Master's thesis

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

Magnus Eide Schjølberg
Nicklas Imanuel Paus Bekkevold

Simulation and Optimization of Emergency Medical Services in Oslo and Akershus

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
June 2022

Photo: Ole Kristian Andreassen (Oslo University Hospital)

NTNU
Norwegian University of Science and Technology

Oslo University Hospital

Magnus Eide Schjølberg
Nicklas Imanuel Paus Bekkevold

Simulation and Optimization of Emergency Medical Services in Oslo and Akershus

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
June 2022

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

Every year, the number of emergency cases handled by the Emergency Medical Communication Centre (EMCC) in Oslo and Akershus increases. Unfortunately, the increasing demand for prehospital services is not met by a corresponding increase in supply, leading to an increased pressure on the workforce.

This thesis aims to solve some of these challenges by optimizing already existing resources, known in the area of operational research as ambulance allocation. For this purpose, a real-world data set containing emergency events from the years 2015–2019 in Oslo and Akershus has been lent to the authors. The main contribution of this thesis is a discrete and trace-based simulation model based on the data set, with the resulting average response times used to evaluate a given set of ambulance allocations and their performance. In addition, heuristic optimization algorithms from the field of artificial intelligence, consisting of a genetic algorithm, stochastic local search, and a memetic algorithm, are used with this simulation model to find optimal ambulance allocations, testing against a set of baseline allocation models. The various models are then compared against each other over various time periods, and the simulation is tested with a number of different scenarios with varying numbers of ambulances, to try to find an optimal number of available ambulances to serve time-critical incidents.

The thesis also presents a review of relevant literature, as well as suggestions for future work that can help improve upon the current state-of-the-art.

Sammendrag

Hvert år øker antallet akutthendelser som håndteres av akuttmedisinsk kommunikasjonssentral i Oslo og Akershus. Dessverre møtes ikke den økende etterspørselen etter prehospitale tjenester av en tilsvarende økning i tilbudet, noe som fører til økt press på de ansatte.

Denne oppgaven har som mål å løse noen av disse utfordringene ved å optimalisere allerede eksisterende ressurser, kjent innen operasjonsanalyse som ambulanseallokering. For denne oppgaven er det lånt ut et datasett som inneholder akutthendelser fra årene 2015 til 2019 i Oslo og Akershus. Hovedbidraget til denne oppgaven er en diskret og sporbasert simulasjonssmodell basert på datasettet. De gjennomsnittlige responstidene som rapporteres fra denne simulasjonsmodellen blir deretter brukt til å evaluere et gitt sett med ambulanseallokeringer og måle deres egnethet. Dette brukes i kombinasjon med heuristiske optimaliseringsalgoritmer fra feltet kunstig intelligens, bestående av en genetisk algoritme, stokastisk lokalt søk og en memetisk algoritme, for å finne optimale ambulanseallokeringer. For å evaluere ytelsen til disse algoritmene så testes disse i tillegg opp mot flere enkle allokeringsmodeller. De ulike modellene sammenlignes deretter mot hverandre over ulike tidsperioder, og simuleringen testes med en rekke ulike scenarier med varierende antall ambulanser, for å prøve å finne et optimalt antall tilgjengelige ambulanser for å betjene tidskritiske hendelser.

Oppgaven presenterer også en gjennomgang av relevant litteratur, samt forslag til fremtidig arbeid som kan bidra til dette forskningsfeltet.

Preface

This thesis is an independent work by Magnus Eide Schjølberg and Nicklas Imanuel Paus Bekkevold, carried out as the finalization of their master's degree in Computer Science at the Department of Computer Science under the Faculty of Information Technology and Electrical Engineering at the Norwegian University of Science and Technology (NTNU). Some of the material in this thesis, in particular the section on related work, is based on a report first written for the course TDT4501 Computer Science, Specialization Project during the autumn semester 2021, as a pre-project for this thesis. The supervisor of this thesis is professor Ole Jakob Mengshoel, also at the Department of Computer Science.

We would like to thank our supervisor for his guidance and enthusiasm towards our thesis. This thesis was supervised by him in close cooperation with a related thesis written on the subject of predicting EMS demand within Oslo and Akershus. We would also like to thank the authors of this related thesis, Erling Van De Weijer and Odd André Owren, for their shared enthusiasm and cooperation.

We also thank Oslo and Akershus Ambulance Department (OAAD) and The Norwegian National Advisory Unit for Prehospital Emergency Medicine (NAKOS) for providing the dataset used in this thesis, and in particular professor Jo Kramer-Johansen for receiving the authors at Oslo University Hospital (OUH) and sharing his knowledge and insights into EMS systems and the ambulance service. We would also like to thank the Emergency Medical Communication Centre (EMCC) for facilitating an on-site excursion and allowing us to gain invaluable insight into the inner workings of their operations.

Finally, we would like to thank Norkart for sharing their advanced route calculation model "Ferd" and accompanying GPS data with us, and in particular Rune Aasgaard for assisting with setup and configuration of this model.

Magnus Eide Schjølberg & Nicklas Imanuel Paus Bekkevold
Trondheim, June 10, 2022

Contents

List of Figures **ix**

List of Tables **xi**

Acronyms **xiii**

1 Introduction **1**
1.1 Background and motivation 1
1.2 EMS incident data set 4
1.3 Goals and research questions 5
1.4 Research method 7
1.5 Contributions 9
1.6 Confidentiality requirements 9
1.7 Disclaimer 9
1.8 Thesis structure 10

2 Background **11**
2.1 Oslo and Akershus ambulance department 11
2.1.1 Triage 12
2.1.2 Triple aims 13
2.1.3 Base stations and Standby points 14
2.1.4 Hospital locations 15
2.2 Data analysis 17
2.3 Solution space size 20
2.4 Software tools and resources 21
2.4.1 Demographic data 22
2.4.2 Travel time and travel distance estimates 22

3 Related work **23**
3.1 Problem ontology . . . 23
3.1.1 Ambulance location . . . 23
3.1.2 Ambulance allocation . . . 24
3.1.3 Ambulance routing . . . 24
3.1.4 Ambulance dispatching . . . 25
3.2 Ambulance optimization in literature . . . 25
3.2.1 Evolutionary algorithms . . . 26
3.2.2 Chromosome representation . . . 26
3.2.3 Simulated EMS system . . . 27
3.2.4 Performance metrics . . . 28
3.2.5 Travel time modeling and routing . . . 30

4 Simulation **33**
4.1 Definitions and notation . . . 33
4.2 Simulation model . . . 34
4.2.1 Discrete Event Simulation (DES) . . . 34
4.2.2 Assumptions . . . 34
4.2.3 Implementation . . . 36
4.2.4 Warm-up buffer . . . 36
4.2.5 Shift rotation . . . 36
4.2.6 Simulation event types . . . 39
4.2.7 Simulated dispatch policy . . . 40
4.3 Travel time estimation . . . 40
4.3.1 Route interpolation . . . 41
4.3.2 Location updates en route . . . 42
4.4 Data pipeline . . . 44
4.4.1 Incident aggregation . . . 44
4.4.2 Structural corrections . . . 45
4.4.3 Exclusion of green missions . . . 45
4.5 Visualization . . . 45
4.6 Verification and validation . . . 46

5 Optimization **49**
5.1 Optimization model . . . 49
5.1.1 Solution encoding . . . 50
5.1.2 Software implementation . . . 51
5.1.3 Heuristic search . . . 51
5.2 Stochastic Local Search (SLS) . . . 52
5.2.1 SLS implementation . . . 53
5.2.2 Alternative SLS implementation . . . 54

5.3 Genetic algorithms . . . 55
5.3.1 Population initialization . . . 56
5.3.2 Genetic operators . . . 59
5.3.3 Diversity . . . 60
5.4 Memetic algorithms . . . 61

6 Experiments and Results **63**
6.1 Experiment 1: Simple allocation methods . . . 63
6.1.1 Objective . . . 63
6.1.2 Design . . . 64
6.1.3 Results and discussion . . . 66
6.2 Experiment 2: Advanced allocation methods . . . 69
6.2.1 Objective . . . 69
6.2.2 Design . . . 69
6.2.3 Results and discussion . . . 70
6.3 Experiment 3: Varying simulated time period . . . 74
6.3.1 Objective . . . 74
6.3.2 Design . . . 74
6.3.3 Results and discussion . . . 74
6.4 Experiment 4: Varying number of ambulances . . . 76
6.4.1 Objective . . . 76
6.4.2 Design . . . 76
6.4.3 Results and discussion . . . 77

7 Conclusion **79**
7.1 Results and discussion . . . 79
7.2 Contributions . . . 80
7.3 Limitations . . . 81
7.4 Future work . . . 81
7.4.1 Improving the simulation model . . . 82
7.4.2 Temporal travel time model . . . 82
7.4.3 Synthesizing historic allocation data . . . 82
7.4.4 Survival Functions: Norwegian Heart Failure Registry . . . 82
7.4.5 Dispatch policy through Reinforcement Learning . . . 83

Bibliography **85**

List of Figures

1.1 EMS timeline illustrating the response time 2
1.2 EMS cases in Oslo and Akershus from 2015–2018 2
1.3 Aggregated EMS cases from 2015–2018 in Oslo and Akershus. . . . 6
1.4 The three main topics constituting the research area of this project: optimization, simulation and the EMS domain. 8

2.1 Photography of Ullevål base station and one of the ambulances managed by OAAD. 12
2.2 Triage hierarchy used in Norway. 12
2.3 Average number of incidents per hour 18
2.5 Violin plot for each month showing the distribution of average incidents per day 18
2.4 Average weekly incidents per hour. 19
2.6 Deviation from daily mean for each month. 20

3.1 Comparison of typical city layouts in the US and Norway. 31

4.1 Map of Oslo and Akershus with junctions 42
4.2 Heatmap showing the Ferd times from Ullevål hospital to other locations 43
4.3 Data pipeline with the associated steps and row sizes presented as a funnel chart. 44
4.4 Simulation visualization. 46
4.5 Historic vs simulated response times for week 32 August 2017. . . . 47

5.1 Overview of the main parts of the optimization model. 51
5.2 The SLS forward step function visualized. 53

5.3 Box plot showing the distribution of average response times for the Forward SLS (FSLS) and the Hamming SLS (HSLS) over the course of 15 runs. 56
5.4 A flowchart demonstrating the life cycle of a generic genetic algorithm. 58
5.5 One-point crossover illustration 59
5.6 Mutate operator illustration 60

6.1 Base station clusters based on population statistics from 2018 . . . 65
6.2 Experiment 1 response times. 67
6.3 Experiment 1 response time distribution from the allocation produced by each method. 68
6.4 Experiment 2 response times taken from the best allocation from each algorithm over 15 runs. 71
6.5 Experiment 2 response time distribution from the best allocation produced by each method. 72
6.6 Box plot showing the distribution of average response times for each of the algorithm used over 15 runs. 72
6.7 SLS search progress from the run that produces the best solution. . 73
6.8 GA and MA search progresses from the run that produced the best solution. 73
6.9 Rank of the allocation produced by the methods in experiment 1 and 2, tested with simulations of varying period length. 75
6.10 Average response time of the GA and PopulationProportionate strategy on a variety of ambulance allocations with a constant nighttime to day-time ambulance ratio of 0.64. 77
6.11 Performance of the PopulationProportionate initializer (on a logarithmic scale) on a variety of day-shift and night-shift ambulance allocations. 78

List of Tables

2.1 Triage levels and their frequency in the data set 13
2.2 Base stations in Oslo and Akershus 14
2.3 Standby points in Oslo and Akershus as of 2019. 15
2.4 Hospitals and emergency wards in Oslo and Akershus 16
2.5 Number of incidents per year in the data set 17
2.6 Dispatch types and their frequency in the data set 19

4.1 The simulation configuration parameters $\boldsymbol{\theta}$ and their default values. 38

5.1 Different types of optimization problems. 50
5.2 FSLS and HSLS comparison test results over 15 runs. 55

6.1 The resulting allocations from experiment 1. 66
6.2 Experiment 1 average response time statistics for the 15 runs. . . . 67
6.3 Parameters used by SLS in experiment 2. 69
6.4 Parameters used by GA and MA in experiment 2. 70
6.5 The best allocation produced by each algorithm from experiment 2 over 15 runs. 71
6.6 Experiment 2 fitness statistics for the 15 runs. 71
6.7 Parameters used by the simulation while running the different time frames in experiment 3. 74
6.8 Parameters used in sub-experiment 1 of experiment 4. 76
6.9 Parameters used in sub-experiment 2 of experiment 4. 76

Acronyms

ALP Ambulance Location Problem.

ARP Ambulance Routing Problem.

DES Discrete Event Simulation.

EMCC Emergency Medical Communication Centre.

EMS Emergency Medical Service.

GA Genetic Algorithm.

ILP Integer Linear Programming.

MA Memetic Algorithm.

MIP Mixed Integer Programming.

NAKOS The Norwegian National Advisory Unit for Prehospital Emergency Medicine.

NDA Non-Disclosure Agreement.

NFL No Free Lunch.

OAAD Oslo and Akershus Ambulance Department.

OUH Oslo University Hospital.

RQ Research Question.

SLS Stochastic Local Search.

SSB Statistics Norway.

Chapter 1

Introduction

This chapter serves as a guide to the master's thesis and summarizes the key points of the document. The history of this project, as well as the motivation behind it, is briefly mentioned before proceeding with the overarching goal and research questions.

1.1 Background and motivation

The Emergency Medical Communication Centre (EMCC) department at Oslo University Hospital (OUH) handles all calls to the emergency number for medical services, 113, in and around Oslo, Norway. This includes the county of Oslo, as well as the regions of Akershus, Østfold and Kongsvinger which has a combined population of 1,687,207 as of 2019.[1] Fast response times from an Emergency Medical Service (EMS) are critical in incidents involving cardiac arrest, stroke, and other severe trauma to reduce unnecessary death, loss of function, and quality of life [The Norwegian Directorate of Health, 2021].

The response time is defined as the period from the time the EMCC receives an emergency call until a unit arrives and stops at the location of the emergency incident, as illustrated by Figure 1.1. This is an important quality indicator for the performance of EMCC and is continuously measured by The Norwegian Directorate of Health [2021]. The same source lists the following as national response goals:

- 90% of acute incidents having a response time less than 12 minutes in densely populated areas.

[1] Population statistics from SSB (Accessed 22.05.2022). https://www.ssb.no/en/befolkning/folketall/statistikk/befolkning

- 90% of acute incidents having a response time less than 25 minutes in sparsely populated areas.

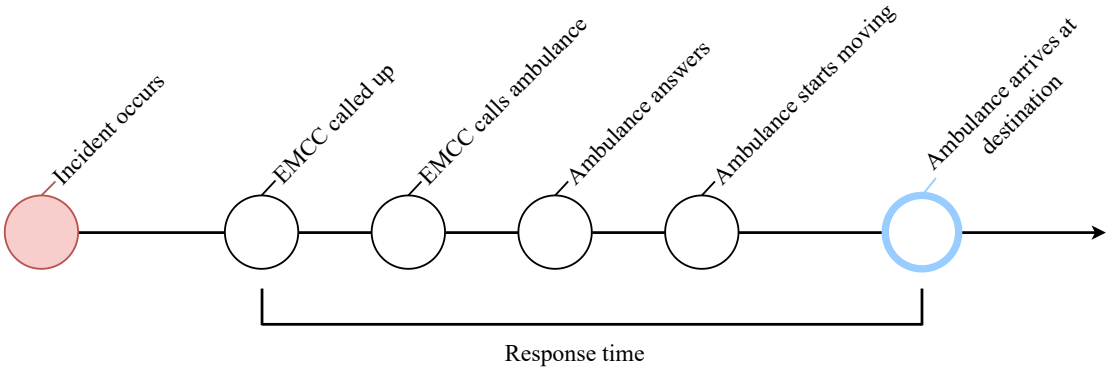


Figure 1.1: EMS timeline illustrating the response time, adapted from Olsen et al. [2018].

Unfortunately, the document also shows that none of the four health regions in Norway was able to meet these response time goals, as of 2020. In addition, the number of daily incidents handled by the EMCC department of OUH increases every year, as can be observed in Figure 1.2. This graph is based on a data set provided by OUH, containing a history of emergency calls during the period 2015–2018, as detailed in section 1.2.

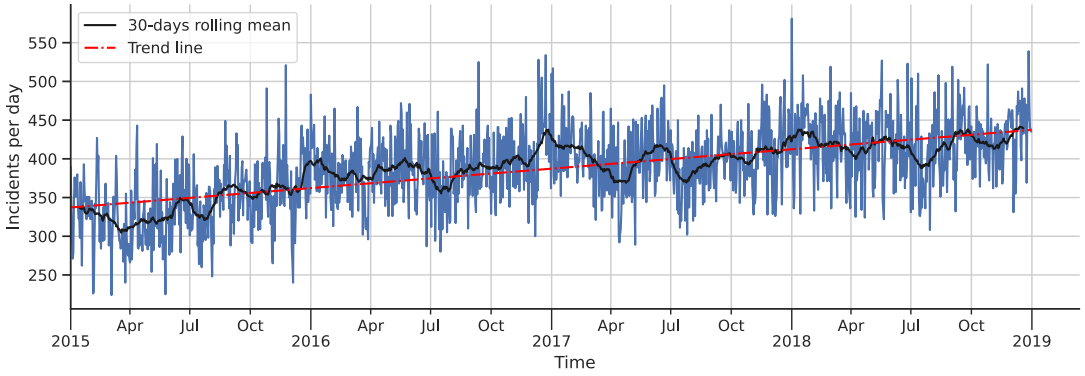


Figure 1.2: EMS cases in Oslo and Akershus from 2015–2018. There is an upwards trend in numbers of cases over the years, as shown by the trend line and 30-days rolling mean.

The increase in EMS demand is also corroborated in a report by The Norwegian Labour Inspection Authority [2020]. Although this trend would imply that the

number of active ambulances should increase over time to accommodate the increasing demand, this is not necessarily the only change needed. According to representatives from OUH and its EMCC department, they cannot meet this increased demand simply by increasing the number of employees, due to demanding work conditions. As a result, sick leave is increasing, employees are quitting, and the work environment is becoming more hectic and stressful. Other approaches that could help in this regard could be to better use existing resources by dynamically redistributing teams over base stations or by varying the number of active ambulances depending on the time of day, day of the week, or month, based on the seasonality of the data.

In a previous related thesis [Hermansen & Mengshoel, 2021b], EMS data from Oslo and Akershus, Norway was collected over a period of time (2015–2019) and applied to a variety of predictive models in an attempt to forecast the spatio-temporal demand for future dates. This thesis represents part of a larger study on whether EMS response times can be minimized by strategically placing ambulances, based on forecasted demand. Since 2016, Oslo and Akershus Ambulance Department (OAAD) (which is part of OUH) has redistributed resources dynamically by employing multiple standby points in strategic locations around Oslo and Akershus where ambulances are periodically stationed throughout the day to better serve the population. More on that in section 2.1. Walsh [2019] studies how the introduction of these standby points has impacted ambulance personnel and their working environment. Although the ambulance personnel interviewed experienced situations where base stations benefited the public, the same study indicates that these base stations have a significantly negative impact on the working environment for ambulance personnel and have led to higher turnover rates. This is attributed in part to missing necessary facilities, such as toilets, and other factors, including shorter time for social interactions with colleagues. This is another factor that, although mostly out of scope for this study, should be taken into account in any framework designed to optimize the utilization of EMS resources.

This master's thesis aims to expand on the work of Hermansen and Mengshoel [2021b], by introducing an optimization framework for EMS resource planning, as well as an appropriate simulation model for estimating response times based on a variety of resource allocation scenarios. An overview of the various optimization problems prevalent in EMS resource planning can be found in section 3.1. In addition, a survey of various travel time estimation models, which is necessary in order to evaluate any optimization model based on response time, is found in section 3.2.5. The resulting optimization framework from this thesis is intended to be agnostic to the origin of the data and can be applied to the spatio-temporal ambulance demand forecasts produced by studies like Hermansen and Mengshoel [2021b] in order to determine the optimal *future* allocation of ambulances in Oslo

and Akershus.

1.2 EMS incident data set

The data set provided by OUH contains 754,811 EMS incidents from January 1st 2015 to February 11th 2019. Each entry in the data set consists of a specific incident that was serviced, either by an ambulance or some other EMS resource, along with relevant timestamps as well as anonymized grid coordinates. Due to the sensitive nature of EMS and data from health registries in general, the data set has been made available in an aggregated and anonymized format. Instead of listing the exact coordinate location of each ambulance event, the locations are mapped to standardized centroids within $1 \times 1\,km$ grid cells. This grid system is a national standard, developed and maintained by Statistics Norway (SSB), the national statistical institute of Norway. Details about the standard and how the grids are calculated are specified in Strand and Bloch [2009]. Each incident is detailed with the assessed priority of the event, as detailed in section 2.1.1, in addition to a unique vehicle resource ID and vehicle type. In addition, each row contains a number of timestamps that will be used to simulate events in chapter 4. The following timestamps are present in the data set:

1. When an incident is reported to the EMCC.
2. When the incident is added to the data system.
3. When an ambulance is notified of the incident.
4. When the ambulance starts responding to the incident.
5. When the ambulance arrives at the scene of the incident.
6. When the ambulance leaves the scene of the incident.
7. When the ambulance arrives at the delivery location (in case of being assigned patient transport duty).
8. When the ambulance becomes available and can respond to new incidents.

The data set contains incidents from the entire country spread across 5,089 grid cells. The region of interest for this project is only within the operating area of OAAD, the county of Oslo, and the region of the former county of Akershus, which colloquially will be referred to as "Oslo and Akershus" from now on. Akershus was merged with two other counties in 2020, but the name is still used today to denote the area covered by the former county. Oslo and Akershus contain a total of 5,569

grid cells, with only 2,606 of them having any registered incidents occurring in the data. A heatmap of all incidents from 2015–2018 in Oslo and Akershus is shown in Figure 1.3.

1.3 Goals and research questions

This section describes what the authors aim to achieve with their research by presenting their goals and research questions (RQs).

Goal *Maximize EMS patient survivability through ambulance demand forecasting and strategic ambulance allocation in Oslo and Akershus.*

In recent literature, patient survivability has been highlighted as a good performance metric when optimizing EMS resources because it models the relationship between response time and survival rate [Erkut et al., 2008].

It is also worth mentioning that this work is part of a larger research initiative. There is another master's thesis being written in parallel with this one by Erling van de Weijer and Odd André Owren (supervisor: Ole Jakob Mengshoel) that focuses on forecasting ambulance demand, building on last year's work from Hermansen and Mengshoel [2021a]. Thus, this goal encompasses both research areas that are being worked on in this collaboration between NTNU and OUH. To make progress towards this goal, two research questions are defined that will be explored in the course of this thesis.

