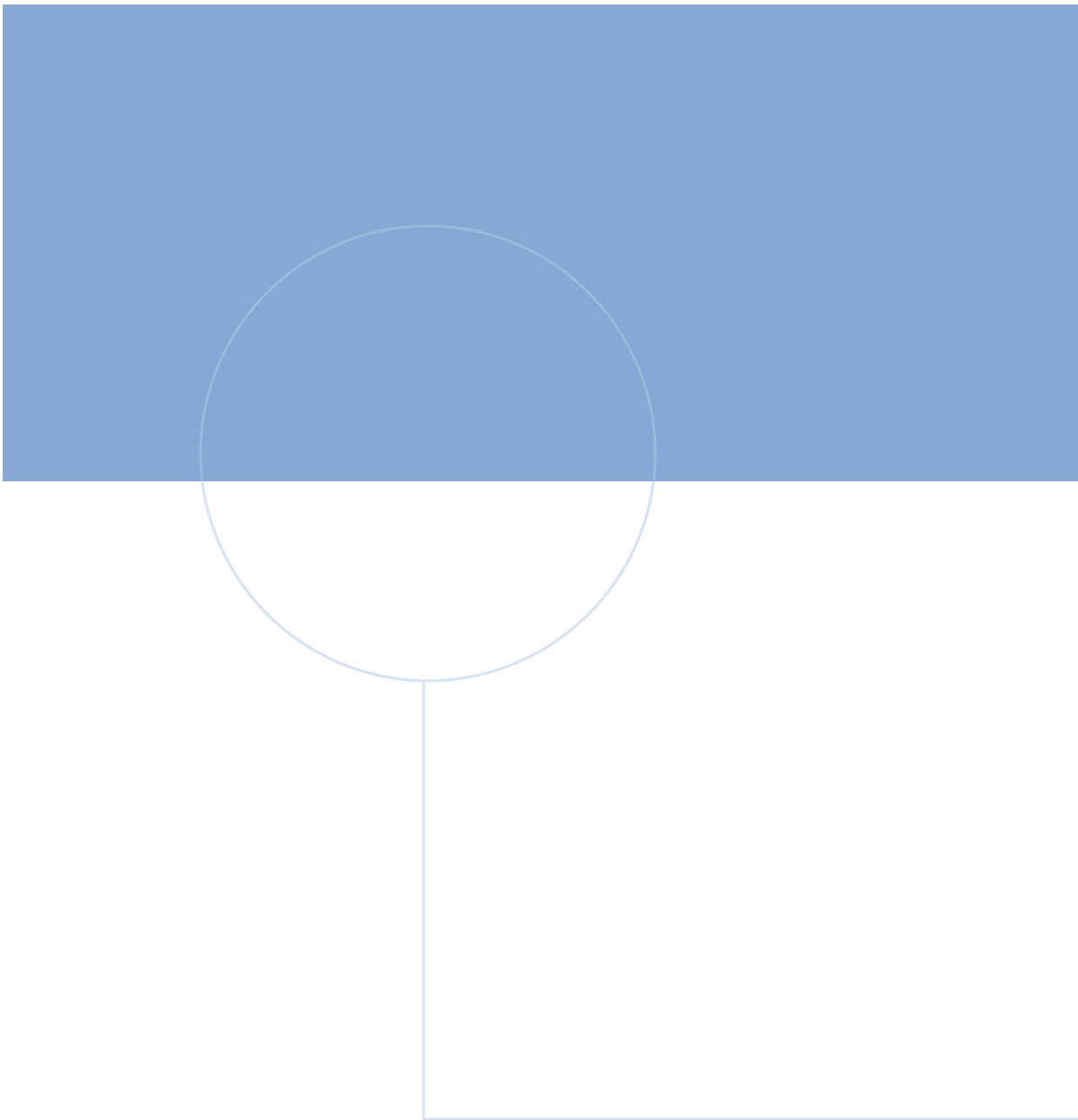
Yang, Shengping and Gilbert Berdine (Jan. 2015). "Poisson Regression". In: *The Southwest Respiratory and Critical Care Chronicles* 3, p. 61. DOI: 10.12746/swrccc.v3i9.191.

Yue, Yisong, Lavanya Marla, and Ramayya Krishnan (2021). "An Efficient Simulation-Based Approach to Ambulance Fleet Allocation and Dynamic Redeployment". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 26.1, pp. 398–405. DOI: 10.1609/aaai.v26i1.8176.

Zaffar, Muhammad Adeel et al. (2016). "Coverage, survivability or response time: A comparative study of performance statistics used in ambulance location models via simulation–optimization". In: *Operations Research for Health Care* 11, pp. 1–12. ISSN: 2211-6923. DOI: https://doi.org/10.1016/j.orhc.2016.08.001.

Zhang, Yongqiang et al. (2022). "Emergency Response Resource Allocation in Sparse Network Using Improved Particle Swarm Optimization". In: *International Journal of Environmental Research and Public Health* 19.16. ISSN: 1660-4601. DOI: 10.3390/ijerph191610295.

Zhou, Zhengyi and David S. Matteson (2015). "Predicting Ambulance Demand: A Spatio-Temporal Kernel Approach". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: Association for Computing Machinery, pp. 2297–2303. ISBN: 9781450336642. DOI: 10.1145/2783258.2788570.



NTNU

Norwegian University of
Science and Technology