Research Question 1 *What level of realism can be achieved with a simulation based on the EMC incident data set provided by OUH?*

Implementing a data-driven simulation model based on historic EMS data has never been done before for the relevant region, so finding out if this is even possible will be an important first step. A simulation model is a computer representation of a real-world system that can be used to analyze that system. In this case, the model will simulate the ambulance service in Oslo and Akershus, with ambulances responding to real historical events and transporting patients to real-world hospital locations. It is not given that such a model will be good enough to give interesting results, as the data required to evaluate a realistic optimization model may be missing or the quality or quantity of the data provided might be insufficient. There are also a number of potential factors that could possibly invalidate the model. Similar experiments have been performed in the relevant recent literature, where some are highly dependent on the content of the data set [Yue et al., 2012] [McCormack & Coates, 2015], and others use a probabilistic model [Yang et al., 2019] or a probability density function [Zhen et al., 2014] to model demand and

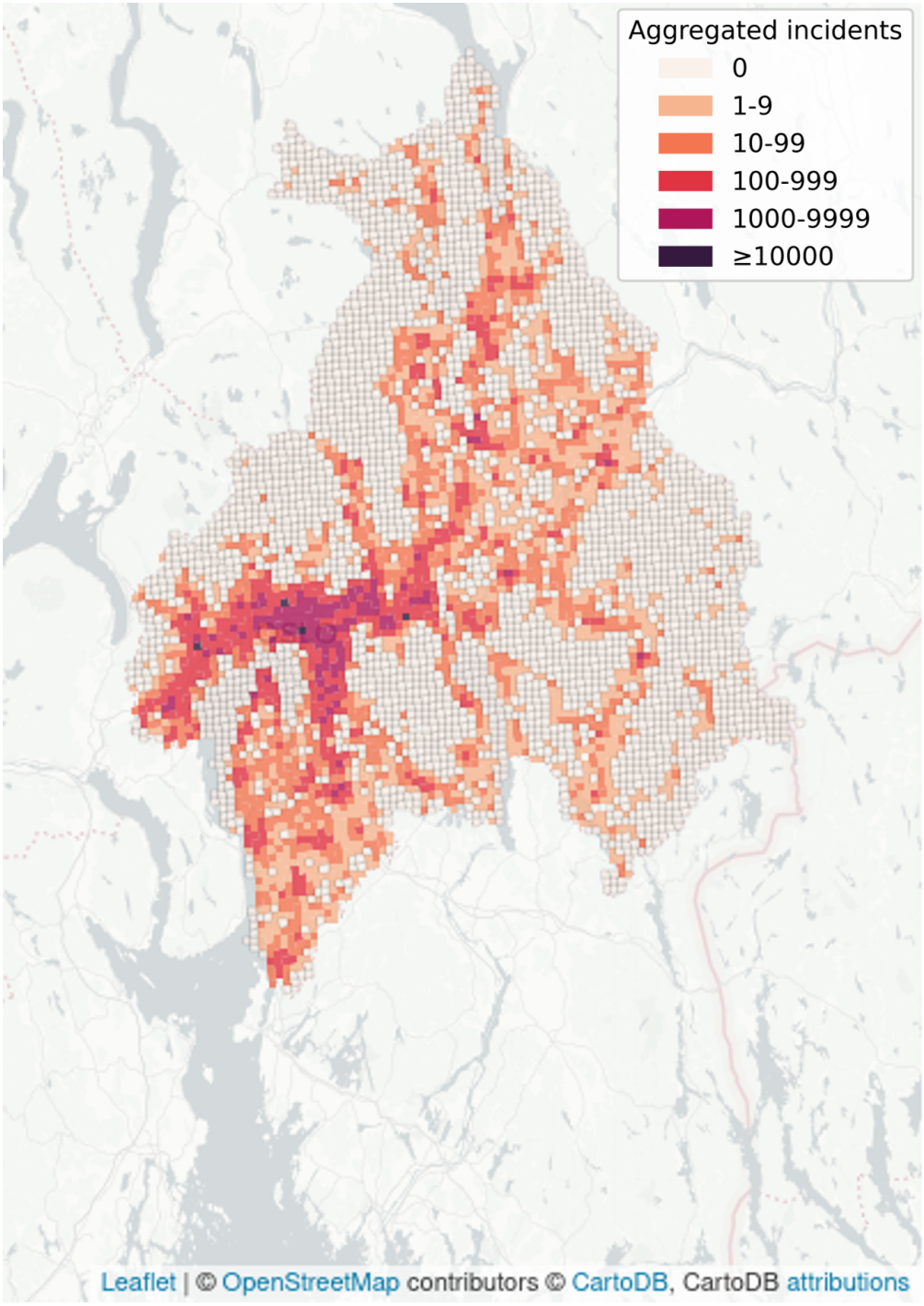


Figure 1.3: Aggregated EMS cases from 2015–2018 in Oslo and Akershus.

is therefore less dependent on an extensive data set. This suggests that it should be possible to create a simulation based on the data set provided by OUH, even if the accuracy or contents of the data are lacking. The level of realism might however need to be reduced, or certain assumptions may need to be made in order to sufficiently simplify and model the system appropriately.

Research Question 2 *In what ways can an ambulance system be optimized using the simulation model from RQ1?*

Multiple previous studies have performed optimization using a simulated EMS system. There are multiple optimization problems inherent in an ambulance system, as mentioned in section 3.1. Previous studies have a varying focus, some have focused on the location and allocation of ambulances, while other studies have focused on the routing aspect of the ambulance service. Little research has yet been done on optimizing the act of dispatching ambulances to events due to the complexity of its implementation. Furthermore, relocating ambulance stations is a costly and politically sensitive issue, and thus one should ideally start with optimizing the allocations themselves to see if adequate gains can be achieved through optimizing only for this. An aspect of the ambulance service that initially showed promise was to optimize the number and location of the ambulance *standby points* specifically; however, they are currently in limited use due to their impact on working conditions. These standby points and the recent history of their use are detailed in section 2.1.3. In addition, it would be interesting to see whether the number of ambulances themselves can be optimized, to see whether there are any significant gains by increasing their number, or even whether their numbers could in some cases be reduced in order to manage resources more efficiently.

1.4 Research method

This thesis uses insights and methods from three main topics: optimization, simulation, and the EMS domain. The three-set Venn diagram in Figure 1.4 is meant to illustrate the relationship between the main topics. It also features the two most influential papers that serve as inspiration for this thesis, namely the works of McCormack and Coates [2015] and Hermansen and Mengshoel [2021a].

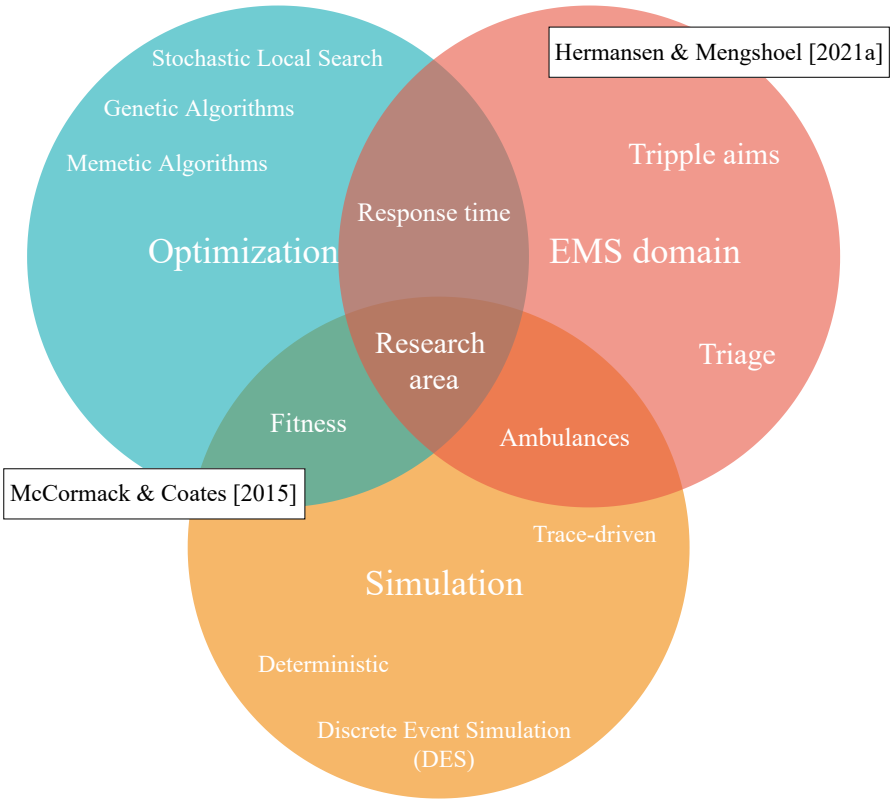


Figure 1.4: The three main topics constituting the research area of this project: optimization, simulation and the EMS domain. Two influential papers McCormack and Coates [2015] and Hermansen and Mengshoel [2021a] are also included.

A simulation model must be implemented using the data set provided by OUH as input. The simulation is based on both previous research implementing similar data-driven simulations of ambulance services, in addition to being based on insights into the specifics of the ambulance system in Norway. The authors visited OUH and had multiple discussions with OUH, OAAD, and EMCC staff to converge on the specific implementation detailed in chapter 5.

To answer RQ2, a working optimization model must then be built that produces results reliably based on the output of the simulation model. The working hypothesis is that this is indeed possible, given that RQ1 is answered adequately first. An approach to solving RQ2 would be to try to build an optimization model using one or more search algorithms.

In this thesis, three families of heuristic search algorithms are studied: local search, global search, and a hybrid model which is a combination of the two. These three families are studied using Stochastic Local Search (SLS), Genetic Algorithm (GA), and Memetic Algorithm (MA), respectively. Each approach,

with the corresponding implementation used, is described in chapter 5. Since each method works differently, they also converge differently. Because of this, in order to have a fair comparison of the three, they are each given the same amount of time to run. Also, all algorithms are stochastic, so running multiple trials and collecting statistics will be an important part of the research. How each experiment will be set up and executed is explained in chapter 6.

1.5 Contributions

The most significant contribution of this work is a simulation model based on the ambulance system of Oslo and Akershus, detailed in chapter 4, in addition to the insights gained from applying a variety of allocation models on this simulation for a variety of scenarios, detailed in chapter 6.

1.6 Confidentiality requirements

This project has two official partners, both of whom have confidentiality requirements. The first being OUH which have provided the EMS incident data set. Anna Hermansen first received the data set for her studies [Hermansen & Mengshoel, 2020] [Hermansen & Mengshoel, 2021b] [Hermansen & Mengshoel, 2021a] in 2020 under an Non-Disclosure Agreement (NDA). The same data set was passed on to the authors of this thesis in 2021 unaltered under the same NDA. Due to the risk of misuse, the data will not be made available.

Norkart has been kind enough to share their routing engine along with proprietary GPS data for computing accurate travel times for ambulances in the simulation. Due to the proprietary nature of both the routing engine and the GPS data, few details about these will be made available to the reader. The use of closed-source software was considered appropriate and a necessary trade-off in this case due to the improvements it affords in time complexity, which will be explained in section 4.3, though the travel model could be easily replaced by an equivalent open-source alternative for verification purposes.

1.7 Disclaimer

This thesis is written during the spring of 2022 and contains formulations and passages and was borrowed from a master's thesis pre-project and literature review written by the same authors in the fall of 2021. In particular, chapter 3 is based primarily on a literary review performed as part of the pre-project. In addition,

certain parts of chapter 1, chapter 2, and chapter 5 are also based on and extend the findings of this project.

1.8 Thesis structure

The thesis is divided into chapters containing sections and subsections, all numbered in the format `CHAPTER.SECTION.SUBSECTION`, which makes for easy reference.

Chapter 1 which you have just read gives a brief introduction to the master's thesis and the context in which it is situated. This chapter serves as a guide for the subsequent chapters.

Chapter 2 introduces the ambulance domain more in-depth and gives an overview and analysis of the data set.

Chapter 3 summarizes other related pieces of research that has been done on optimizing EMS resources, as well as an ontological account of the different ambulance problems.

Chapter 4 runs through the cardinal piece of this project, the *simulation model*, which is the proxy by which the optimal solutions are found for the ambulance system.

Chapter 5 goes in-depth into the computational methods and heuristic algorithms used with the simulation to optimize ambulance allocations.

Chapter 6 details a number of experiments intended to investigate RQ1 and RQ2. This chapter presents an objective for each experiment, a neutral description of the experimental setup used and the parameters and time frame of the simulation, as well as a presentation and discussion of the corresponding results.

Chapter 7 wraps up the master's thesis and discusses whether the research questions are answered and whether there has been any progress toward the overall goal. It also explores the limitations and future work based on experiences acquired while writing this thesis.

Chapter 2

Background

This chapter aims to equip the reader with the necessary background knowledge to comprehend the following chapters. The ambulance domain and the details of the OAAD on which the simulation model is based are explained. The different types of ambulance optimization problems that exist will be presented, as well as some promising metaheuristics to solve them: genetic algorithms, stochastic local search, and memetic algorithms.

2.1 Oslo and Akershus ambulance department

OAAD operates the emergency medical service in the Oslo and Akershus area in close cooperation with the EMCC. Akershus was an independent county in Norway up until 2020 before it was merged with two other counties in the county reformation of 2020. Despite this, the ambulance service still serves the same geographical area of Oslo and Akershus, in addition to an area called the Kongsvinger region.

The ambulance service is a complex service that cannot be easily modeled with a simulation. First, it consists of a variety of different vehicle types: acute ambulances, psychiatric ambulances, transport ambulances, rapid response cars, rapid response cars with a physician ("physician car"), motorcycles, helicopters, planes, boats, and even bikes for densely populated events such as Norway's national day. Furthermore, the system involves many ethical concerns and complex decision-making processes that are required for the system not to become overwhelmed. The following subsections will describe some of these processes, in addition to providing a general overview of the ambulance system in Oslo and Akershus.

Figure 2.1a shows a photography of Ullevål base station, one of several base stations that OAAD uses during day-to-day operations. Next to it is one of the easily recognizable yellow Mercedes-Benz Sprinter ambulances in Figure 2.1b.

(a) Ullevål base station.

(b) Mercedes-Benz Sprinter ambulance outside of Ullevål base station.

Figure 2.1: Photography of Ullevål base station and one of the ambulances managed by OAAD.

2.1.1 Triage

Given the limited number of ambulance workers, ambulances, and base stations available, there will always be an incentive to manage those resources efficiently to save the most lives. An often used tool by EMCC operators for this purpose is *triaging*. When a person calls the EMCC, an operator with medical background quickly assesses the severity based on the conversation with the caller. The operator will classify the incident into three buckets based on this assessment, each with increasing acuteness from the next. The triage hierarchy used in Norway is shown in Figure 2.2. There is also a fourth, implicit, acuteness level, which is given to events that are never responded to.

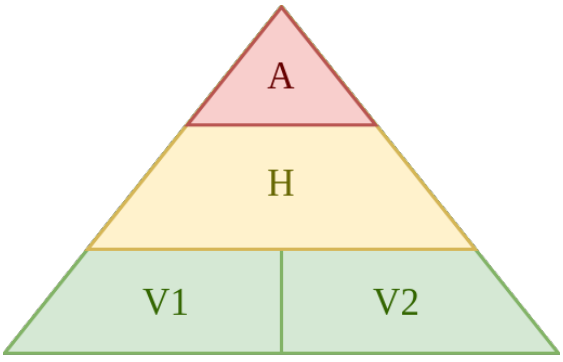


Figure 2.2: Triage hierarchy used in Norway.

The most acute incidents are classified as red or acute (A), then followed by yellow or urgent (H), and finally green, which is divided into unplanned (V1) and planned (V2). During conversations with EMCC and ambulance service employees, it

became apparent that they used colors (as shown in Figure 2.2) instead of codes to distinguish events; however, due to our focus on the data set and its content, we will instead refer to incidents by their code in this thesis. It is worth noting that the triage level is a best-guess and can sometimes be wrong. If an incorrect triage is noticed before the ambulance arrives at the scene, the ambulance is notified and can speed up or slow down according to the new information. When an incident is classified as more acute than it actually is, it is said to be *over-triaged*, and vice versa for *under-triaged*. The distribution of the triage levels, as they appear in the data provided by OUH, is shown in Table 2.1.

Color	**Code**	**Name**	**Incidents**	**%**
Red	A	Acute	242 492	42.9%
Yellow	H	Urgent	216 462	38.3%
Green	V1	Unplanned regular	56 527	10.0%
Green	V2	Planned regular	50 233	8.9%

Table 2.1: Triage levels and their frequency in the data set

The difference between responding to an acute and urgent mission was described to the authors as "whether the personnel had time to tie their shoelaces before entering the ambulance or not". In Table 2.1, the most frequent triage level was acute. This was, according to OUH, in large part caused by deliberately over-triaging events. When faced with uncertainty, the EMCC chooses to triage incidents as more acute to err on the side of caution, as over-triaging is always better for the patient needing care than under-triaging. However, as representatives from OUH pointed out, this could pose a challenge when it is done consistently on a large scale, since more frequent responses classified as "acute" mean that resources also are blocked more often, resulting in delayed responses for subsequent events.

2.1.2 Triple aims

Most healthcare systems today operate on the principle of improving *the triple aims*: the level of care experienced by the patient, the health of the society as a whole, and the overall cost of providing care to the populace [Berwick et al., 2008]. Recent research introduces the notion of expanding upon the triple aims with a greater focus on employee and team well-being, forming the *quadruple aims* [Arnetz et al., 2020]. It is argued that this could help alleviate stress and burnout prevalent among healthcare personnel. Increasing employee well-being is a focus that was considered highly interesting for optimization purposes; however, this was considered outside the scope of this study due to the political and ethical complexity it introduces. It would be interesting to see whether this is also an

objective that could be optimized in addition to the survivability of patients. This would require more qualitative research on the impact of various aspects of the ambulance system on its employees.

2.1.3 Base stations and Standby points

The ambulance service operates 15 base stations in Oslo and Akershus. A base station is a building in which medical equipment and vehicles are stored and where ambulance personnel can rest between missions, much like a fire station. The base stations and their locations in the coordinates of UTM zone 33 are given in Table 2.2.

ID	Name	Region	Easting	Northing
0	Eidsvoll	North	287187	6692448
1	Nes	North	304199	6669959
2	Ullensaker	North	286455	6671754
3	Aurskog-Høland	East	307577	6642937
4	Lørenskog	East	275840	6650643
5	Nittedal	East	270631	6663254
6	Brobekk	East	267085	6651035
7	Sentrum	Mid	262948	6649765
8	Ullevål	Mid	261774	6652003
9	Northern Follo	South	266827	6627037
10	Southern Follo	South	259265	6621267
11	Prinsdal	South	265048	6640259
12	Akser	West	244478	6641283
13	Bærum	West	248901	6648585
14	Smestad	West	259127	6652543

Table 2.2: Base stations in Oslo and Akershus with their associated IDs, region, and location in the UTM-zone 33 coordinate format.

In addition to these stations, Kongsvinger has, as of 2022, three additional ambulance stations currently in use that are part of the ambulance service of OUH. However, the Kongsvinger region was not part of OAAD before 2019. The region is therefore considered outside of scope and its stations are ignored for the purpose of this study.

Because building new base stations is difficult due to limited plot availability and budgetary concerns, the idea of installing cheaper, more lightweight base stations emerged. These stations are called *standby points* and provide stationed ambulance personnel with faster access to patients without providing the full range

of services to ambulance workers that a base station would provide. Services could include, but are not limited to, things such as: heating, toilets, a kitchen, internet, TV, sofas, and sleeping quarters. The standby points currently used are shown in Table 2.3.

ID	Name	Easting	Northing
15	Ryen	265439	6646945
16	Grorud	270248	6654139
17	Skedsmokorset	279154	6657789
18	Bekkestua	253295	6650494

Table 2.3: Standby points in Oslo and Akershus as of 2019.

According to OAAD's own research, the achievement of the response time target has increased by more than 20% since the introduction of standby points in 2016. This highlights the unexplored possibilities and potential improvements that can be made toward improving performance measures by finding new and optimized solutions without necessarily increasing the number of ambulances and resources available. However, an aspect that must be carefully considered is how these standby points affect the working conditions of ambulance workers. In particular, Walsh [2019], concludes that ambulance workers expect a certain minimum standard of facilities on site and the ability to socialize with colleagues during standby to experience a satisfying work environment. Due to negative experiences and inquiries from the Norwegian Labor Inspection Authority, the OAAD phased out the most provisional standby points in favor of more permanent structures with better accommodation for ambulance personnel.

Together, the base stations in Table 2.2 and the standby points in Table 2.3 make up all the locations where OAAD can station ambulances. This is the basis for the simulation described in chapter 4 and consequently the locations that are optimized over, as described in chapter 5. For convenience, let the set of all IDs be called B for base station.

2.1.4 Hospital locations

A list of hospitals and emergency wards used by OAAD is displayed in Table 2.4. It is worth mentioning that some of the services are co-located. These are the hospital locations used in the simulation for delivery locations, as described in chapter 4. Some of these can be observed in the visualization shown in section 4.5.

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

Table 2.4: Hospitals and emergency wards in Oslo and Akershus. Some base stations and emergency wards are co-located with hospitals.

2.2 Data analysis

To accurately model and simulate a functioning EMS system, a data analysis is first performed to identify any trends and insights that could be relevant to incorporate into the model. As mentioned in section 1.2, the data set is not publicly available due to a confidentiality agreement. The event dates are primarily from the time period 2015–2019. The exception is for the years 2001, 2002, and 2005, which also appear in the data with only four incidents in total. 2019 has a significant number of incidents, but it is incomplete, as seen by the lower number of incidents compared to previous years in Table 2.5. These irregular years were removed from the data analysis to not skew the data in any way. Data points from regions outside of Oslo and Akershus were also removed, as discussed in section 1.2. This brought the number of cases down to 565 738.

Year	Incidents	Included
2001	1	✗
2002	1	✗
2005	2	✗
2015	147 880	✓
2016*	185 976	✓
2017	193 086	✓
2018	201 675	✓
2019	26 190	✗

Table 2.5: Number of incidents per year in the data set. The years 2001, 2002, 2005, and 2019 were not included in the data analysis because of lacking data. Asterisk (*) refers to a leap year.

Perhaps one of the most interesting insights from the data analysis is the clear pattern emerging from the hourly demand plots. There is a clear peak around noon, a dip at 5:00 and a plateau around 18:00, as shown in Figure 2.3.
Trends in the data are even more prominent when looking at the average over each day of the week next to each other. There is less demand on weekends than on weekdays, as shown in Figure 2.4. This is likely because there is less planned transportation on weekends. The same pattern appears in the previously mentioned paper by McCormack and Coates [2015], and can also be observed in other domains like activity on social media [Mengshoel et al., 2013]. It is not unreasonable to think that this is correlated with when most people are awake.
Looking at the demand on a per-month basis shows that there are no big differences between the months in terms of average demand. However, the shape of the months differ to some degree, as is made clear by the violin plot in Figure 2.5;

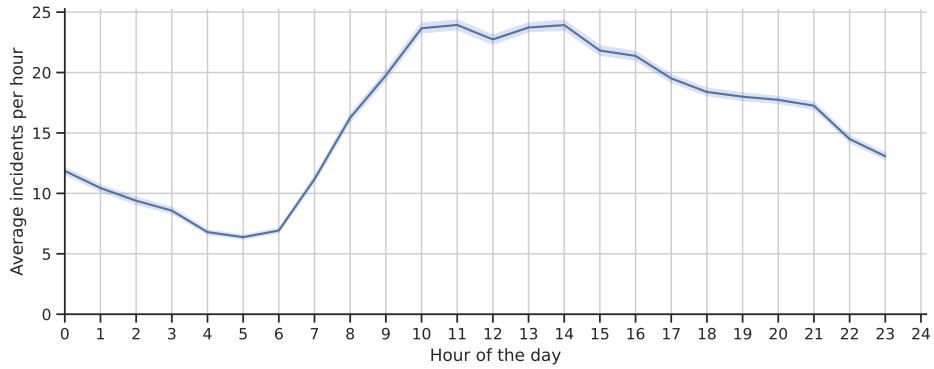


Figure 2.3: Average number of incidents per hour. There is a clear low point at 5:00 and a peak around noon.

the months March–May and October–December show tendencies of being multimodal. Figure 2.6 exaggerates the differences in average demand between months to show which months tend to have higher and which tend to have lower demand. From the figure, the months January, November, and December have a higher than average demand, while March, April, and July have a lower than average demand. December remains the busiest month on average. One reason for this could be the large number of accidents that occur on New Year's Eve, as can be seen in Figure 1.2 on page 2. July is the least busy of the months. The lower demand could be due to the common summer break in Norway that takes place in July.

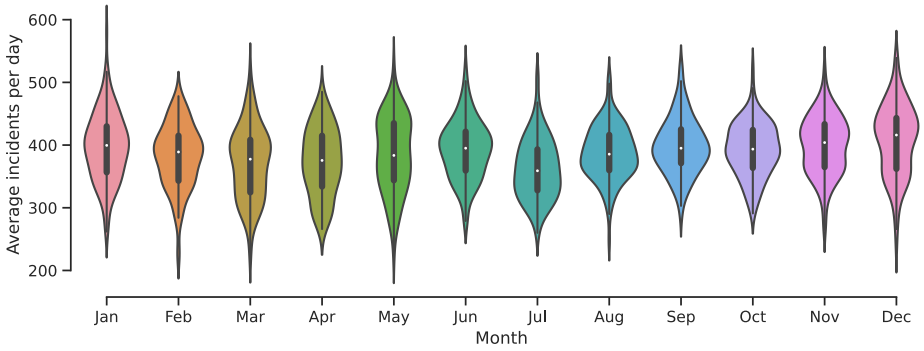


Figure 2.5: Violin plot for each month showing the distribution of average incidents per day

The data set also provides information about the priority level of the incident. 42.9% of the incidents were classified as acute, 38.3% as urgent, while planned

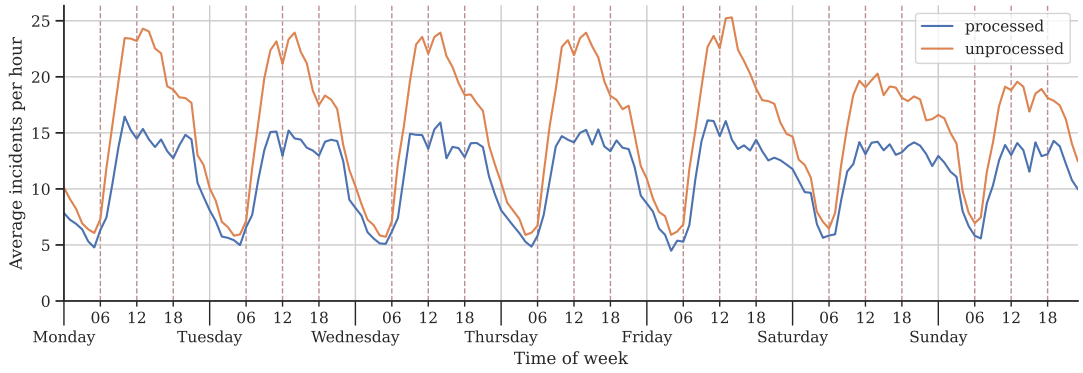


Figure 2.4: Average weekly incidents per hour. In this plot "unprocessed" refers to the full data set containing all types of incident, while "processed" is the result of the data processing performed in section 4.4, containing no regular unplanned or planned incidents.

and unplanned regular incidents account for 18.9% of the incidents in total. In addition, the type of response vehicle used is included, and an overview of their frequency can be found in Table 2.6. In particular, the most common type of vehicle is "Ambulance", which covered 95.2% of all incidents. McCormack and Coates [2015] includes the proportion of rapid response vehicles to ambulances in their chromosome representation, although in this case this vehicle type represents a very small percentage of incidents responded to. Considering that more than 95% of events in the data set are serviced by standard ambulances and to reduce the complexity of the problem, the scope of this study will generalize all these vehicle types as a single "Ambulance" type.

Dispatch type	**Incidents**	**%**
Ambulance	538 791	95.2%
Operations Manager	18 261	3.2%
Response Vehicle (with physician)	8 036	1.4%
Patient transport	547	0.0%
Rapid Response Vehicle	103	0.0%

Table 2.6: Dispatch types and their frequency in the data set

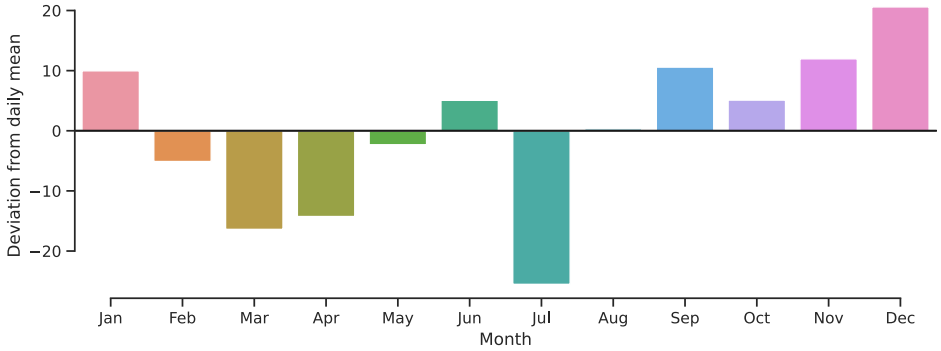


Figure 2.6: Deviation from daily mean for each month.

2.3 Solution space size

In computer science, the set of all solutions to a problem is called *the solution space*. Sometimes, it can be useful to know the size of the search space. If there are relatively few solutions, writing out all of the solutions one by one could be a viable approach. However, for larger search spaces, combinatorics is needed to find the size of the solution space.

Since this thesis will focus on the simulation and optimization of ambulance allocations, knowing the number of possible ambulance allocations would be very useful. Allocating ambulances is the process of determining how many ambulances each base station should have. If ambulances are considered to be indistinguishable from each other and base stations distinguishable, then the number of combinations with replacement can be used to quantify the solution space. This relationship is often referred to in combinatorics as "stars and bars", with the stars representing ambulances and the space between the bars representing the base stations. An illustration is given in Equation 2.1. Note that gaps are allowed to be empty; this corresponds to there being zero ambulances allocated to that base station.

$$\mathbf{x} = \star\star\star\big|\,\star\star\star\star\star\,\big|\big|\,\star\star\big|\,\star\star\star\star\ldots\big|\,\star\star\star\,\big| \tag{2.1}$$

Let the number of ambulances be n and base stations k, then the number of stars will also be n, while the number of bars will be one less than the number of base stations, that is, $k-1$. Finding the solution space size now becomes the same as answering the question: How many ways are there of choosing $k-1$ bars out of $n+k-1$ objects (stars + bars)? The formula for this is given by Equation 2.2.

$$h(n,k) = C^R(n,k-1) = \binom{n+k-1}{k-1} = \frac{(n+k-1)!}{(k-1)!n!} \tag{2.2}$$

Regarding the computation of the actual number, OAAD uses *two* allocations, one during the day shift and another for the night shift. They use 45 ambulances during the day shift and 29 during the night. The number of base stations is assumed to be 19. Then, the total way to set up the two allocations, which is the same as the size of the solution space, is $h(45, 19) \cdot h(29, 19) = 2.588713819 \times 10^{15} \cdot 4.568648126 \times 10^{12} = 1.182692254 \times 10^{28}$.

Using a simulation model, which takes $1\,s$ to complete on average, it would take about $3.750292491 \times 10^{20}$ years to simulate all solutions. This is why more powerful computers or a smarter search algorithm is needed. This thesis will focus on the latter.

2.4 Software tools and resources

Most of the algorithms presented in this thesis, in addition to the simulation software itself, are written from scratch to allow fine-grain control. The software tools and resources presented here were selected based on a combination of project requirements and the author's familiarity with the technology.

Java 17 was used to model the domain and build the core components of the system: simulation, optimization, and execution of the experiments. Java was chosen above other languages for its object-oriented focus, good performance, and its static type system. No 3rd-party code was included in the simulation code itself to maintain a high level of control, although the real-time visualization tool created for the simulation uses JavaFX 17, in addition to P.J. Meisch's Mapjfx library for map visualizations[1].

Python 3.9 was used as the main tool to build the data pipeline, perform analysis, and create visualizations. As a highly popular programming language, Python has a large number of free open-source modules published for everyone to use online. This has been taken advantage of in the course of this project to increase productivity. The modules used in the Python part of the project are described below:

- **pandas** was a very useful tool when working with tabular data in Python.
- **numpy** is bundled with pandas and was mainly used to perform fast array operations.
- **utm** is a module for converting back and forth between the two coordinate systems used in this project: utm and lat/lon.

[1] Mapjfx (Accessed 19.05.2022). https://github.com/sothawo/mapjfx

- **matplotlib** played nicely with the other modules and made visualizing the data easy.
- **folium** handled the geographical plots in this project and was combined with **selenium** to export the graphics from a web browser.

2.4.1 Demographic data

This project has used population data from Oslo and Akershus aggregated into the same $1 \times 1\,km$ standardized cells as detailed in section 1.2. This population data is aggregated by SSB and is publicly available from their website.[2]

2.4.2 Travel time and travel distance estimates

A proprietary route calculation engine, Ferd, was made available to the authors. This routing engine is optimized to calculate paths between multiple origins and destinations, reducing the time complexity and resources required for this calculation. This is further detailed in section 4.3.

[2]Kart og geodata fra SSB (Accessed 05.05.2022). https://www.ssb.no/natur-og-miljo/geodata#Nedlasting_av_rutenettsstatistikk

Chapter 3

Related work

This chapter will define some of the problems within the field of EMS research and present previous research aimed at solving these problems. This is followed by a literature review on relevant topics, giving an overview of recent trends within the cross section of EMS, AI, and resource optimization research.

3.1 Problem ontology

There are multiple optimization problems inherent in ambulance services. This section will define some of these optimization problems, which are frequently examined in the literature, as well as various proposed approaches to modeling and solving these problems.

3.1.1 Ambulance location

The Ambulance Location Problem (ALP) is the problem of determining the most effective locations to place ambulances to serve a population, as well as several other possible conditional criteria [Tassone & Choudhury, 2020]. Early models for ambulance location include Toregas et al. [1971], which were based on minimization with the Set Cover Problem (SCP). This approach aims to find the *least* number of emergency facilities that can cover a given population. In later contemporary literature, it is modeled as the Maximum Coverage Problem (MCP) [Church & ReVelle, 1974], where the number of ambulances available is set to a fixed number of available resources. In MCP, the problem is then modeled such that the *unions* of overlapping ambulance service areas are maximized. Both models focus on *covering*, while more recent literature has focused instead on measuring the resulting patient *survivability* [Erkut et al., 2008] of different location configurations, which is further detailed in section 3.2.4.

Due to the inherent budgetary concerns and ethical complexity involved in deciding which regions should be covered with ambulances and ambulance stations, a reduced focus on this specific problem is placed in this thesis. However, this problem is still highly interconnected with the problem chosen for this thesis (detailed in the next section), and it is important to also consider this problem for long-term optimization of ambulance resources.

3.1.2 Ambulance allocation

Ambulance allocation is a term often used in the related literature to denote the act of distributing a fleet of ambulances or EMS teams over a given set of base stations. Ambulance allocation is more of a tactical challenge in comparison to ALP, which is generally a long-term strategic problem, with the objective of optimizing the long-term placement of static base locations which are costly and infeasible to move. However, as demonstrated in Sariyer et al. [2017], the geographical locations of demand points for EMS services can vary between weekdays and weekends, as well as between different time periods during the day. Although the locations of the base stations themselves are expensive and time consuming to relocate, the distribution of EMS teams and ambulances to the various stations is not necessarily fixed. When demand points are time dependent, the allocation of teams can therefore be dynamically redistributed over the set of base stations in order to better serve the changing demand. In some models, like McCormack and Coates [2015], the act of optimizing the location of base stations and allocating ambulances to them is done simultaneously, as detailed in section 3.2.2. The framework proposed in chapter 5 will focus mainly on this optimization problem as this problem raises less ethical and political questions, considered outside of the scope of this study, compared to some of the other optimization problems.

3.1.3 Ambulance routing

The Ambulance Routing Problem (ARP) is considered a variant of the more general Vehicle Routing Problem (VRP) where the objective is to determine the most effective routes for ambulances to choose, both in emergency situations and for pre-planned missions [Tassone & Choudhury, 2020]. Like ALP, ARP is a problem with many variations, and both are highly related. For ambulance routing, it is paramount to be able to accurately estimate travel times for different routes, as for many emergencies, the response time is directly related to survival probability as detailed in section 3.2.4. Some common methods for estimating travel times in related literature are discussed in section 3.2.5, while possible approaches using state-of-the-art mapping and routing services in combination with machine learning predictions are detailed in section 4.3.

For this thesis, the notion of ambulance routing will be relevant mainly for use in the simulation model. The simulation model will inherently attempt to find an optimal route based on historical GPS data but does not make any assumptions about ambulance crews choosing to respond to multiple incidents in the same dispatching event. This is considered out of scope for this study as this is a highly situational decision, and the data set is not detailed enough to enable such a process to be modeled.

3.1.4 Ambulance dispatching

Ambulance dispatching is the process of selecting which ambulance should respond to an emergency event. In practice, these dispatching rules are often reduced to choosing the current closest ambulance to respond to the event [Schmid, 2012]. Therefore, this problem is also closely related to ambulance routing, as estimated response time is generally the deciding factor for dispatch policies. Recent literature [Jagtenberg et al., 2017] however argues that, contrary to popular belief, this approach should not be considered near-optimal as it found great improvements in deviating from this policy, although the study itself was unable to identify any specific preferable policy.

During our discussion with operators in the OUH EMCC department, it became apparent that the choice of which ambulance to dispatch is a decision that, for now, remains a "human" choice, as the operator must also consider the potential risks of lost coverage for a certain area when dispatching an ambulance stationed in that area. EMCC personnel noted that this choice could be made easier if operators had a way to assess the probability that an incident occurs within a certain area in the near future. This could possibly be achieved through the use of machine learning as studied in Hermansen and Mengshoel [2021b], or through the use of other probabilistic models.

Due to the ethical complexity and human factors involved in this decision, the simulation incorporates a basic nearest-ambulance dispatch policy, although a possible approach using Reinforcement Learning is proposed in section 7.4.

3.2 Ambulance optimization in literature

There has been a lot of research on optimizing ambulance resources, and although many approaches have been proposed, this is such a complex system that there are many interesting areas with little research, often due to the lack of precise data. Early research like Toregas et al. [1971] uses global deterministic optimization approaches like Integer Linear Programming (ILP). Similar analytical models are still used today, though with more realistic features [Brotcorne et al., 2003]. A

common approach in recent literature includes using Mixed Integer Programming (MIP), a variant of ILP with discrete and continuous variables that is applied, among many others, in Leknes et al. [2017], Wang et al. [2020], Boutilier and Chan [2020]. These approaches share the disadvantage of having high time complexity, making them impractical for use in a real-time system that can predict optimal future allocation. Therefore, the focus of this literature review will instead be on metaheuristic methods, in combination with the insights provided by related research that uses such analytical methods.

3.2.1 Evolutionary algorithms

To the authors' knowledge, GAs were first applied to the ambulance location and allocation problems in Aytug and Saydam [2002] with generated synthetic data. Sasaki et al. [2010] applies a GA on data generated with a statistical predictive model based on regression of real-life emergency call data from Niigata, Japan. McCormack and Coates [2015] expands upon this, incorporating real-life data into a simulation model to evaluate its GA approach. Although GAs have been extensively used to model the ambulance location and allocation problems, little research has yet been done on the use of MAs in this field. This seems to be changing as of recent literature – a neighborhood local search is used to improve the solution obtained by crossover and mutation, in Kochetov and Shamray [2021], which focuses on optimizing the redistribution of ambulance teams throughout the day in Vladivostok, Russia. In this study, a *neighborhood* is defined as a feasible solution similar to the original individual, with a team distributed from one specific station to another to see if this can improve the original solution. Furthermore, Zhang et al. [2015] uses an MA for patient transport, modeled as the *Multi-Trip Dial-A-Ride Problem*, a problem similar to *Multiple Depot Vehicle Routing Problem*. These problems are different from the ARP in that they focus on minimizing travel costs for non-urgent transportation rather than response time itself. Behmanesh and Pannek [2021] showed that an MA outperformed GA when optimizing a supply chain network where demand is known, which can be compared to an EMS optimization case where future demand is predicted with machine learning or other probabilistic methods, as proposed in Hermansen and Mengshoel [2021b].

3.2.2 Chromosome representation

It is a common approach to try to solve the ALP with a GA by encoding a set of predefined or generated candidate locations as a binary string to be used as the chromosome for fitness evaluation. When a gene in this string is given a value of one, an emergency center or standby point should be built at that location, and when it is zero, no facility needs to be located there. This approach is used in

both Kaveh and Mesgari [2019], Deng et al. [2021], and Song et al. [2020], among others, to identify the optimal placements of emergency centers.

In both Guimarães and Vinicius Cruzeiro Martins [2018] and Comber et al. [2011], the positions within the chromosome map to a given set of available ambulances (instead of physical locations), with the value of each gene instead mapping to the ID of a specific ambulance base station with a predefined fixed location. This is an approach to solving the ambulance allocation problem, where the solution is constrained by a limited amount of resources available. Kochetov and Shamray [2021] uses a similar approach to this problem, but instead encodes the positions on a chromosome as representing a specific ambulance station and the corresponding gene value that encodes the number of ambulance teams stationed at that station.

It is also possible to combine the approaches described above using *composite chromosomes*. This approach is used in Wang et al. [2020] where the genotype is a composite chromosome divided into two smaller chromosomes of equal size. The positions in the sub-chromosomes encode the same specific disaster sites (which can be interpreted as a cluster of emergency events). The values in the first sub-chromosome encode the number of allocated ambulances to that area, while the values of the other sub-chromosome specifies which hospitals should respond to the disaster site. A slightly more complex approach is used in McCormack and Coates [2015], where the chromosome consists of three sub-chromosomes. The first sub-chromosome encodes the coordinates of the base stations with unfixed locations. The middle sub-chromosome, consisting of only one gene, encodes the ratio of fast response cars to ambulances. The last sub-chromosome is similar to previously described approaches, where the positions map to specific ambulances, and the value of the genes map to the ID of a base station, whose location is given by the coordinates optimized in the first sub-chromosome, instead of being fixed. With this approach of using composite chromosomes, both ambulance location and allocation can be optimized concurrently.

3.2.3 Simulated EMS system

A simulation is often used to estimate the performance of a GA when simple calculations are infeasible to evaluate the fitness of a solution. Yue et al. [2012] introduces a data-driven Discrete Event Simulation (DES) to assess ambulance allocations. In this simulation, requests are generated based on a model, with a myopic dispatch policy in place, ensuring that the vehicle with the shortest estimated travel time is deployed. McCormack and Coates [2015] applies a similar simulation for the London area and innovates with the notion that vehicles will not necessarily respond from their base station in a busy EMS system, arguing that the assumption that ambulances will respond from only a single location

is unsound. In their simulation, when dispatching ambulances, the travel time from the current location of an ambulance to the incident location is evaluated instead of the base-to-scene time. This was also previously touched upon in Zhen et al. [2014], where ambulances could respond directly to a new waiting event after unloading a patient from a hospital.

A similar simulation model is used in Yang et al. [2019], though in their approach they opt to use synthetic data generated by a Gaussian Mixture Model (GMM) in order to quantify the spatial randomness of demand. It is argued that much of the contemporary literature simplifies the spatial distribution of demand by aggregating spatial demand to a set of predefined administrative regions and that the inherent randomness of such a system can be better modeled by a GMM generator.

Kochetov and Shamray [2021] incorporates an estimated traffic flow and current load and capacity of the road network in their simulation, to dynamically redistribute ambulances at certain times of the day. This traffic flow simulation is based on synthetic data generated from statistical parameters for the population, EMS, and workforce in Vladivostok, Russia. Other approaches have been used to provide a more accurate representation of the real-time load variations of the road network, as discussed in section 3.2.5.

3.2.4 Performance metrics

Using an accurate objective function is paramount to the validity of an optimization model. As detailed in section 3.1.1, a lot of contemporary literature use survivability as a performance measure when evaluating different models and solutions, as first introduced in Erkut et al. [2008]. In practice, the fitness function of a metaheuristic algorithm is often the same as the objective function, but not necessarily.

Earlier examples of performance metrics include maximizing coverage, as first introduced in Toregas et al. [1971], and minimizing the standard deviation between response time for different locations, both of which were combined with the Non-dominated Sorting Genetic Algorithm (NSGA-II) in Guimarães and Vinicius Cruzeiro Martins [2018].

A classic performance metric, as described in Amorim et al. [2018], is to evaluate the response time r_i for an event i. McCormack and Coates [2015] uses a variation of this measure based on a binary evaluation of whether or not the response time for a specific event i exceeds a given threshold value, T, which translates into the following performance metric:

$$P_i^r = \begin{cases} 0, & \text{for } r_i > T \\ 1, & \text{for } r_i \leq T \end{cases} \tag{3.1}$$

This performance metric is applied to all life-threatening events, except cardiac arrests. For these types of incidents, a continuous survival function is applied instead that is meant to approximate the survival probability:

$$P_i^c = (1 + \mathrm{e}^{-0.26+0.139 \cdot r_i})^{-1}. \tag{3.2}$$

This function is similar to a measure presented in Erkut et al. [2008] which is based on coefficients calculated by applying logistic regression to historical EMS data. Both studies only model incidents involving cardiac arrests with these survival functions, as most research within this field currently focuses on cardiac arrests. Therefore, quantifiable survival probabilities are readily available for this type of incident, but less effort has been spent quantifying other types of emergency incident. In addition, these coefficients may have a number of factors that impact them, and a survival function based on data from one location may not necessarily apply to another. Amorim et al. [2020] further formalizes the notion of survival functions for any type of event k as an exponential formula similar to those identified to approximate survivability for cardiac arrests. By finding a coefficient m_k and a constant c_k, a survival function can be created to model the survivability of any medical emergency of type k:

$$P_i^s = (1 + \mathrm{e}^{c_k + m_k \cdot r_i})^{-1}. \tag{3.3}$$

In the data set supplied by OUH, only the urgency of the event is provided and not the specific type of emergency event. All events are classified as acute (A), urgent (H), or regular (V). For this specific data set, it might, therefore, be more relevant to look at identifying some generic survival functions for these categories, based on logistic regressions of various representative incident types within each category, rather than individual incident types themselves. Using the general formula suggested in Amorim et al. [2020] (Equation 3.3) the event type k could be assigned to the different urgency levels of the OUH data set, with the survival function representing the average survival for various representative incidents of that urgency level.

If such survival functions could be modeled, the performance of an ambulance system simulation could be calculated by a heterogeneous survival function, consisting of the survival function in Equation 3.3 applied to different the most interesting urgency classes, urgent (H) and acute (A). In addition to the constant c_k and the coefficient m_k, the weighting factor w_k, related to the assessed priority of the class, must also be decided. In the case of acute emergencies, they should obviously have a higher weight than those that are classified as urgent. Discarding the non-urgent events, V1 and V2, from the data set, a heterogeneous survival function with the relevant call emergency categories for optimization, urgent (H) and acute (A), can then be defined as follows:

$$P^s = \frac{w_H \cdot \sum_{i=1}^{n_H} (1 + \mathrm{e}^{c_H + m_H \cdot r_i})^{-1} + w_A \cdot \sum_{i=1}^{n_A} (1 + \mathrm{e}^{c_A + m_A \cdot r_i})^{-1}}{w_H \cdot n_H + w_A \cdot n_A}. \tag{3.4}$$

where n_k is the number of events of type $k \in \{V1, V2, H, A\}$.

Coefficients for cardiac arrest incidents are currently being worked on in Norway by the healthcare sector, as indicated by our discussion with representatives from OUH, however more research needs to be done into other types of emergencies in order to accurately model these. It also remains to be seen whether it is feasible to estimate the survivability for general EMS urgency classes or whether this would invalidate the model.

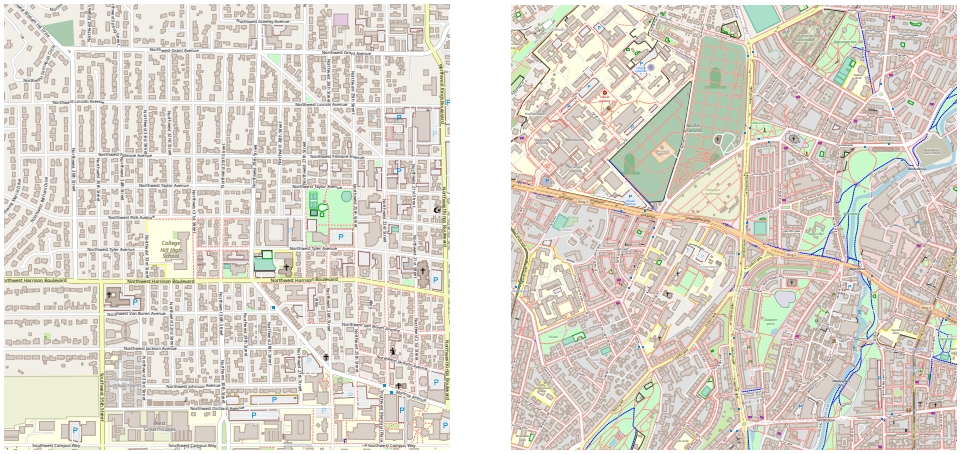
3.2.5 Travel time modeling and routing

To simulate the traffic load and mobility of ambulances, it is necessary to have a robust method to estimate travel times between base stations, hospitals, and incident scenes. In addition, as detailed in section 3.2.4, modern EMS research commonly uses either a survival function based on response time, or the response time directly, as a performance measure. Travel time has a significant impact on response time, which also requires the implementation of an accurate travel time model. However, there does not seem to be any consensus on how this is best achieved.

McCormack and Coates [2015] calculates the travel time between an ambulance and a dispatch event in London using Google Maps to calculate the travel distance between neighboring centroids within a discrete cell grid. An assumption is then made that combining these routes between pairs of neighboring cell centroids, is a reasonable approximation for a route between any two points within a grid. It should be noted that this approach does not include alternative routes that could be taken with a lower traffic load or higher speed limits, particularly for longer distances, though the study includes a correcting factor to compensate for this. A strength of this study is that, by extrapolating and averaging intervals between incident scenes and hospitals, the authors are able to estimate the average speed of ambulances on various days throughout the week. By combining estimated travel distances for neighboring cells with average velocity for any day of the week, they can better model factors such as varying traffic congestion and the general time-dependent nature of travel time.

Less complex approaches are also still in use. Harish Dayapule et al. [2018] uses a deterministic function based on *Manhattan distance* to approximate the travel time between different coordinates in Corvallis, USA. Although this approach approximates the *shortest path*, a more realistic model, as noted by EMCC staff, would also approximate which route is actually *fastest*, based on several factors,

such as traffic load, number of lanes, and road size, among others. Furthermore, a city block-based distance may be an adequate estimate in countries like the USA with high degrees of grid-based urban planning; however, in Europe, this type of urban planning is less common, as illustrated in Figure 3.1. This could make such a model less appropriate for a Norwegian case study.



(a) Corvallis, USA (b) Oslo, Norway

Figure 3.1: Comparison of typical city layouts in the US and Norway. Map data provided by OpenStreetMap.

Going beyond simplified mathematical travel estimation, Amorim et al. [2018] uses Google's Directions API to estimate travel times for its model by pre-computing an Origin-Destination (OD) matrix for different origins and destinations, consisting of centroids on a $500\,m \times 500\,m$ cell-based grid. The Google Directions API is a web service that, among other features, takes as input the origin and destination coordinates and returns the most efficient route using several transportation methods, as well as the distance and duration for that route[1]. This API is also used in Guimarães and Vinicius Cruzeiro Martins [2018] to estimate the response time between base stations and incident scenes. Using the Directions API captures the general time-dependent nature of traffic and the varying driving speeds for different types of roads, and in this study, the search space is reduced by only calculating the distances between a limited set of base stations and incident locations. It is important to note that the algorithm that calculates travel times for the Directions API is, as of 2022, non-deterministic. The output of the model is based not only on speed limits, but also on a prediction based on varying real-time traffic conditions reported by user devices and machine learning. Therefore, the

[1]Getting directions through the Directions API (Accessed 15.11.2021). https://developers.google.com/maps/documentation/directions/get-directions

predictions for a specific weekday made at one point during the year may not be identical to the estimates given for other times of the year as they may inherit variations based on historical, hidden, patterns.

Boutilier and Chan [2020] compares using several machine learning models as estimators for travel time in Dhaka, Bangladesh. By combining GPS data collected from multiple vehicles and census data, they are able to predict travel times between any two points with a random forest model. This estimator is then used in a simulation model similar to that specified in McCormack and Coates [2015], replacing the simple route approximation used for their simulation in London. This newer simulation model also incorporates which routes are appropriate for an ambulance to take, considering that not all streets are wide enough to accommodate a full-size ambulance. The researchers manually predetermined which roads to choose using detailed maps and external consultants. Marla et al. [2021] combines some of the previous efforts by fitting a linear regression model on the historical travel times observed from base stations to incidents and the corresponding travel distance obtained from Google Maps. This requires the location of the ambulance origin to be recorded; however, this is missing in the data set provided by OUH. In Roa et al. [2020] a somewhat similar approach is used to forecast the average travel speed by applying linear regression to the travel times reported by Google Maps and the geodesic distance between different areas of Bogotá, Colombia. This approach has the advantage of not being dependent on any specific type of data set; however, it also has the disadvantage of only reporting "typical" travel times, while ambulances often drive above the speed limit and can pass through traffic faster when responding to an incident.

Chapter 4

Simulation

All the details on the simulation and the basic assumptions made are described in this chapter. The model used for travel time estimation will also be looked at in more detail, as well as the data preprocessing required to tun the simulation using the OUH data set.

4.1 Definitions and notation

Formally, an ambulance allocation can be defined as a list of numbers, where each element corresponds to an allocated number of ambulances and the index corresponding to a base station ID. Therefore, the element at index 0 is the number of ambulances assigned to base station 0, the next base station 1, etc., up to index $k-1$, which is the ID of the last base station out of k base stations. The sum of the elements in the allocation must always be equal to the number of ambulances available n. All allocations that follow this definition are considered solutions to the ambulance allocation problem. Equation 4.1 below defines exactly such an allocation $\mathbf{x}$, which is the same representation used by Katoch et al. [2021].

$$\mathbf{x} = (x_0, x_1, \ldots, x_i, \ldots, x_{k-1}), \quad x_i \in [0..n-1], \quad \sum_{i=0}^{k-1} = n \tag{4.1}$$

As noted in section 2.3, OAAD operates with a day shift and a night shift with different numbers of vehicles, and therefore the simulation also supports two allocations as input. These allocations follow the same structure as the definition given in Equation 4.1, but are not required to use the same number of ambulances. Let Equation 4.2 define a tuple $\mathbf{X}$ that contains m allocations. This will be referred to as a composite allocation from now on.

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m) \tag{4.2}$$

4.2 Simulation model

A simulation model was chosen for the purpose of evaluating ambulance allocations, as this is a viable approach that has been used numerous times in previous research, as shown in section 3.2.3. This section will provide a detailed overview of the simulation model and its implementation for the purpose of this study.

4.2.1 Discrete Event Simulation (DES)

A dynamic Discrete Event Simulation (DES) model was chosen as the most appropriate for simulating the EMS system, based on its previous use in related research such as Yue et al. [2012], Zhen et al. [2014], Jagtenberg et al. [2017], among many others. Formally, the simulation is a discrete state model, which means that the state of the simulation is assumed to not change between events, in addition to being a discrete time model, meaning that the state is only defined in certain instants [Jain, 2017]. Additionally, the simulation is trace-driven, meaning that the system's input, in this case the queue of historical incidents, is based on time-ordered records of observations made on a non-simulated system. This eliminates the possibility of ignoring or estimating correlations incorrectly in the data, and alternative scenarios with incidents of varying characteristics can still be simulated in practice through various methods, as detailed in Banks et al. [2010, p.538]. It should also be noted that, in our implementation, the simulation is deterministic, which means that for any composite allocation $\mathbf{X}$, it will always produce the same result.

4.2.2 Assumptions

A number of assumptions have been made when modeling the EMS system to make the simulation feasible. These include the following:

Assumption 1 Only acute (A) and urgent (H) incidents are simulated, and both are prioritized equally when dispatching. Once an ambulance has been dispatched for an incident, it cannot be re-dispatched.

Assumption 2 Ambulances currently on duty will not be replaced or returned to the base during shift changes, except when the next shift has fewer ambulances

allocated to the base station, where the ambulance is assigned (for example, when switching from day to night shift).

Assumption 3 Ambulances will drive at the expected speed reported by the route calculation software.

Assumption 4 Ambulance personnel will not take regularly scheduled breaks.

Assumption 5 Incidents with identical timestamps and destinations are considered the same incident.

Assumption 6 All types of vehicles in the data set are treated as if they were an ambulance and follow the same travel time model.

Some of these assumptions would need to be reworked to create a more realistic model, but the current level of realism was considered adequate for the purpose of this study due to the uncertainty involved in the various processes. In particular, assumption 2 can be unrealistic in many cases, considering that emergency vehicles can be exempt from standard speed limits in the case of an emergency[1].

If we had access to the destinations or origins of the various vehicles, we could have interpolated an average distance as in Marla et al. [2021], or we could have added an appropriate correcting factor to the estimates made by the route calculation model. However, this is not present in the data set, as detailed in section 2.2, and therefore the estimated travel time for normal traffic is as close as we can get.

Furthermore, Assumption 4 will obviously not apply in a real-life system. This specific assumption is introduced because there is a high amount of uncertainty involved in when a specific ambulance team will be assigned to breaks, and in the case of an emergency, a team might be held up for a long period of time. These breaks could be simulated using a naive approach as the one used for shift rotations, detailed in section 4.2.5, however, this would require making potentially invalid assumptions about how many ambulances could be assigned to breaks at a certain time. During our conversations with EMS personnel, it became apparent that the assignment of ambulance teams to breaks is highly situational and based on the same level of complex decision-making inherent in dispatching, which is detailed in section 3.1.4. As a consequence, breaks were considered outside the scope of this study.

[1]Lov om vegtrafikk (vegtrafikkloven) (Accessed 01.06.2022). https://lovdata.no/lov/1965-06-18-4

4.2.3 Implementation

The pseudocode for the main loop of the simulation is shown in algorithm 1. The input of the algorithm is a set of ambulance allocations $\mathbf{X}$ in the format described in section 4.1. The simulation output is a set of response times, $\mathbf{r} = (r_0, r_1, \ldots, r_{max})$, corresponding to the events queued by the *initializeEventQueue()* procedure. Note that certain parts of the code, in particular, the day-shift handling, have been omitted in order to keep the code concise, though their implementation is discussed later. Furthermore, the simulation model is configurable with a set of parameters $\boldsymbol{\theta}$ as shown in algorithm 1, with the default parameters detailed in Table 4.1.

4.2.4 Warm-up buffer

The simulation includes a warm-up buffer, configurable as part of $\boldsymbol{\theta}$, inspired by the simulation model implemented in McCormack and Coates [2015]. This warm-up buffer ensures that the simulation is allowed to run for a certain time period in the beginning, before the allocation is evaluated, in order to make the simulation less sensitive to the initial state. Considering the continuous nature of an EMS system and the level of demand observed in the dataset, it would be somewhat unrealistic for the ambulances to all be stationed at the base stations at any point (such as in the initial state).

4.2.5 Shift rotation

As noted in section 4.2.3, the pseudocode omits some details of the shift-changing process implemented to model the changing number of ambulances active throughout the day. In algorithm 1, *initializeAmbulances* on line 4 is a procedure that generates a set of ambulances allocated to their specific base station, with the current initial shift assigned to a variable and the off-duty shift kept in memory. The *setCurrentShift* procedure on line 7 will then take the current time reported by the currently handled event and rotate these shifts during the first event the time passes either 08:00 or 20:00 (with default parameters, as noted below). The implementation chosen for this study, as noted in Assumption 2, will remove or add ambulances from the current shift, based on the difference between the number of ambulances currently assigned to the base station during the current and the next shift. If there are fewer ambulances assigned to a specific base station in the upcoming shift and not enough vehicles available to remove from the shift, this difference is kept in memory and handled whenever an ambulance assigned to this specific base station finishes responding to an incident, as noted in section 4.2.6. This is to simulate that an ambulance crew would obviously finish their currently assigned event before retiring. Even though their shift is over, ambulance crews

Algorithm 1: Discrete event simulation

Input : allocations $\mathbf{X}$, configuration parameters $\boldsymbol{\theta}$
Output: list of response times $\mathbf{r} = (r_0, r_1, \ldots, r_{max})$

```
1  function Simulate(X; θ)
2      event ← ∅, t ← ∅
3      r ← ( ), C ← ∅, Q ← initializeEventQueue()
4      ambulances ← initializeAmbulances(X)
5      while Q.isNotEmpty() do
6          event ← Q.pop(), t ← event.time()
7          ambulances ← setCurrentShift(t)
8          switch event do
9              case NewCall do
10                 ambulances ← dispatch(event)
11                 if |ambulances| > 0 then
12                     Q.add(SceneDeparture(t + event.duration, event))
13                 else
14                     C.add(event)
15             case SceneDeparture do
16                 append event.responseTime onto the end of r
17                 foreach transport ∈ event.ambulances do
18                     transport.doTransport()
19                     Q.add(JobCompletion(t +
                           event.transportTime, transport))
20                 foreach nonTransport ∈ event.ambulances do
21                     nonTransport.flagAsAvailableOrFinishShift()
22                     Q.add(LocationUpdate(t + Δt, nonTransport))
23                 CheckQueue(C)
24             case JobCompletion do
25                 event.ambulance.flagAsAvailableOrFinishShift()
26                 Q.add(LocationUpdate(t + Δt, event.ambulance))
27                 CheckQueue(C)
28             case LocationUpdate do
29                 event.ambulance.updateLocation()
30                 if event.ambulance.isNotAtDestination() then
31                     Q.add(LocationUpdate(t + Δt, event.ambulance))
32     return list of response times r = (r_0, r_1, ..., r_max)
```

Parameter	Default value	Description
Start date (incl.)	07.08.2017 00:00:00	Starting time of the simulation.
End date (excl.)	14.08.2017 00:00:00	Ending time of the simulation.
Warm up buffer	4 hours	Time before start date which the simulation runs without tracking response time.
Day shift start	08:00	When the day allocation takes effect.
Night shift start	20:00	When the night allocation takes effect.
Dispatch policy	Fastest first	Which ambulance is dispatched.
Δt	5 minutes	Time interval between when the locations of ambulances are updated, when they are returning to base station.

Table 4.1: The simulation configuration parameters $\boldsymbol{\theta}$ and their default values.

may be required to service even more incidents in the case of an emergency (as per usual overtime work), but in order to simplify the model, the simulated ambulances will instead return to their base station as soon as they finish their current job if their ambulance base station has too many ambulances in use for the current shift.

Note that in the current implementation, shift rotations will not occur immediately when the shift rotation time passes. This is due to the discrete-time property of the simulation, as noted in section 4.2.1, which means that shift rotations will be handled at the time of the next event occurring *after* the time set for a shift rotation in the parameters. Based on an assumption that the events occur at such a regular interval that this should not create a discrepancy greater than a couple of minutes, this was considered advantageous by the authors in order to capture the natural small variations in shifts starting and ending that are inherent in a time-critical system like this. The shift rotation could however be set to occur precisely at the time defined for the shift rotation in $\boldsymbol{\theta}$ by inserting synthetic events that occur exactly at these times into the event queue.

4.2.6 Simulation event types

The proposed trace-driven simulation consists of several asynchronous events that are handled in a main loop. This section will detail the specific procedures that the simulation follows for each event.

NewCall The historic number of transport ambulances and non-transport ambulances is assigned to the incident. If the demand for ambulances is greater than the supply at the time the event is processed in the event queue, an additional NewCall event is inserted into the call queue with the demand of the original event replaced with the residual ambulance demand. This partially processed call is then handled again at a later time in the simulation when ambulances become available in order to dispatch the remaining ambulances that are required.

SceneDeparture At the time of scene departure from an incident, one of three actions may occur:

1. If the ambulance was assigned transport duties, it will find the nearest hospital and begin a journey to this location. A JobCompletion event is then created that schedules the ambulance to arrive at a certain time, based on the estimated travel time.

2. If the ambulance was not assigned transport duties, the ambulance will instead be marked as available and will begin moving back to its base station. This is handled by inserting regularly scheduled LocationUpdate events into the queue, which are processed at a certain interval, where for each event, the ambulance will update its location and check if any NewCall events are queued. This is further detailed in section 4.3.2.

3. Finally, if a shift rotation has recently occurred, as detailed in section 4.2.5, and the corresponding base station of this ambulance in the current shift has fewer ambulances assigned to it but too many ambulances were busy servicing incidents during the time of the shift rotation, the ambulance may now return to base without responding to new events en route in order to bring the number of assigned ambulances down to the currently assigned amount of vehicles. This action is handled within the *flagAsAvailableOrFinishShift()* procedure in algorithm 1 on line 21 and 25.

JobCompletion This event is inserted into the queue when an ambulance departs for a hospital with a patient, and is processed when the ambulance reaches the hospital and has handled the patient. This event type is also pushed to the

event queue whenever an event lacks a scene departure time, where it is then assumed that the incident response was canceled or the patient did not need to be transported to the hospital, at least immediately. The handling of this event type is similar to SceneDeparture, and whenever this event is handled, the simulation can choose to perform either action 2 or 3 from SceneDeparture, which is detailed above.

LocationUpdate This event ensures a step-wise update of the location of the ambulances upon returning from a mission, either from the incident scene itself or from a hospital or other emergency unit. This is necessary to make ambulances dispatchable on their way back to the base station. NAKOS and AMK confirmed that this way of dispatching was close to what they were doing realistic. The LocationUpdate-event can call itself until the ambulance has reached its destination, or is assigned to a new call based on a time interval Δt. Simulation-wise, this makes this step synchronous because of the clock-based update cycle.

4.2.7 Simulated dispatch policy

While AMK uses a combination of nearest-first and fastest-first, this simulation uses fastest-first as the dispatch policy based on the travel time data provided by the route calculation software, as detailed in section 4.3. The term "fastest time", means that the ambulance with the shortest expected travel time to the location of the incident is selected. The same policy is then used when selecting a hospital for transport ambulances as well.

4.3 Travel time estimation

Since simulated ambulances should be able to respond from their current location when returning to base, an extensive travel time estimation must be implemented in order to calculate simulated response times and in order to determine the closest ambulances and hospitals to any incident. Although the resolution of incident locations in the data set provided by OUH is discretized into a grid $1 \times 1\,km$, there are still 2598 cells with historical incidents within the Oslo and Akershus area. Adding the known coordinates of ambulance stations and hospitals to this list amounts to a total of 2624 origins and destinations that should have travel costs estimated between each other. Calculating a distance matrix for these cells amounts to $2624 \times (2624 - 1) = 6\,882\,752$ operations to create a general simulation model that can be applied to any time period within the data set. If various time periods and potential variations caused by the inherent machine learning in Google Directions (as discussed in section 3.2.5) are to be considered, this matrix would

need to be calculated multiple times. This makes it somewhat infeasible to use relatively expensive APIs such as Google Directions for the purpose of this study.

Another possible approach, as detailed in section 3.2.5, is to approximate travel times by fitting a machine learning model on a set of known travel times between different coordinates on the grid, using starting and ending coordinates as input and travel time as the target for the model using a similar approach to Marla et al. [2021]. In that study, the model is using the distance between an observed set of base station and incident scene locations calculated with Google Maps as input and the historical response time for the events as target. The originating location of the ambulance is not present in the data set from OUH however, which necessitates the use of some other data to train a potential machine learning model.

Instead, a proprietary graph-based engine, Ferd, created by Norkart, was lent to the authors for the purpose of this study. This model could be replaced by any open source model equivalent in precision, though this would require validating the performance of these estimates. Ferd offers optimized tools for generating an Origin-Destination (OD) matrix for route costs using the concept of *highway hierarchies*, further detailed in Schultes et al. [2008]. The data used in Ferd are based on data collected by GPS service providers over a period of time. The OD matrix is calculated by finding a set of road network junctions close to the centroid of each relevant cell of the grid. Because of this, there is a minor discrepancy between the centroids of the grids and the actual geographical point that a route is calculated to or from. A comparison of grid centroids and these road junctions used by Ferd can be seen in Figure 4.1. A visualization showing the variation in the travel time from Ullevål hospital can be seen in Figure 4.2.

4.3.1 Route interpolation

The route calculation software is unable to calculate distances to or from certain cells within the grid, mainly due to road restrictions and missing connections. Therefore, several strategies are used to interpolate these. The first step involves inverting the path and approximating the route cost by substituting with the same route run in reverse. Approximately 50% of all missing origin-destination pairs can be approximated using this method. For the remaining pairs, several strategies are involved, including finding all neighbors of a grid coordinate within a radius of 6 km and calculating the mean of all routes between these neighbors and the destination grid. If no such routes can be found, the same procedure of finding neighbors is used for both the origin and destination grid, with the resulting travel time consisting of the mean of all such distances between the origins and destinations of neighboring grids.

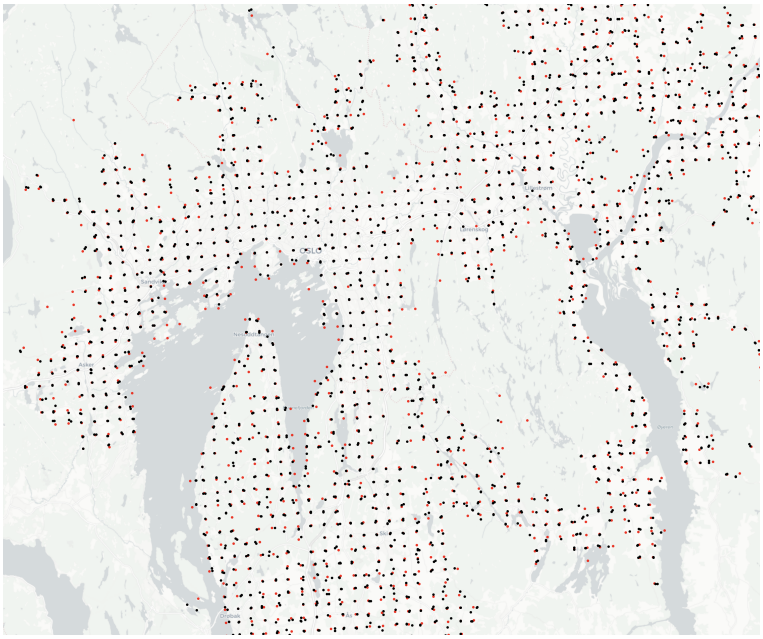


Figure 4.1: Map of Oslo and Akershus overlaid with the centroids of grids with historical events (in red) and nearby junction points used by Ferd to calculate the travel matrix (in black).

4.3.2 Location updates en route

For the simulation model, it is necessary to update the locations of the ambulances that are returning to their base station every 5 minutes, as they should be able to respond to nearby incidents from their *current location* without first returning to the station. This somewhat complicates the travel-time modeling, as the ambulances can choose to travel through grids without any previous historical events. A list of grid centroids exported from SSB shows that there are, in total, 6469 grid cells within the total area of Oslo and Akershus. The authors considered this to be too computationally expensive, even for an optimized model such as Ferd.

Another option would be to calculate the specific routes for all 6,882,752 origin-destination pairs. However, this would require a lot of processing and working memory. An approximation is used instead, where every five minutes, the ambulance is snapped to a nearby grid centroid with the closest travel time to the currently remaining travel time (calculated by subtracting five minutes for every location update).

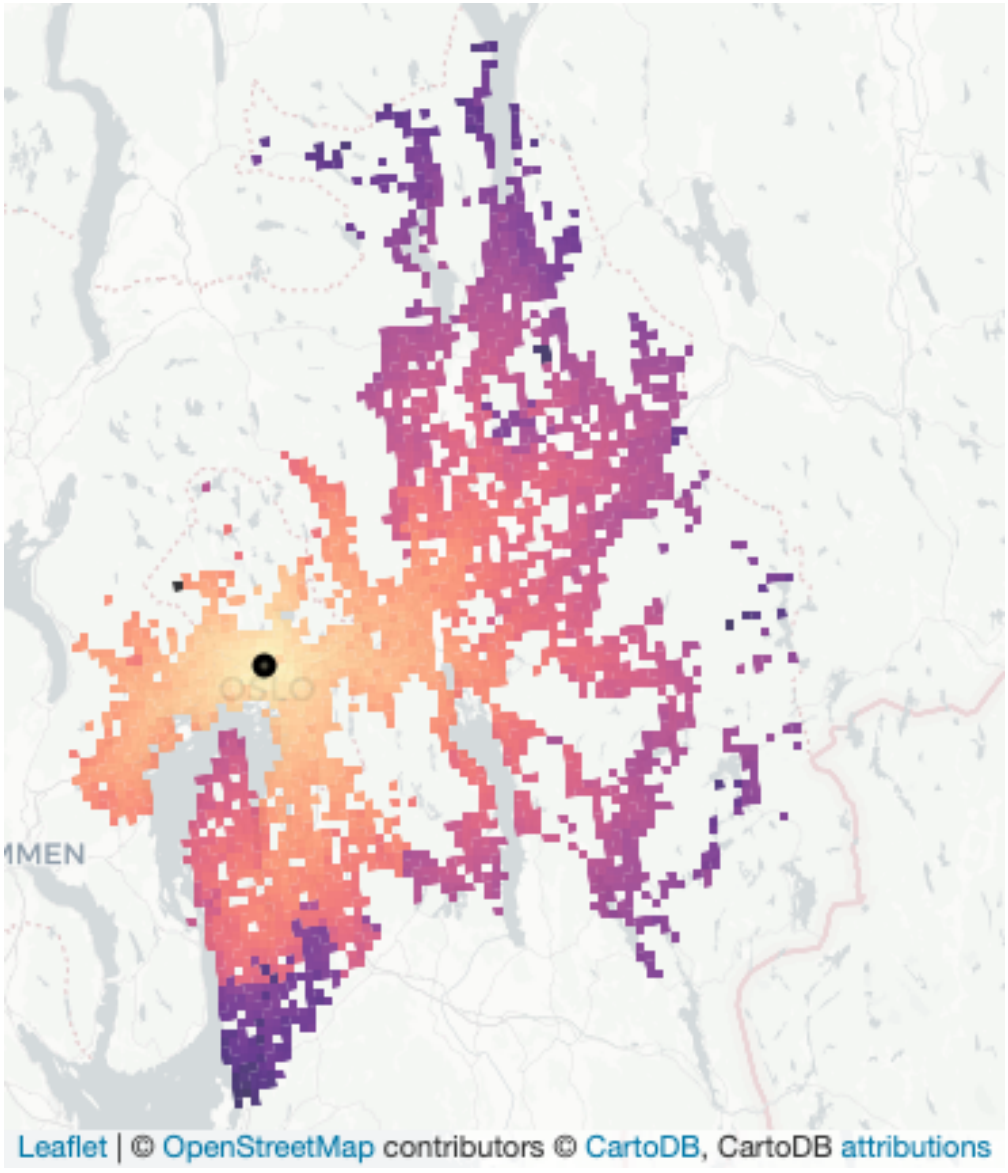


Figure 4.2: Heatmap showing the time it takes from Ullevål hospital to reach other locations using Ferd. The black circle indicates the location of the hospital (origin) and the more purple the color becomes, the further away it is.

4.4 Data pipeline

In order to run experiments using the provided raw data set, a considerable amount of data pre-processing was needed. The reason for this was two-fold: data quality and experiment requirements. Although some fields in the data set are automatically tracked from the incident response system, others have to be entered manually by ambulance personnel or resource coordinators after or during a mission. The authors discovered several defects while working with this material. Sometimes errors had a minor impact, such as, for example, supplying the "Ambulance" vehicle type when actually a car was used instead, and other more critical mistakes, such as entering a wrong date. The latter yielded negative response times in some cases during testing. (Which would be very good for patients if possible!) Because of this, some rows in the data set had to be dropped, while some were fixed if possible. The details of this process are described in the following sections.

The main steps in the data pipeline and how they affect the size of the original dataset is visualized by the funnel plot in Figure 4.3.

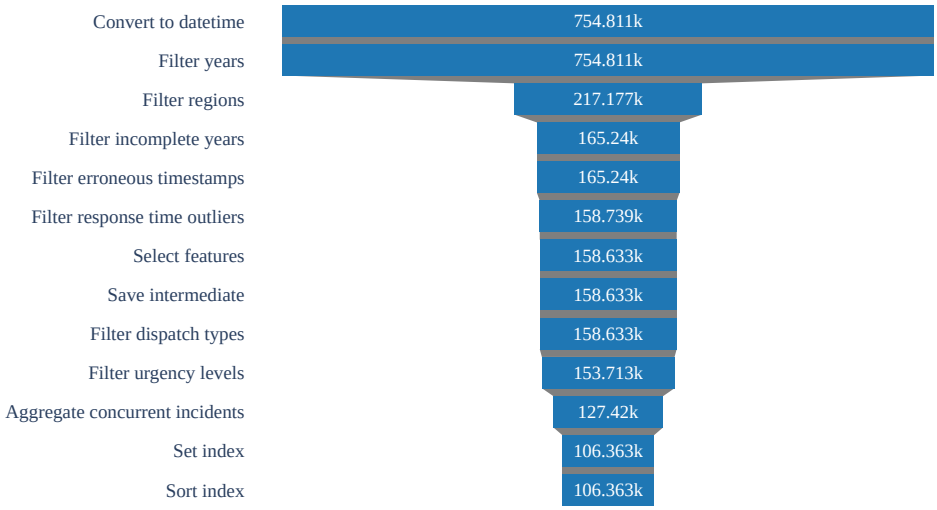


Figure 4.3: Data pipeline with the associated steps and row sizes presented as a funnel chart.

4.4.1 Incident aggregation

In order to keep track of which dispatches belonging to the same incident, similar dispatches were aggregated. Incidents that were created at the same time while also having the same urgency level and destination were considered concurrent

incidents. The number of units dispatched per incident was added as a new feature to keep track of demand. This number was then split into two: transporting units and non-transporting units. This was based on whether the "departure-from-scene" feature was present.

4.4.2 Structural corrections

These corrections to the data set are trivial, but left for completeness. The raw data set would not easily be processed by the Python package pandas due to some inconsistencies in the CSV header, namely, the "id" column was empty and one column had an excess comma. Furthermore, all timestamps were parsed following the standard format: `YYYY-MM-DD HH-MM-SS`. These structural corrections were made in a one-shot script before the rest of the pre-processing steps mentioned above were run.

4.4.3 Exclusion of green missions

The non-urgent unplanned (V1) and planned (V2) incident types have been excluded from the dataset used in the simulation, as these were considered too complex to model accurately using the current level of detail. Since these incidents do not need to be served immediately, a dispatcher can choose to delay dispatch for these incidents if there is a high number of acute and urgent incidents currently in queue for dispatch, as observed by the authors during their visit to the EMCC. OUH personnel noted that in many situations some of these transportation missions are used for scheduled operations, which can be costly and difficult to reschedule, in which case the policy mentioned above may not be applicable. Therefore, these missions were excluded on the same basis of complexity as break assignments (section 4.2.2) were considered out of scope for.

4.5 Visualization

A real-time visualization tool was written in JavaFX, using the Mapjfx library described previously in section 2.4. Figure 4.4 shows a screenshot of the visualization with added labels that indicate what the different components on the screen represent. This tool is used to visually verify that the simulation is working as intended.

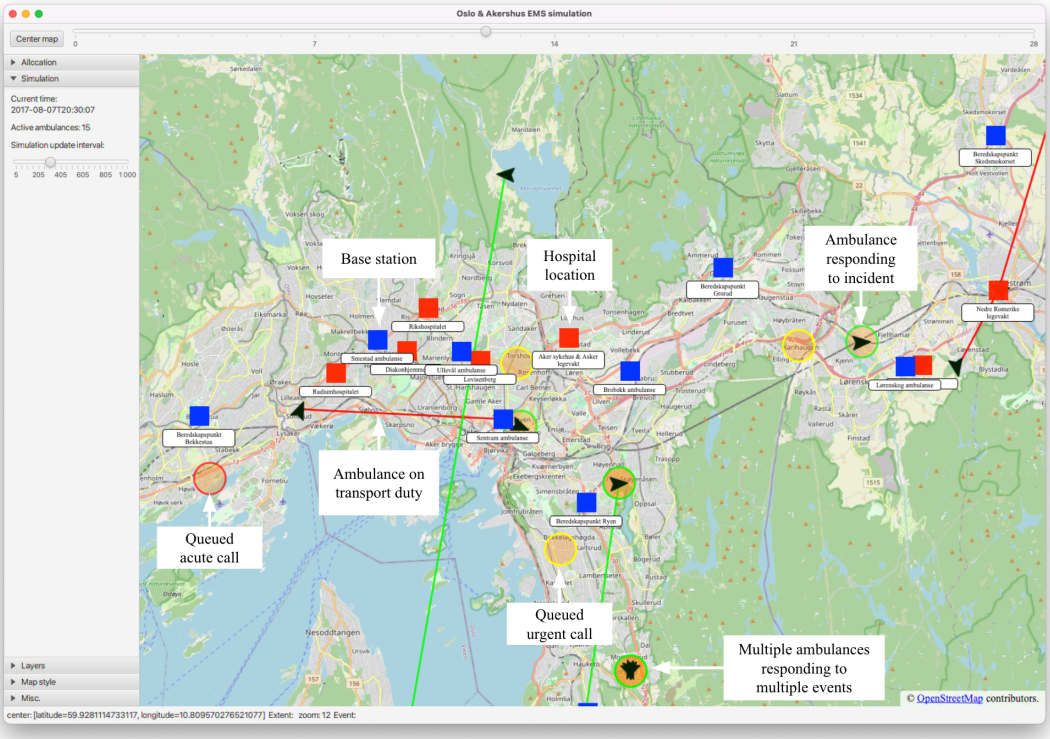


Figure 4.4: Simulation visualization. Screenshot of the simulation running in visualization mode with added markers (in white). Not shown in this screenshot are the blue lines representing an ambulance returning to its base station. Acute and urgent incidents are differentiated in the visualization, even though their prioritization remains the same.

4.6 Verification and validation

To ensure that the simulation model works as intended and does not have substantial flaws, some form of quality control was necessary. Generally speaking, verification refers to *internal* quality control mechanisms, while validation is aimed at an *external* entity. As for this project, verification has been the activity of checking that the code works as intended, e.g., the code compiles, there are no bugs, dates are formatted correctly, random procedures are indeed random, and deterministic procedures are indeed deterministic. Validation has been the activity of checking whether the results and behavior of the software make sense. An example of when the software did not behave as intended was an event earlier in the project, when negative response times started appearing in the plots. This event emphasized the importance of why continuous verification and validation was important for this

project, even though the simulation was not intended to be used in production by OUH.

In the early days of the project, spotting obvious mistakes, such as the negative response times mentioned above, was done mainly using text-based sources (terminals, CSV files, and logs), checking the output and concurrency of events in the simulation. The later stages of validation included more visual tools, such as graphs and a visualization tool, described previously in section 4.5, to inspect the behavior of the system.

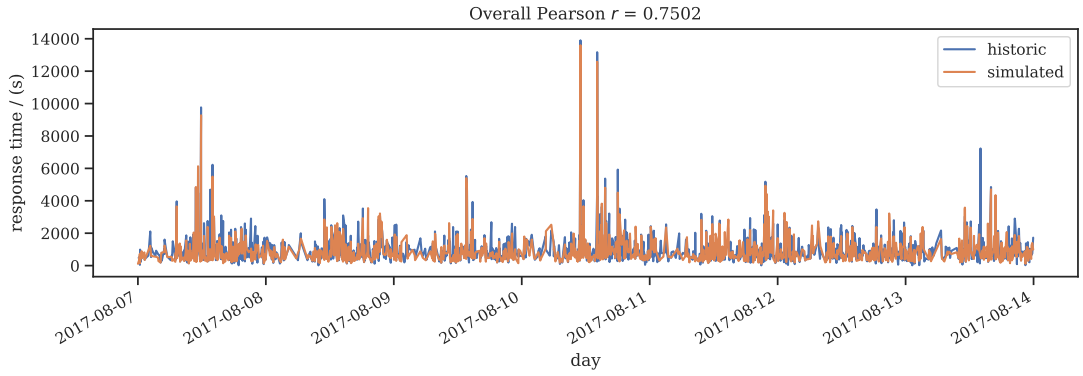


Figure 4.5: Historic vs simulated response times for week 32 August 2017. The two time series has an overall Pearson correlation coefficient r of 0.7502.

Another way to validate the simulation is to overlay the simulated response times on the historical ones. This is what Figure 4.5 is showing; response times for week 32, August 2017 yielded a Pearson correlation coefficient of $r = 0.7502$. This is an indication that the simulation is not too far from reality. Striving towards a higher correlation would not necessarily be helpful or desirable, as there are many essential pieces of information missing from the historic data which would be needed to reconstruct what happened more accurately. This would be things such as: detailed information about shift changes, when breaks occurred, historic dispatching decisions, which allocations were used, and so on. The point being that the simulation will probably never reach 100% accuracy. In addition, the simulation is meant to try new scenarios, which are *supposed* different from the past. Said differently, a high enough correlation is important for the validity of the model, but it is very hard to know all the factors that went into generating the data set.

Chapter 5

Optimization

Included in this chapter is the formal definition of the optimization model, an allocation, a composite allocation, and the fitness function using the simulation from chapter 4. The remaining sections focus on the specifics of each optimization algorithm and how these are implemented in this project.

5.1 Optimization model

In order to do optimization, the optimization function F needs to be defined. Let F be defined by Equation 5.1 as the average response time from the list of response times $\mathbf{r} = (r_1, r_2, \ldots, r_i, \ldots, r_{max})$ returned by the simulation model for a given composite allocation $\mathbf{X}$. This will be referred to as the fitness function from now on. However, since a lower average response time is preferable, the fitness function needs to be minimized, not maximized, which is conventionally done. Both functions are defined in Equation 5.1 below. Since the simulation is parameterized by the configuration parameters $\boldsymbol{\theta}$, the fitness function necessarily needs to include the same parameterization.

$$F(\mathbf{X}; \boldsymbol{\theta}) = \frac{1}{|\mathbf{r}|} \sum_{i=0}^{|\mathbf{r}|} r_i, \quad \text{where } \mathbf{r} = \text{Simulate}(\mathbf{X}; \boldsymbol{\theta}) \tag{5.1}$$

Optimizing this function would mean that $\mathbf{X}$ must always follow the definition given earlier in Equation 4.2, which requires each individual allocation $\mathbf{x}$ to sum up to the exact number of ambulances defined for each allocation. This problem would fall into the category of a *constraint optimization problem* (COP) in Table 5.1 below, since both *constraints* and an *objective function* (F in this case) are present. To ensure that the constraints of $\mathbf{X}$ are always satisfied, one might design special operators that ensure that $\mathbf{X}$ always satisfies the constraints by checking and

repairing the solutions if necessary. This process of checking and repairing can decrease the performance of the algorithm. This is why changing the representation of the solution to transform the problem into a *free optimization problem* (FOP), which is easier to solve, can be useful. This was the case for this thesis, and the next section will explore this transformation in more detail.

	Objective function (Fitness function)	
Constraints	Yes	No
Yes	Constrained optimization problem	Constraint satisfaction problem
No	Free optimization problem	No problem

Table 5.1: Different types of optimization problems. Reproduced from Eiben and Smith [2003, p.7].

5.1.1 Solution encoding

Encoding the solution to have fewer constraints can be done by changing the representation from cardinal to ordinal numbers. Instead of an allocation being a list of the number of ambulances at each base station, let it alternatively be a list of base station IDs assigned to each ambulance. The set of base station IDs B, and their mapping to the physical world, was previously defined in section 2.1.3. Let this encoded representation $\mathbf{g}$ be defined by Equation 5.2, and the composite version $\mathbf{G}$ by Equation 5.3.

$$\mathbf{g} = (g_0, g_1, \ldots, g_i, \ldots, g_{n-1}), \quad g_i \in B \tag{5.2}$$

$$\mathbf{G} = (\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{m-1}) \tag{5.3}$$

In the GA literature, an encoding for a solution, such as $\mathbf{g}$, is called a genotype, and the elements simply as *genes*, while the solution $\mathbf{x}$ is called the phenotype of the solution. As long as there exists a mapping from a genotype to a phenotype, optimization algorithms are free to use $\mathbf{g}$ in the optimization step. This mapping m from a given $\mathbf{g}$ to a solution $\mathbf{x}$ is given by Equation 5.4 below, which uses Iverson bracket notation to count the occurrences of the base station IDs.

$$m(\mathbf{g}) = \mathbf{x} = (x_i)_0^{k-1}, \quad x_i = \sum_{j=0}^{n-1} [g_j = i]$$
$$m(\mathbf{G}) = \mathbf{X} = (m(\mathbf{g}_0), m(\mathbf{g}_1), \ldots, m(\mathbf{g}_{m-1})) \tag{5.4}$$

For composite genotypes $\mathbf{G}$, the mapping is applied independently for each element $\mathbf{g}$. This new representation enables free optimization of the fitness function, since the constraints are handled implicitly by the representation as long as the initialization of the genes is done correctly and the search operators provide closure over B.

5.1.2 Software implementation

The actual implementation of the optimization model is done with two primary components: the simulation and the optimizer. The simulation is described in detail in chapter 4 and is by far the most complex part of the system. The optimizer has a reference to the simulation and uses it each time a new fitness evaluation is needed. Because the simulation is deterministic, it can be treated as a pure function by the optimizer once it has been initialized with the distances and incident data. The optimizer uses this fact to build a picture of which solutions produce the best fitness values. Figure 5.1 gives a visual representation of how optimization is implemented.

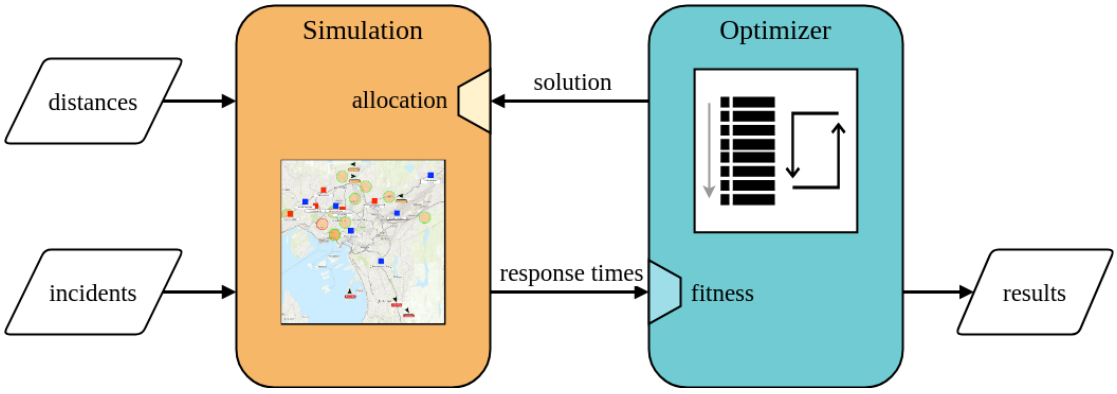


Figure 5.1: Overview of the main parts of the optimization model.

5.1.3 Heuristic search

Weise [2009, p.22] defines a heuristic as a part of an algorithm that guides its search process. Often, a heuristic search algorithm is thought of as an approxi-

mate or "good enough"-algorithm trading correctness for speed. This thesis will employ three such algorithms for optimization: SLS, GA, and MA. The use of this algorithm family is motivated by the enormous solution space, quantified earlier in section 2.3.

5.2 Stochastic Local Search (SLS)

Stochastic Local Search (SLS) is, as the name suggests, a local search algorithm, which means that it searches the immediate neighborhood of the current genotype for better configurations that improve upon this genotype. This is in contrast to global search algorithms, which do not have the same restriction.

The algorithm used in this project is based on the binary SLS for costly fitness functions (SLS4CFF) presented in Mengshoel and Riege [2022]. The pseudocode of the algorithm is given in algorithm 2. The implementation will be described in section 5.2.1.

Algorithm 2: Stochastic Local Search (SLS)

Input : max time τ_{max}, restart probability p_R, noise probability p_N, number of ambulances day n_{Day}, number of ambulances night n_{Night}

Output: best solution found $\mathbf{X}^* = m(\mathbf{G}^*)$

```
1  function SLS
2      G* ← Restart(n_Day, n_Night)
3      G ← Restart(n_Day, n_Night)
4      while elapsedTime() < τ_max do
5          if random(0,1) < p_R then
6              G ←Restart(n_Day, n_Night)
7          else
8              if random(0,1) < p_N then
9                  G ← Noise(G)
10             else
11                 G ← LocalSearch(G)
12         if F(m(G);θ) > F(m(G*);θ) then
13             G* ← G
14     return m(G*)
```

An important feature that distinguishes this version of SLS from others is the random restarts, as opposed to restarting at a fixed number of tries. The stochastic

parts of the algorithm come from the random restarts and the Noise($\mathbf{G}$) operator, while the LocalSearch($\mathbf{G}$) operator always selects the best genotype from the neighborhood. The implementation of the operators and representations is problem-dependent.

5.2.1 SLS implementation

This version of SLS uses the same encoding $\mathbf{G}$ defined earlier in Equation 5.3 which uses the base station IDs from B as its smallest component. It searches through possible genotypes and keeps track of the best ones found by first mapping to solutions before evaluating using the fitness function defined in Equation 5.1. Since a genotype is represented using integers, the operators must also be defined for integers. First, let the forward step f function from a genotype $\mathbf{g}$, at gene i, be defined by Equation 5.5 which is effectively an increment with loop around at the number of base stations k.[1] For composite genotypes $\mathbf{G}$, a second index j also needs to be specified.

$$\begin{aligned} f_k(\mathbf{g}, i) &= (g_0, g_1, \ldots, g_i + 1 \mod k, \ldots, g_{n-1}) \\ f_k(\mathbf{G}, i, j) &= (\mathbf{g}_0, \mathbf{g}_1, \ldots, f_k(\mathbf{g}_j, i), \ldots, \mathbf{g}_{m-1}) \end{aligned} \tag{5.5}$$

When the forward step function is applied to all genes in a given $\mathbf{g}$, it forms a tree-like structure with $\mathbf{g}$ as the root and n unique leaf nodes. This is visualized in Figure 5.2.

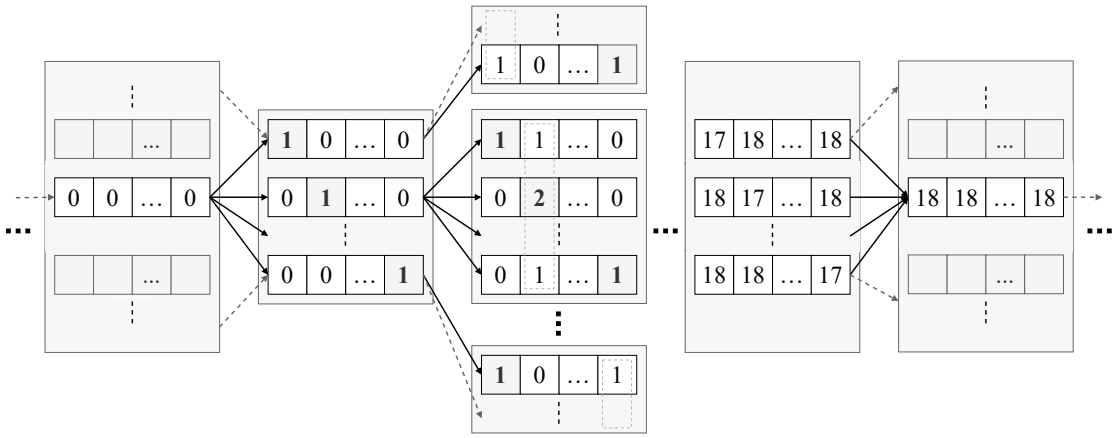


Figure 5.2: The SLS forward step function visualized.

[1] In this thesis, k will always be set to 19, but is left as parameter for the possibility of changing the number of base stations in the future.

Based on the forward step, the forward neighborhood FN of a genotype can be defined by Equation 5.6 as the set of all genotypes generated by applying the function on all genes.

$$\begin{aligned} FN_k(\mathbf{g}) &= \{\, f_k(\mathbf{g}, i) \mid i \in [0..n-1] \,\} \\ FN_k(\mathbf{G}) &= \{\, f_k(\mathbf{G}, i, j) \mid i \in [0..n-1], j \in [0..m-1] \,\} \end{aligned} \tag{5.6}$$

Restart sets all genes in all allocations in $\mathbf{G}$ to a random gene from the set of base station IDs B. The length of each genotype is determined by the number of ambulances during the day shift n_{Day} and night shift n_{Night}, respectively. This is used as both the initialization function and as the Restart(n_{Day}, n_{Night}) operator, which is run based on the restart probability p_R.

Noise selects a random allocation index j from $\mathbf{G}$ and a random gene index i in the allocation to apply forward step on.

LocalSearch evaluates all neighbors in the forward neighborhood of the current composite genotype $\mathbf{G}$ and selects the one with the best fitness greedily.

As a side note, the implementation described above is that the algorithm implicitly exerts a form of tabu search by only going in one direction (forward). Tabu search is another search algorithm in which visited solutions are placed in a forbidden, or *tabu*, list to escape local minima. The implicit tabu list in this version of the SLS is emptied at each reset because previous genotypes are allowed to be visited again by going forward.

5.2.2 Alternative SLS implementation

Some readers might find the neighborhood definition in the above description a bit restricted, as this treats the variables as cardinal numbers and searches through them in increasing order. The reason behind this is simply to limit the branching factor of the greedy search because the fitness function is so costly. For comparison purposes, another alternative neighborhood with another, more general, neighborhood function is also implemented.

The alternative SLS uses hamming distance to determine the neighborhood. The Hamming distance, $H(s, s')$, is defined as the number of positions in which the symbols are different in two strings of equal length. As an example, the Hamming distance between 1212 and 1234 becomes:

$$H(12\mathbf{12}, 12\mathbf{34}) = 2.$$

With this in mind, the Hamming neighborhood of a genotype is defined as the set of all genotypes that have a Hamming distance of 1 from the current genotype. A formal definition is given by Equation 5.7.

$$\begin{aligned} HN(\mathbf{g}) &= \{\, \mathbf{g}' \mid H(\mathbf{g}', \mathbf{g}) = 1 \,\} \\ HN(\mathbf{G}) &= \{\, \mathbf{G}' \mid H(\mathbf{g}', \mathbf{g}) = 1 \,, \mathbf{g} \in \mathbf{G} \} \end{aligned} \tag{5.7}$$

We define the first algorithm as Forward SLS (FSLS), and the alternative version, using the Hamming distance, as Hamming SLS (HSLS). To find which performs the best, a quick experiment is performed in which both algorithms are run using the exact same parameters. The results of this experiment are summarized in Table 5.2.

		Average response time (Fitness)		
Algorithm	**Rank**	**Best**	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
FSLS	1	931.82 s	944.41 s	6.88 s
HSLS	2	950.03 s	964.32 s	10.23 s

Table 5.2: FSLS and HSLS comparison test results over 15 runs.

In this test, each algorithm was given 15 independent runs with a maximum time τ_{max} of 10 minutes. The default simulation configuration parameters $\boldsymbol{\theta}$ from Table 4.1 were used. Restart probability p_R was set to 0.025, and noise probability p_N to 0.8.

Looking at the test results, it is clear that FSLS performed the best overall, getting a mean average response time of 944.41 s, beating the mean average response time of HSLS by 19.91 s. FSLS also had a bit lower standard deviation than HSLS. A box plot visualizing this difference is given by Figure 5.3.

Based on these results, only FSLS will be used in the following sections and will be referred to as simply SLS from now on.

5.3 Genetic algorithms

A Genetic Algorithm (GA) is often used when faced with an optimization problem where the search space is too large for classical computers to solve by exhaustive search. In contrast to SLS, it is a global search algorithm, which means that it does not use the notion of a neighborhood anywhere. GAs are classified as metaheuristic algorithms because they are general problem-solving algorithms that can solve different types of problems.

GAs are a type of evolutionary algorithm, inspired by Darwinian evolution, which mimics the principle of "survival of the fittest" to find the best solution. The

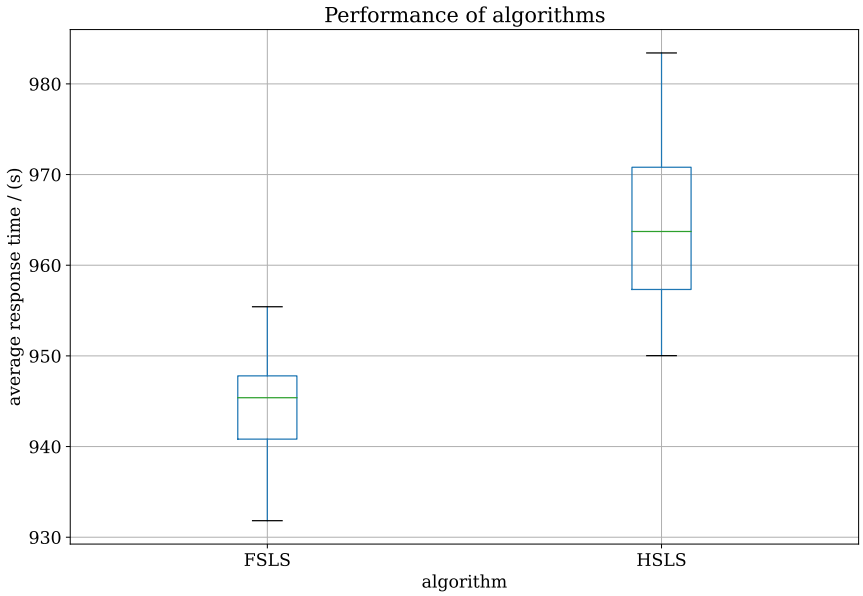


Figure 5.3: Box plot showing the distribution of average response times for the Forward SLS (FSLS) and the Hamming SLS (HSLS) over the course of 15 runs.

language used in the GA literature borrows many terms from biology to highlight the similarities between linked concepts. In GA terms, the tuple of a genotype and its corresponding decoded phenotype $(\mathbf{G}, \mathbf{X} = m(\mathbf{G}))$ is called an individual, and a collection of individuals is called a population. The GA has operators that work on an individual level and others that work on the population level.

The pseudocode for the generational GA, adapted from Eiben and Smith [2003], is given by algorithm 3. A general flow chart for the GA is given by Figure 5.4, which illustrates the main processes. The implementation used in this thesis is a generational GA, which means that each cycle the population is replaced by a new one. When the GA terminates after τ_{max} seconds, the best solution found, $\mathbf{X}^* = m(\mathbf{G}^*)$, is returned by the algorithm.

5.3.1 Population initialization

The first step of a genetic algorithm, as illustrated in the flowchart in Figure 5.4, is the Initialize(s_{pop}, n_{Day}, n_{Night}) procedure where s_{pop} individuals are randomly generated to ensure a diverse population. The length of each genotype in the composite genotype contained in each individual is determined by the parameters for the number of ambulances during the day n_{Day} and the number of ambulances during the night n_{Night}.

Algorithm 3: Genetic Algorithm (GA)

Input : max time τ_{max}, population size s_{pop}, elite size s_{elite}, tournament size s_{tourn}, crossover probability p_C, mutation probability p_M, number of ambulances day n_{Day}, number of ambulances night n_{Night}

Output: best solution found $\mathbf{X}^* = m(\mathbf{G}^*)$

```
1  function GA
2      P ← Initialize(s_pop, n_Day, n_Night)
3      E ← EvaluateFitness(P, F, m)
4      while elapsedTime() < τ_max do
           // Save best fitness, average fitness, entropy:
5          saveStatistics(P, E)
6          P_next ← Elite(P, E, s_elite)
7          while |P_next| < s_pop do
8              G_a, G_b ← ParentSelection(P, E, s_tourn)
9              G_a, G_b ← Crossover(G_a, G_b, p_C)
10             G_a, G_b ← Mutate(G_a, p_M), Mutate(G_b, p_M)
11             P_next ← P_next ∪ { G_a, G_b }
12         P ← P_next        // Replace the old population with the new
13         E ← EvaluateFitness(P, F, m)
14     return m(Elite(P, E, 1))
```

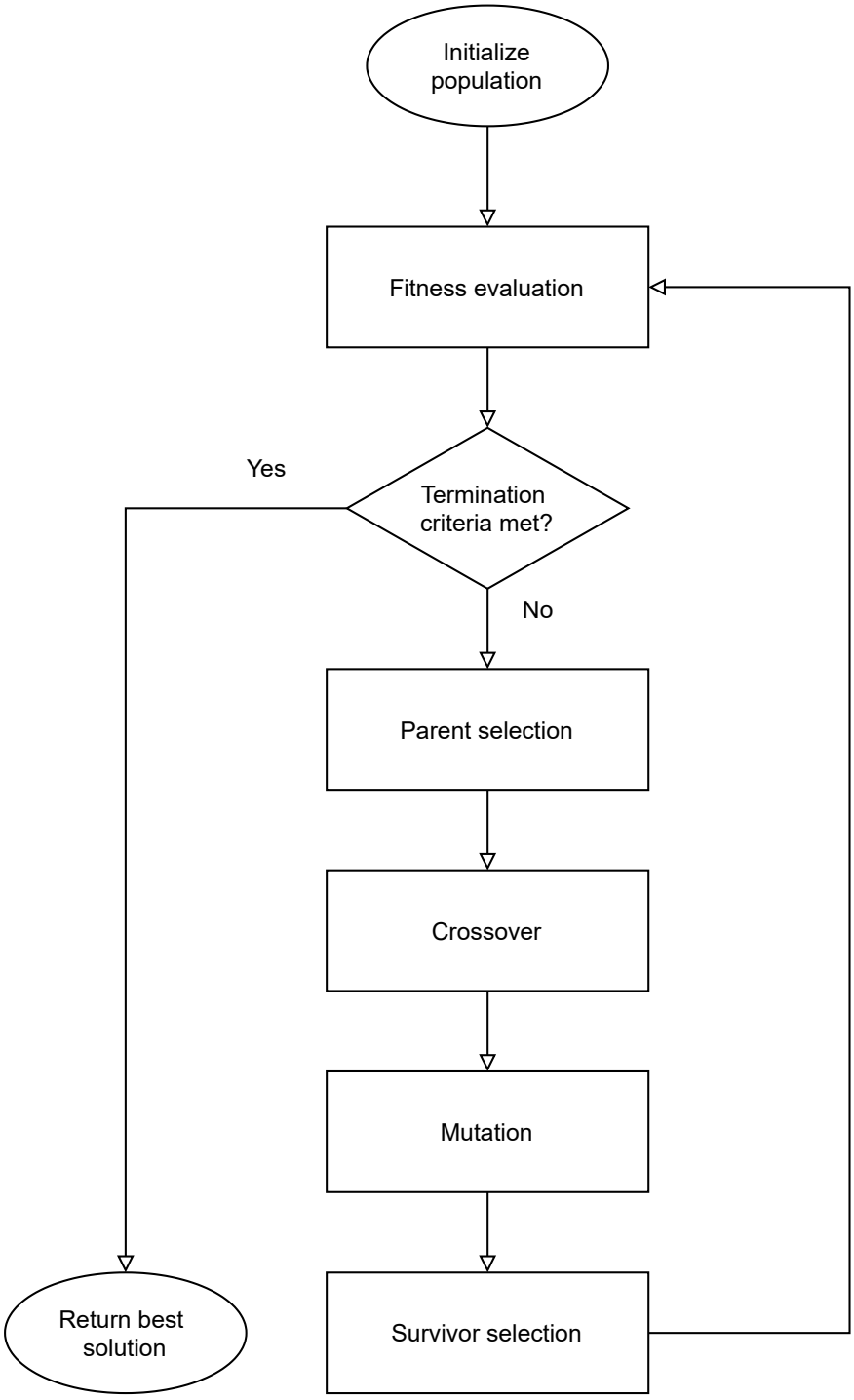


Figure 5.4: A flowchart demonstrating the life cycle of a generic genetic algorithm.

After initialization, each individual is given a fitness through simulation in the fitness function F. This is done in parallel to increase performance. The algorithm then enters what is referred to as the generation loop, where the next population is generated based on the current population. The first individuals in the next population, which are passed on directly, are the s_{elite} best individuals from the previous population selected by the E, s_{elite} procedure. The s_{elite} individuals with the highest fitness are called elite.

5.3.2 Genetic operators

In the generational loop of the GA implementation, lines 7–11, *genetic operators*, which work at the individual level, are applied to explore new genotypes and exploit existing ones to guide the algorithm towards finding an optimum solution. The loop begins with the ParentSelection(P, E, s_{tourn}) procedure where the fitness evaluations E are used to select two individuals, $\mathbf{G}_a$ and $\mathbf{G}_b$, based on a concept known as tournament selection. This is done by selecting s_{tourn} individuals from the population at random. The function returns the two individuals with the highest fitness values. These two individuals, sometimes called parents, then undergo *crossover* with some probability p_C. If crossover is not applied, the parents still move on to mutation, which is the next step. During the Crossover($\mathbf{G}, \mathbf{G}$, p_C) procedure, the parents' genes are combined into a new pair of individuals, sometimes called offspring, using the one-point crossover method, illustrated with two example genotypes in Figure 5.5 below. For composite genotypes, the crossover is applied pairwise, sequentially through each of the genotypes in the two parents.

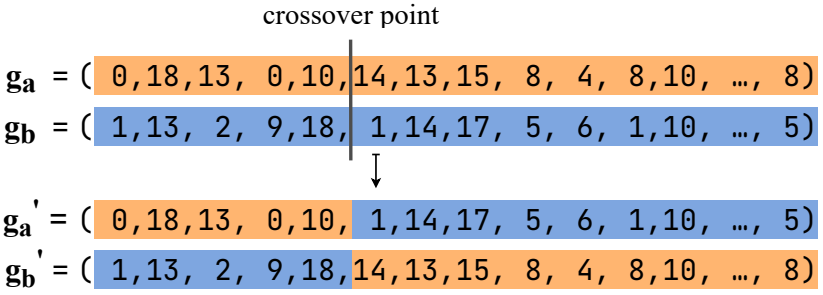


Figure 5.5: One-point crossover illustration. The parents (top) first gets a random crossover point selected, which splits each genotype in two. The first part of parent $\mathbf{g}_a$ is then copied to the first part of offspring $\mathbf{g}'_a$, and the last part to the last part of $\mathbf{g}'_b$. The remaining parts are filled with parent $\mathbf{g}_b$ in the same way.

After crossover, the Mutate($\mathbf{G}_a$, p_M) operator is applied, which changes genes in the genotype at random with a probability p_M per gene. For composite genotypes,

this procedure is simply repeated at each genotype. This introduces random variation in individuals, which pushes the GA to explore new genotypes. An example of the effect of applying the Mutate($\mathbf{G}_a$, p_M) operator is illustrated in Figure 5.6 below.

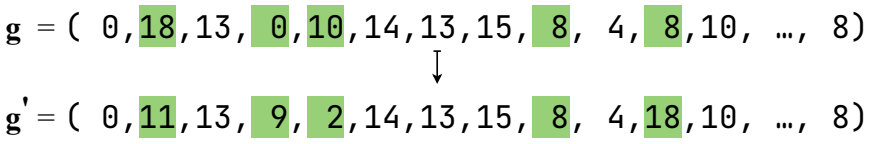


Figure 5.6: Mutate operator illustration. An example of a random mutation operator applied to a genotype g that randomly changes genes (marked in green). Notice the number 8 which by chance got changed back to another number 8, this is also allowed.

5.3.3 Diversity

The concept of how different the individuals in the population are is called diversity. The tendency to select individuals with the best fitness, combined with the fact that individuals can be copied directly to the next generation, means that the diversity of the population will tend to decrease over time as the algorithm optimizes. Generally, it is preferable that the diversity is not too low and yet not too high either.

To quantify the diversity in this implementation, the Shannon entropy formula will be used for the frequency of occurrence of the genes. The occurrence frequency, of, for a gene, b, is found by counting all occurrences of that gene in all genotypes of the population and dividing this number by the size of the population multiplied by the length of the genotype. The Shannon entropy H for the occurrence frequency for a population p is defined in Equation 5.8.

$$H(\mathbf{p}) := -\sum_{b \in B} of(b, \mathbf{p}) \log_{|B|} of(b, \mathbf{p}) \tag{5.8}$$

If the occurrence frequency for a gene is zero, then that term is defined as zero. For composite allocations, H is taken to be the normalized sum of the Shannon entropy for each allocation type in the population. The Shannon entropy is a number between 0 and 1, where 0 means that all individuals are equal, and 1 means that they are all different.

In line 5 of algorithm 3, the best fitness, the average fitness, and the entropy are saved for each generation. These are useful high-level metrics to monitor the GA search process, which will be used later.

5.4 Memetic algorithms

A Memetic Algorithm (MA), also called a hybrid genetic algorithm, is in essence an extension of the standard GA, at least the version used in this project. The MA differs from the GA in that it includes a local search in a step known as the *improve procedure* [Neri & Cotta, 2012]. algorithm 4 shows the pseudocode for the MA, which includes such an Improve($\mathbf{G}$, p_I) procedure on line 11 with some probability p_I. Other than that, it is exactly the same as the GA mentioned in the preceding section.

Algorithm 4: Memetic Algorithm (MA)

Input : max time τ_{max}, population size s_{pop}, elite size s_{elite}, tournament size s_{tourn}, crossover probability p_C, mutation probability p_M, improve probability p_I, number of ambulances day n_{Day}, number of ambulances night n_{Night}

Output: best solution found $\mathbf{X}^* = m(\mathbf{G}^*)$

```
1  function MA
2      P ← Initialize(s_pop, n_Day, n_Night);
3      E ← EvaluateFitness(P, F, m);
4      while elapsedTime() < τ_max do
           // Save best fitness, average fitness, entropy:
5          saveStatistics(P, E);
6          P_next ← Elite(P, E, s_elite);
7          while |P_next| < s_pop do
8              G_a, G_b ← ParentSelection(P, E, s_tourn);
9              G_a, G_b ← Crossover(G_a, G_b, p_C);
10             G_a, G_b ← Mutate(G_a, p_M), Mutate(G_b, p_M);
11             G_a, G_b ← Improve(G_a, p_I), Improve(G_b, p_I) ;      // NB! New
12             P_next ← P_next ∪ { G_a, G_b };
13         P ← P_next ;      // Replace the old population with the new
14         E ← EvaluateFitness(P, F, m);
15     return m(Elite(P, E, 1))
```

The implementation of the Improve($\mathbf{G}$, p_I) procedure is the same as the one used in the SLS algorithm described in section 5.2.1. Unlike SLS, the improve procedure of an MA will only replace the original individual if the newly found individual is actually an improvement in terms of fitness, which the name of the procedure would suggest. If the new individual is worse, the old individual is kept.

Some of the motivation behind testing hybridization with an MA is that it can help reduce premature convergence and better explore the solution space [Garg,

2009]. In other cases, it can help improve the performance of the search by converging to a feasible optimum faster than a pure GA approach [Behmanesh & Pannek, 2021].

Although the MA is an extension of the GA, one should not automatically assume that this means that an MA will always perform better. One explanation for this can be found in the No Free Lunch (NFL) theorem in optimization problems, popularized by Wolpert and Macready [1997]. A consequence of this theorem, assuming it holds, is that the computational performance of any optimization algorithm is the same when it is averaged for all problems within a specific class. This is supported by Garg [2009] which compared the performance of a GA and an MA on the same problem and found that the best-performing algorithm varied with both the size of the search space and the required precision of the results. Because of this implication, it would be sensible to evaluate both a standard GA and an MA with an appropriate local search operator against each other with different parameters for an optimization approach attempting to utilize the benefits of the local search operators.

Chapter 6

Experiments and Results

In this chapter, each experiment is presented with its associated objective, design, results, and discussion of the results. There are four experiments in total. The first two experiments use different techniques to find good allocations based on the simulation. The latter two focuses on testing the flexibility of the simulation itself by keeping the allocations constant and varying other parameters.

All experiments were run on a laptop using Java 17, as mentioned in section 2.4. The experiments were performed on a MacBook Pro with M1 Pro chipset (10-core CPU) with 32 GB RAM. For reproducibility, the pseudo-random number generator received a fixed seed with a value of 10062022.

Each experiment was given a separate script and run independently. The results of each experiment were saved to comma-separated value (CSV) files for analysis and visualization. A centralized Python-based script was responsible for aggregating the results and producing the figures. The specific packages used are described in more detail in section 2.4.

The simulation model presented in chapter 4 is used in all experiments. Experiments 1, 2, and 4 use the default configuration parameters $\boldsymbol{\theta}$ from Table 4.1. Experiment 3 changes the start and end date of the simulation period, keeping the rest of the parameters constant.

6.1 Experiment 1: Simple allocation methods

6.1.1 Objective

The objective of experiment 1 is to test some simple allocation strategies and find which of them performs the best. This will later serve as a baseline for the more sophisticated methods detailed in chapter 5.

6.1.2 Design

Five allocation strategies are used: random allocation, all city center allocation, uniform allocation, uniform with random allocation, and population proportionate allocation. What they all have in common is that they are memoryless and do not make use of the data set in any way to generate allocations.

The general setup of the experiment is simple: Generate an allocation from each strategy, execute the simulation with default parameters using the generated allocation, and save the resulting response times. The response times can then be analyzed and aggregated to compare each method. For stochastic strategies that do not produce the same allocations persistently (detailed below), 15 runs are used, with the allocation producing the median average response time used for comparison against the other methods.

Random allocation is the simplest allocation strategy in the list and would not be realistic to use in a real operational context—at least not directly. As the name suggests, this is a stochastic strategy that results in a different allocation each time it is run.

All city center allocation was included as a reference technique to see how one extreme would perform. This technique places all ambulances in the city center at base station 7, which means that there is no geographical spread. The other extreme is to place the ambulances as evenly as possible, which is what **Uniform allocation** does.

Uniform allocation attempts to assign the same number of ambulances to each base station. If the number of ambulances is not a multiple of the number of base stations, the remaining r ambulances are distributed over the first r base stations in the order presented in Table 2.2. This clearly has a bias for selecting the first base stations, which is why **Uniform with random allocation** was also included to "smooth out" the residual ambulances evenly.

Uniform with random allocation eliminates the bias towards allocating additional ambulances to the first base stations in Table 2.2 by introducing stochasticity, although with less variation between each run than **Random allocation**. The procedure is as follows: continue adding one ambulance to all base stations until doing so would exceed the number of ambulances available. The remaining r ambulances (if any) are then randomly assigned until there are no ambulances left to assign.

Population proportionate allocation uses k-means clustering to estimate the population served by each base station and distributes ambulances according to the proportion of the total population that each base station services. The result of the clustering can be seen in Figure 6.1. The allocation was obtained by multiplying the population proportion for each base station with the number of ambulances to be distributed rounded to the nearest whole number. Due to the need for rounding, there is a chance that the number of allocated ambulances is not equal to the actual number of ambulances available. To fix this issue, ambulances were either removed or added from the allocation, ranked according to the rounding error. In the case where more ambulances were allocated than available, the base stations that received additional ambulances because the number was rounded up were removed and vice versa for the case with fewer ambulances than available.

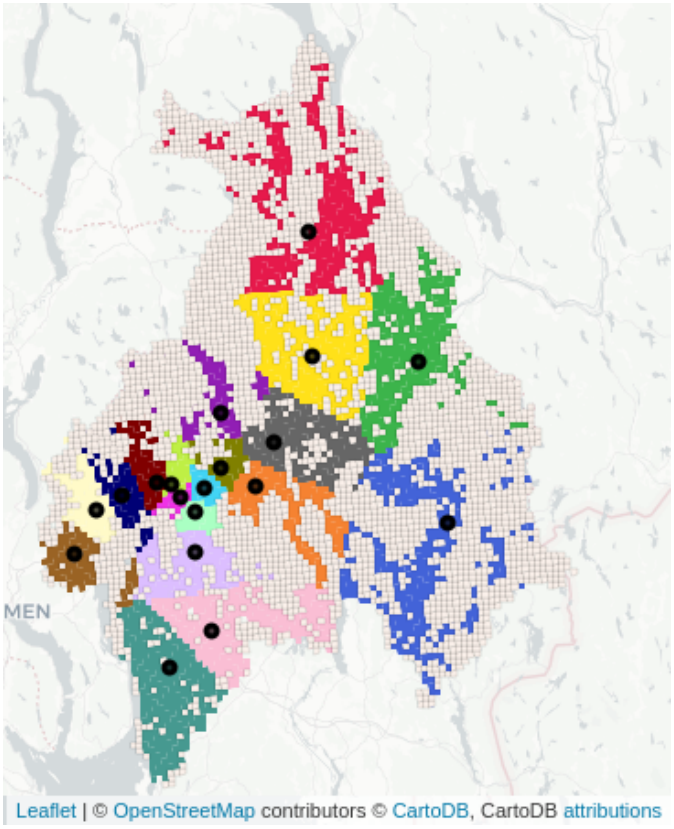


Figure 6.1: Base station clusters based on $1 \times 1\,km$ Statistics Norway grids with population statistics from 2018. This is what the **Population proportionate allocation** uses to allocate ambulances. Grid cells with a population of zero are colored beige.

6.1.3 Results and discussion

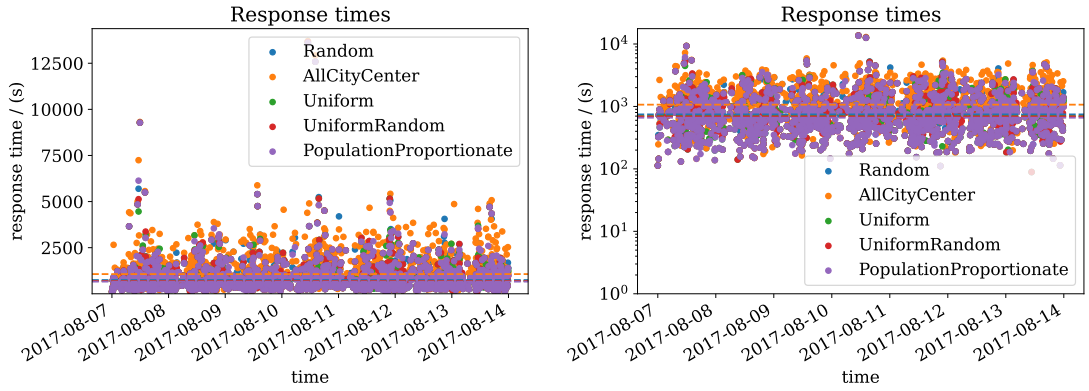
Table 6.1 shows the allocations produced by each simple strategy. Table 6.2 shows the rank of each method, in addition to the median, mean, and standard deviation of the average response times for the 15 runs. The average response time for the best allocation generated is also included. The rank is based on the median run.

Figure 6.2 plots the various response times in chronological order, while Figure 6.3a shows the response time distributions, sorted from the lowest response time to the highest. This shows that most missions had a response time below $2000\,s$. Beside it, Figure 6.3b shows the same plot zoomed in on the interquartile range on a logarithmic scale to more easily differentiate the different methods.

It is clear that the AllCityCenter allocation strategy underperforms compared to the other strategies; this makes sense, since ambulances may have to travel a long distance in some cases to respond to incidents on the outskirts of the operation area. It is also interesting to note that the PopulationProportionate strategy performs the best and that UniformRandom is not far behind. This would suggest that in terms of ambulance allocation, the performance of policies that use coverage and population to decide the allocation is not that different when the average response time is considered.

Strategy	**Shift**	**Allocations ($\mathbf{X}$)**
Random	Day	$\mathbf{x}_0 = (1,1,6,3,2,2,3,3,2,3,5,4,1,2,0,1,2,1,3)$
	Night	$\mathbf{x}_1 = (1,1,0,0,0,2,1,4,3,1,2,2,2,4,3,0,0,0,3)$
AllCityCenter	Day	$\mathbf{x}_0 = (0,0,0,0,0,0,0,45,0,0,0,0,0,0,0,0,0,0,0)$
	Night	$\mathbf{x}_1 = (0,0,0,0,0,0,0,29,0,0,0,0,0,0,0,0,0,0,0)$
Uniform	Day	$\mathbf{x}_0 = (3,3,3,3,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2)$
	Night	$\mathbf{x}_1 = (2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1)$
UniformRandom	Day	$\mathbf{x}_0 = (2,2,3,2,3,3,2,3,3,3,2,2,2,3,2,2,2,2,2)$
	Night	$\mathbf{x}_1 = (2,1,2,2,1,1,2,1,2,1,1,2,1,2,2,2,1,1,2)$
Population-Proportionate	Day	$\mathbf{x}_0 = (1,1,1,1,3,1,3,5,4,2,2,3,2,2,3,3,3,2,3)$
	Night	$\mathbf{x}_1 = (1,1,1,0,2,0,2,3,3,1,1,2,1,2,2,2,2,1,2)$

Table 6.1: The resulting allocations from experiment 1. Deterministic methods (AllCityCenter, Uniform, and PopulationProportionate) were run once, while the stochastic methods (Random, UniformRandom) were run 15 times. For the latter, the allocation that produced the median average response time is the one shown.



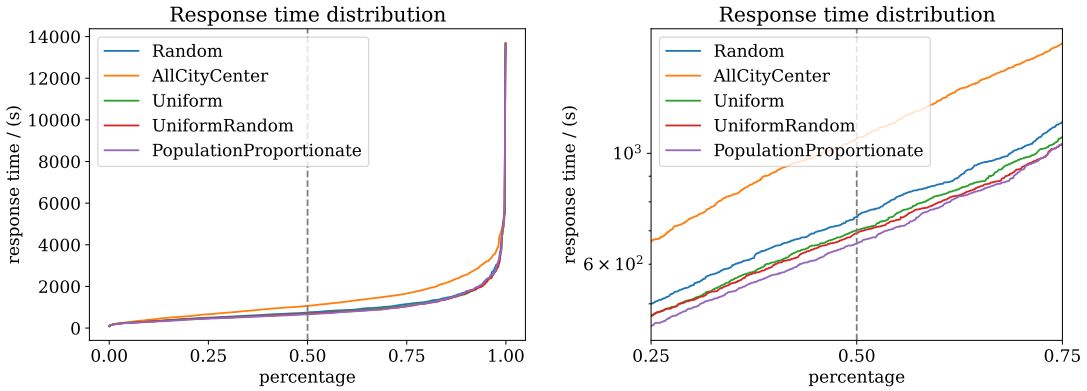
(a) Response times.

(b) Response times using a logarithmic scale.

Figure 6.2: Experiment 1 response times. The results from the non-deterministic methods (Random and UniformRandom) are taken from the run yielding the median average response times.

		Average response time (Fitness)			
Strategy	**Rank**	**Best**	**Median**	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
Random	4	911.06 s	959.38 s	955.29 s	28.25 s
AllCityCenter	5	1325.98 s	1325.98 s	1325.98 s	–
Uniform	3	913.8 s	913.8 s	913.8 s	–
UniformRandom	2	886.55 s	901.47 s	900.73 s	8.24 s
PopulationProportionate	1	897.63 s	897.63 s	897.63 s	–

Table 6.2: Experiment 1 average response time statistics for the 15 runs.



(a) Response time distribution.

(b) Response time distribution interquartile range using a logarithmic scale.

Figure 6.3: Experiment 1 response time distribution from the allocation produced by each method. The results from the non-deterministic methods (Random and UniformRandom) are taken from the run with the median average response time. The median is indicated by the dashed line.

6.2 Experiment 2: Advanced allocation methods

6.2.1 Objective

In experiment two, the various optimization approaches presented in chapter 5 are tested with the simulation in order to compare the viability of these against each other.

6.2.2 Design

Similar to experiment 1, this experiment will use different methods to obtain allocations, although in this case a variety of allocations is tested against each other by running each allocation through the simulation once and iterating using the metaheuristic algorithms. Each method is allowed to evaluate allocations at will and use this information to search for allocations that yield high average response times for a given time. Since all algorithms used in this experiment use stochasticity, each is given 15 runs to find the most optimal solution, each run having a soft maximum time τ_{max} of 4 minutes. A *soft* maximum because the algorithm is allowed to complete its current task between every time τ_{max} is checked, which is done in the outermost loop in the pseudocodes in algorithm 2, algorithm 3, and algorithm 4. This introduces a slight bias towards the algorithms with the longest running times (SLS and MA in this case) because they are more likely to exceed τ_{max}.

The allocation that produces the best overall average response time will then be saved and used to compare with the other methods. Using metaheuristics requires some time to be spent testing different configurations of hyperparameters to perform the best possible optimizers. The hyperparameters used by the SLS optimizer are shown in Table 6.3, and the hyperparameters used by the GA and the MA are shown in Table 6.4.

Symbol	**Parameter**	**SLS**
τ_{max}	Max time	4 min
p_R	Restart probability	20%
p_N	Noise probability	5%
n_{Day}	Number of ambulances day	45
n_{Night}	Number of ambulances night	29

Table 6.3: Parameters used by SLS in experiment 2.

Parameter	Name	GA	MA
τ_{max}	Max time	4 min	4 min
s_{pop}	Population size	30	30
s_{elite}	Elite size	4	4
s_{tourn}	Tournament size	5	5
p_C	Crossover probability	20%	20%
p_M	Mutation probability	5%	5%
p_I	Improve probability	–	25%
n_{Day}	Number of ambulances day	45	45
n_{Night}	Number of ambulances night	29	29

Table 6.4: Parameters used by GA and MA in experiment 2.

6.2.3 Results and discussion

The best allocation produced by each of the three algorithms is shown in Table 6.5. Table 6.6 shows the rank of each algorithm, as well as the median, mean, and standard deviation of the average response times collected over 15 runs. The average response time for the overall best allocation produced is also included. This ranks the GA first, MA second, and SLS last based on the best fitness obtained.

The plots showing the response times produced by the best allocation of each algorithm are shown in Figure 6.4 with and without a logarithmic scale. The response time distributions is displayed in Figure 6.5a sorted from the lowest response time to the highest, and Figure 6.5b shows the same plot zoomed in on the interquartile range on a logarithmic scale.

Figure 6.6 visualizes the results as box plots. This demonstrates that SLS is outperformed by both the GA and MA. The MA performance falls between that of SLS and GA. Considering that the MA is a hybrid between the GA and the SLS, this might not come as a surprise. Other local search operators could be explored for the improvement procedure to see if this could improve the performance of the MA. However, it could also be that this could be a case of the No Free Lunch (NFL) theorem in action, as discussed earlier in section 5.4, and that in this specific case a simpler GA might be more appropriate.

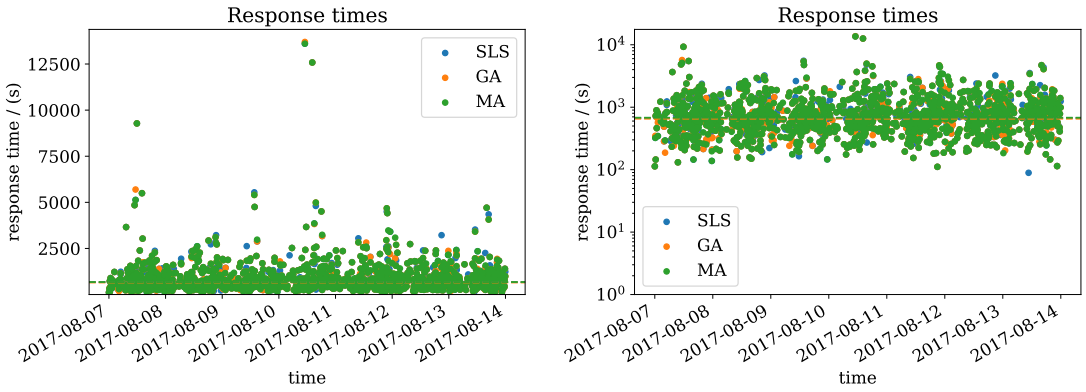
Figure 6.7 shows the SLS search progress of the run that produces the best solution. Because the SLS either does a noise step or a greedy step, the fitness value of the current genotype (the blue line) will tend to jump up and down. In Figure 6.8 the search progresses from the GA and the MA from the run that produced the best solution is illustrated. This also highlights performance differences between the GA and the MA due to the fact that the MA manages to complete fewer generations in the allotted time of τ_{max} than the GA.

Strategy	Shift	Allocations (X)
SLS	Day	$\mathbf{x}_0 = (2, 1, 1, 2, 2, 3, 3, 5, 3, 4, 3, 2, 1, 2, 2, 2, 2, 4, 1)$
	Night	$\mathbf{x}_1 = (1, 2, 2, 1, 3, 0, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 0)$
GA	Day	$\mathbf{x}_0 = (2, 1, 3, 2, 3, 1, 4, 4, 3, 2, 3, 3, 2, 2, 1, 3, 3, 2, 1)$
	Night	$\mathbf{x}_1 = (1, 2, 1, 1, 2, 1, 2, 4, 3, 1, 2, 2, 1, 1, 1, 1, 0, 2, 1)$
MA	Day	$\mathbf{x}_0 = (2, 1, 1, 3, 2, 1, 3, 2, 3, 4, 5, 3, 2, 0, 2, 3, 2, 4, 2)$
	Night	$\mathbf{x}_1 = (1, 2, 1, 2, 1, 1, 2, 5, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1)$

Table 6.5: The best allocation produced by each algorithm from experiment 2 over 15 runs.

		Average response time (Fitness)			
Algorithm	**Rank**	**Best**	**Median**	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
SLS	3	885.85 s	900.45 s	899.95 s	5.21 s
GA	1	863.58 s	867.02 s	867.06 s	1.71 s
MA	2	875.09 s	882.86 s	882.91 s	3.46 s

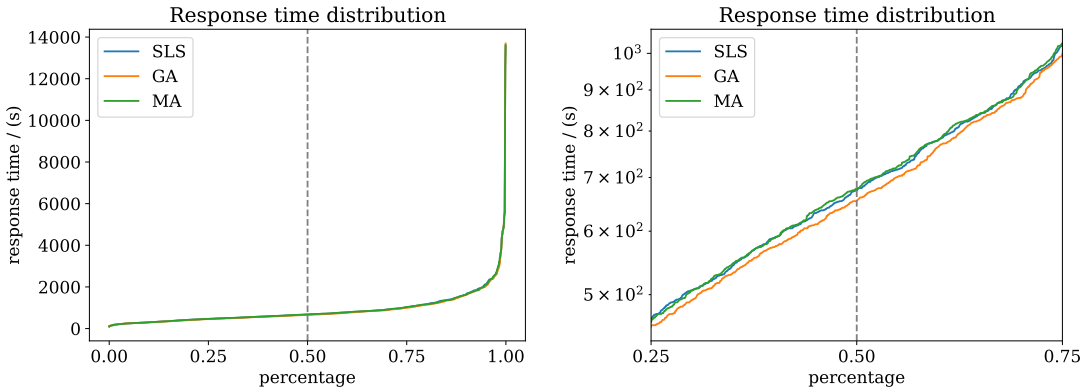
Table 6.6: Experiment 2 fitness statistics for the 15 runs.



(a) Response times.

(b) Response times using a logarithmic scale.

Figure 6.4: Experiment 2 response times taken from the best allocation from each algorithm over 15 runs.



(a) Response time distribution.

(b) Response time distribution interquartile range using a logarithmic scale.

Figure 6.5: Experiment 2 response time distribution from the best allocation produced by each method.

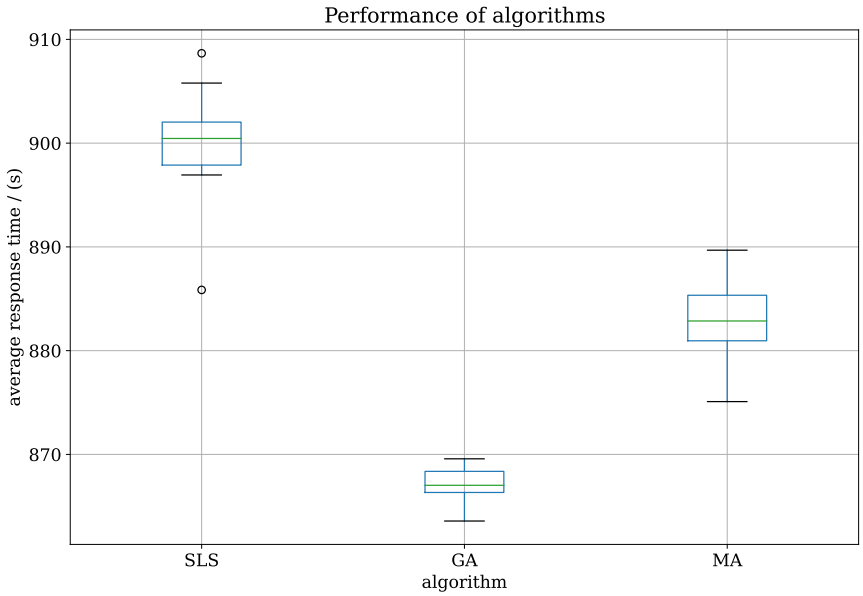


Figure 6.6: Box plot showing the distribution of average response times for each of the algorithm used over 15 runs.

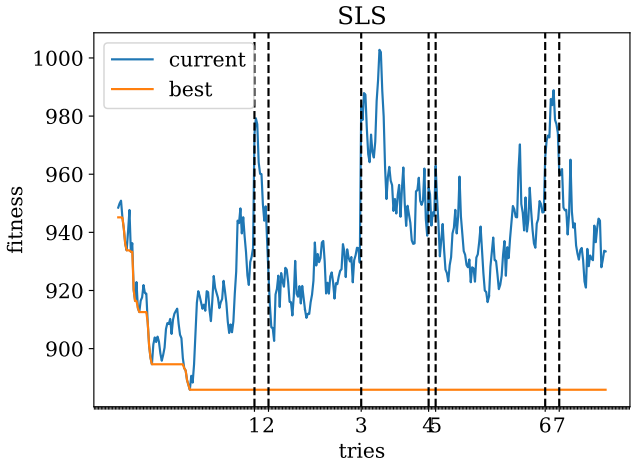
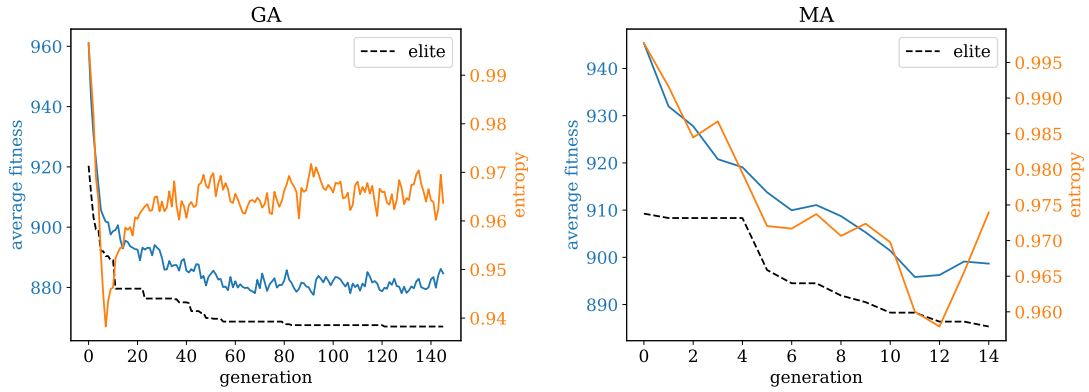


Figure 6.7: SLS search progress from the run that produces the best solution. The random restarts are shown by the black, dashed lines.



(a) GA search progress.

(b) MA search progress.

Figure 6.8: GA and MA search progresses from the run that produced the best solution. The average fitness (the blue line) and the diversity (the orange line) of the population is plotted at each generation, as well as the best individual, elite, which is monotonically decreasing (the black dotted line).

6.3 Experiment 3: Varying simulated time period

6.3.1 Objective

In the third experiment, we aim to validate a selection of pre-computed allocations created by the various allocation strategies and optimization models proposed in experiments 1 and 2 against a variety of time intervals. The rationale behind this is to observe how efficient the various simple allocation strategies are at different time intervals, in addition to testing the generalization capabilities of the optimization algorithms.

6.3.2 Design

The experiment runs through a total of five simulations (for each time period) per allocation. The time frames used for the purpose of this experiment are listed in Table 6.7. The allocations used for this purpose are the best allocations found for each model in experiments 1 and 2. Furthermore, for comparison, a new allocation is introduced in this experiment, generated by a GA that has been optimized with simulations using all incidents that occurred in 2017, which will be referred to as "GA (1 year)" in order to see what impact the simulation period has on model performance and generalizability.

Time frame	Start Date (inc.)	End Date (exc.)
One week	06.08.2018 00:00:00	13.08.2018 00:00:00
Two weeks	06.08.2018 00:00:00	20.08.2018 00:00:00
One month	01.08.2018 00:00:00	01.09.2018 00:00:00
Three months	01.07.2018 00:00:00	01.10.2018 00:00:00
One year	01.01.2018 00:00:00	01.01.2019 00:00:00

Table 6.7: Parameters used by the simulation while running the different time frames in experiment 3. The rest of the parameters were the same as in Table 4.1.

6.3.3 Results and discussion

From Figure 6.9, it can be clearly observed that the AllCityCenter allocation performs consistently worse than all other methods, regardless of the observed period. Further, it can be observed that the PopulationProportionate strategy, although initially outperformed on lower time frames, wins out when the simulation is run over the course of a year. What is particularly interesting is that this strategy even outperforms the GA trained on simulations running for the entire course of a year. Considering that this GA is trained on data from 2017 and tested on data

from 2018 this would suggest that either the PopulationProportionate strategy is always more optimal in the long run, or that the GA is overfitted for the period tested.

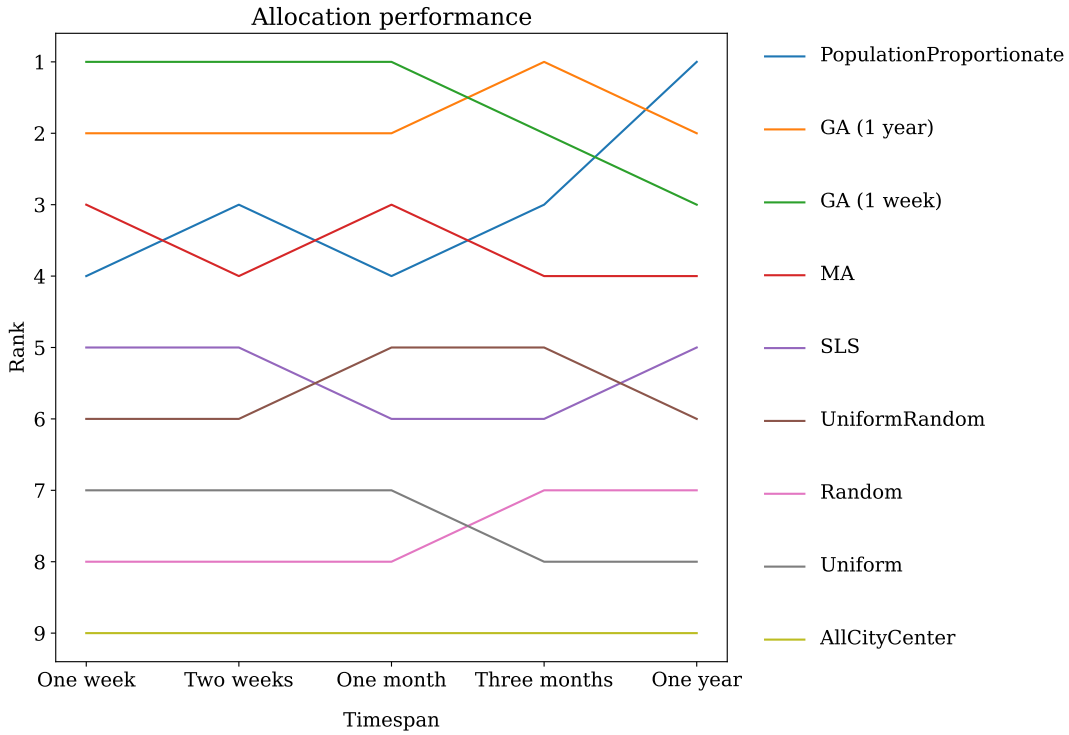


Figure 6.9: Rank of the allocation produced by the methods in experiment 1 and 2, tested with simulations of varying period length. The resulting allocation of GA (1 year) is also included.

6.4 Experiment 4: Varying number of ambulances

6.4.1 Objective

For the final experiment, it would be interesting to observe the general impact that the number of ambulances in operation has on response time and to see whether this impact is different when comparing allocations generated by a simple allocation strategy against allocations generated by an optimizer that optimizes the current ambulance allocation for the varying number of ambulances. For this purpose, several simulations are run with a varying number of ambulances.

6.4.2 Design

This experiment is divided into two sub-experiments. In sub-experiment 1 it is assumed that the number of night-time ambulances is equal to a constant ratio of day ambulances. This ratio is set at 0.64, based on the ratio of the 45 day/29 night split of ambulances reported by OAAD. Sub-experiment 1 is run with a minimum of five ambulances in the day, and increments by one until a final simulation with 70 ambulances during the day is run. These parameters are displayed in Table 6.8.

Parameter	**Name**	**Value**
n_{Day}	Number of ambulances day	[5..70]
n_{Night}	Number of ambulances night	$n_{Day} \cdot 0.64$

Table 6.8: Parameters used in sub-experiment 1 of experiment 4. The rest of the parameters are the same as in Table 4.1.

For sub-experiment 2, all permutations of daytime and nighttime ambulance allocations between 5–70 are tested, as indicated by the cross product of the number of ambulances in Table 6.9. This experiment is only run on the PopulationProportionate allocation method, as the results from sub-experiment 1 showed that the average response times of both the PopulationProportionate and GA follow each other closely, as detailed in the next section. Due to this, as well as the high time complexity of optimizing for 70×70 simulations, this sub-experiment was only run on the PopulationProportionate strategy, which is constant time.

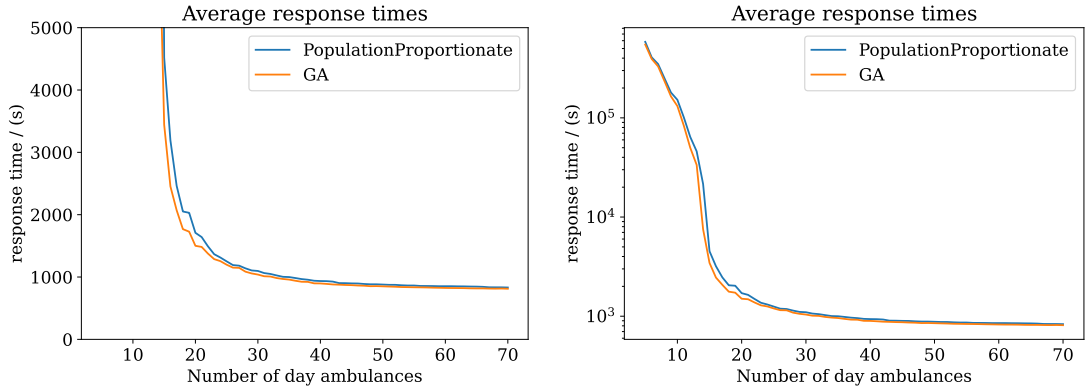
Parameter	**Name**	**Value**
n_{Day}, n_{Night}	Number of ambulances day, night	[5..70] $\times$ [5..70]

Table 6.9: Parameters used in sub-experiment 2 of experiment 4. The rest of the parameters are the same as in Table 4.1.

6.4.3 Results and discussion

For sub-experiment 1 it can be observed in Figure 6.10 that there is an exponential relationship between the average response time and the number of ambulances assigned to the simulation. It is interesting to note that the number of ambulances currently used by OAAD, 45 during the day and 29 during the night, is quite well suited, as any further increase in the number of resources has a very limited impact on response time.

Based on this graph, it can be concluded that the number of units can be reduced somewhat without significantly affecting the performance of the ambulance service. However, this does not consider the non-linear relationship between response time and survivability, as discussed in section 3.2.4. It would be interesting to see what kind of pattern would emerge if one considered an exponential survival function based on response time and other factors such as incident type. This could potentially better capture the time-critical nature of these events. Comparing the performance of the GA and PopulationProportionate suggests that the simple PopulationProportionate strategy is generally quite similar in performance to an optimized approach, suggesting that such an allocation itself is somewhat close to the optimum found by the methods suggested in this thesis.



(a) Average response time per number of ambulances.

(b) Average response time per number of ambulances using a logarithmic scale.

Figure 6.10: Average response time of the GA and PopulationProportionate strategy on a variety of ambulance allocations with a constant nighttime to day-time ambulance ratio of 0.64.

For sub-experiment 2 in Figure 6.11 it can be observed that a lower number of ambulances during the day is worse for overall response time compared to reducing the number of vehicles during the night. However, this is expected considering the

increased number of events during the day compared to night, as can be observed in Figure 2.4.

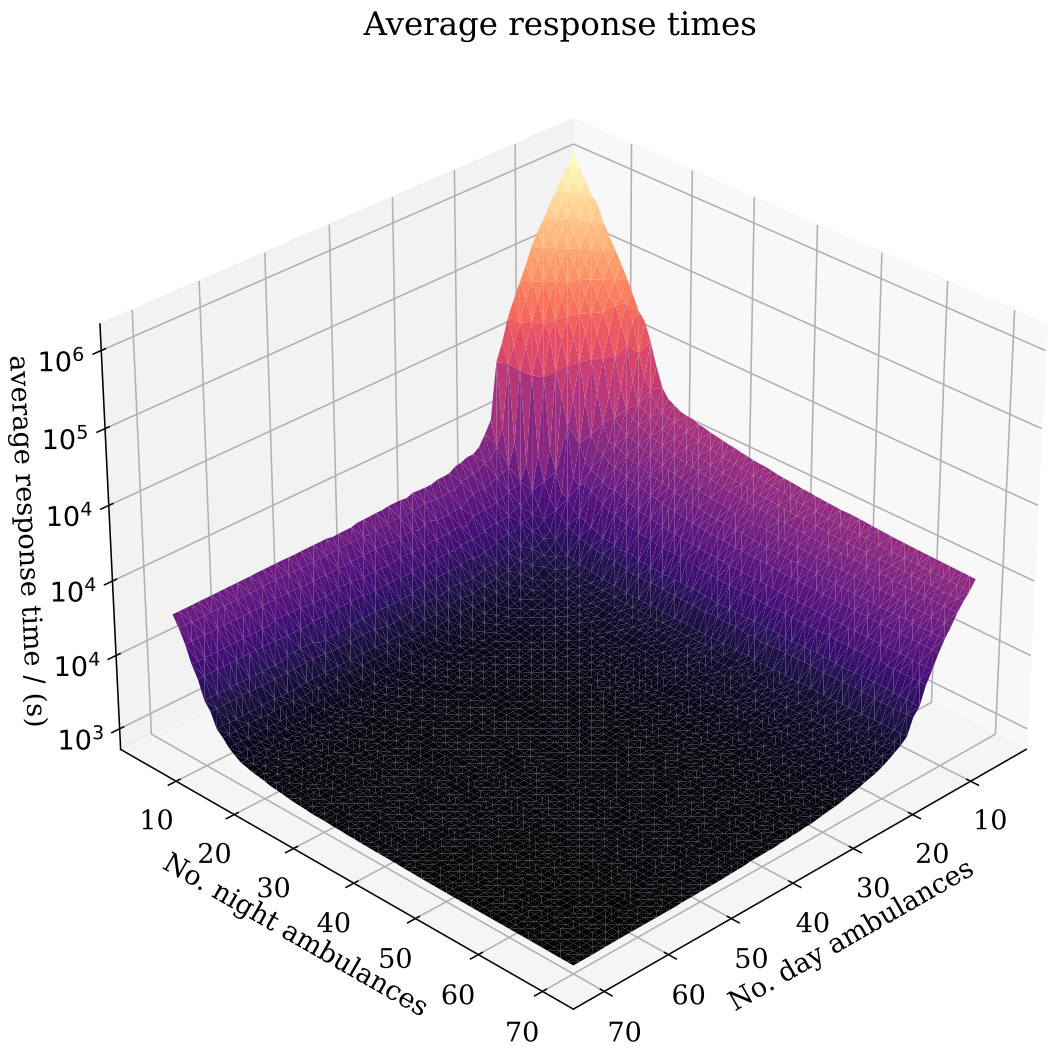


Figure 6.11: Performance of the PopulationProportionate initializer (on a logarithmic scale) on a variety of day-shift and night-shift ambulance allocations.

Chapter 7

Conclusion

With the relevant background information, related work, simulation, optimization models, and experiments presented, this chapter will discuss the key findings and also point out some limitations with this approach and further work.

7.1 Results and discussion

In light of the results of the experiments in chapter 6, the research questions and the overarching goal of chapter 1 are revisited and discussed.

Research Question 1 *What level of realism can be achieved with a simulation based on the EMS incident data set provided by OUH?*

The short answer to this RQ is that a certain level of realism is possible, although the model is of limited use unless more domain knowledge and important factors can be factored in, such as lunch breaks. Certain assumptions have to be made, both due to the inherent complexity of the domain, as well as missing data points, as detailed in section 4.2.2. When comparing raw response times with simulated ones, as shown in section 4.6, the response times have a high correlation. The value also seems to be in the same range; no response times are 0 or negative, for example. Another important point is that the simulation always terminates in a reasonable time (less than one minute). In section 7.4, the authors give suggestions on how the simulation model can be made even more realistic.

Research Question 2 *In what ways can an ambulance system be optimized using the simulation model from RQ1?*

The results of experiments 1 and 2 show that the heuristic search algorithms outperform the simple methods by a good portion, with the GA performing the

best overall. Even more interesting, the plots in Figure 6.7 and Figure 6.8 show the different metaheuristics making steady progress toward finding optimal solutions. They also have a relatively low standard deviation, as reported in Table 6.6, which means that they consistently find good solutions. However, experiments 3 and 4 suggest that the improvements gained through optimization are smaller when the simulation is run with a static allocation over a larger time frame, suggesting that although there is definitive room for optimization, the time frames for which the optimization is performed should be relatively small. It might indicate that there is no general "best" allocation to be found and that instead of finding the specific allocation most optimal for use in longer time periods, the allocations should instead be changed frequently over time by repeatedly optimizing for shorter time periods.

Goal *Maximize EMS patient survivability through ambulance demand forecasting and strategic ambulance allocation in Oslo and Akershus.*

This thesis makes important strides towards this overarching goal. A first-iteration simulation model is implemented, with room for improvement. In addition, the results from the various experiments indicate that a Genetic Algorithm (GA) or other similar model can optimize response times for a specific limited time frame. These models tend to be outperformed by population proportionate allocation when the simulation is run for a longer period of time however. This might indicate either that the models have problems with generalization, or perhaps that ambulance allocations should be considered and optimized for within limited time frames and that the most optical allocation can vary quite a lot between different days, weeks, and months of the year. Although response times are important for survival in many time-critical incidents, it would be even more interesting if we had more ways to apply survival functions for the most common types of time-critical incident. This could perhaps be an even better way to maximize patient survivability, as detailed in section 3.2.4.

7.2 Contributions

The main contributions of this project are summarized in this section.

1. A comprehensive simulation model of the ambulance service in Oslo and Akershus using historical data from real-life events.
    (a) Implementing different ambulance personnel shifts and shift changes into the simulation model.
    (b) Simulating with a comprehensive travel time-model.

   (c) Dynamically dispatching ambulances while returning to the base station.

2. Comparing multiple metaheuristic optimization models against a number of baseline allocation models and over multiple simulated time periods.

3. Finding a theoretical floor for the number of ambulances needed to handle acute and urgent incidents.

All the points mentioned above have been handled and included in the final model and will be covered in sections 4 through 6.

7.3 Limitations

Currently, our simulation model does not take into account the specified urgency level of the event, and all events are prioritized equally. This is in line with previous research [McCormack & Coates, 2015], and as discussed in section 2.1.1, the urgency level is only a best-guess based on the information available to the EMCC at the time. Some incidents are even deliberately over-triaged due to uncertainty related to the status of the patient. Because of this, there is a certain level of ambiguity involved in this labeling, and additionally, choosing whether to abandon an incident in favor of another incident rated as more urgent is a matter of dispatch policy, which is outside the scope of this study due to the complex human factors involved in these decisions.

In addition to prioritizing both acute and urgent incidents equally, the simulation model also disregards events classified as the least urgent (V1 and V2). These events can usually be delayed in the case of other more pressing emergencies, which would again be difficult to model or simulate, and are therefore considered outside the scope of this study. These incident types do, however, represent a significant portion of the demand that OAAD handles, and should optimally in some way be factored into a simulation model.

Furthermore, as noted in section 6.2.3, the performance of the MA may be limited due to its dependence on the same local search operator implemented in the SLS algorithm. The performance of this algorithm could potentially be improved by testing more improvement procedures.

7.4 Future work

This section will focus on future potential improvements that could be made towards the contributions of this thesis. The proposed future work is categorized into several subsections.

7.4.1 Improving the simulation model

It remains to be seen what can be achieved by introducing **multiple shift changes** during the simulation. Currently, only day and night cycles are implemented, but it would be interesting to see whether the performance can be further increased by splitting day and night shifts into multiple shifts. Additionally, different weekdays could have different ambulance allocations, or weekdays could have one allocation, while weekends are staffed differently, to better accommodate typical periodic spikes in EMS demand. More simulations could also be run over shorter periods of time (e.g., one simulation per weekday) to achieve this effect. This might introduce some problems with a too short simulation warm-up time, which is detailed in section 4.2.3.

7.4.2 Temporal travel time model

The travel time model used for the purpose of this study is a static model. However, more dynamic models, incorporating the changing traffic flow on various days and throughout the day, are available and could possibly be used to increase the validity of the model. Using a temporal model would also require a much larger amount of precomputed data compared to only calculating for a general average and this therefore represents a significant challenge to overcome in order to improve the travel model.

7.4.3 Synthesizing historic allocation data

This project has focused on evaluating ambulance allocations and using this to compare different allocation strategies. An interesting approach would be to extract historic allocation data from the data set as a comparison against the other methods. This would also be excellent for validating the model and getting a more accurate picture of the number of units available at a given time.

7.4.4 Survival Functions: Norwegian Heart Failure Registry

Initially, a great effort of this project was focused on finding better performance metrics to guide the optimization algorithms. Despite the fact that response time frequently appears in national guidelines and reports, it is not the best metric from a medical perspective, mainly because it is linear and diseases are not. In the literature, using for instance survival functions is seen as an improvement to using response times directly when optimizing EMS features [Erkut et al., 2008]. The partners for this project at OUH has mentioned that a statistical model with coefficients based on data from the Norwegian Heart Failure Registry is expected

to be published in the near future if all goes well. This would probably be a great addition to this project.

7.4.5 Dispatch policy through Reinforcement Learning

An interesting approach that has yet to be investigated is whether the simulation can be used as a "game" in order to test multiple different dispatch strategies throughout the day. A general weakness of most dispatching models used in simulations similar to the one proposed by this paper is that they use a *myopic*, or short-sighted, dispatch policy. Based on discussions with EMCC personnel, this decision is usually much more complex, with the dispatcher having to consider the effects of lost coverage for that area and the likelihood of future events occurring in the area where an ambulance is currently located.

According to employees at the EMCC, it would be very beneficial for their work if they got some quantification of the risks involved in dispatching a certain ambulance to an incident, with regards to lost coverage of that particular area. An interesting approach would be to treat the simulation created for the purpose of this study as a regular board game like chess. Using modern innovations like Deep Reinforcement Learning (RL) and Monte Carlo Tree Search (MCTS), one could possibly run hundreds or even thousands of short-term simulations in the instant an incident is registered in order to evaluate which ambulances should optimally be dispatched, given a known, estimated future demand. General purpose RL models using MCTS have already been shown to outperform state-of-the-art AIs in a variety of games such as Chess, Go and Shogi without prior knowledge of the domain [Silver et al., 2017]. In this case, the simulation could be considered a one-player game.

Bibliography

Amorim, M., Ferreira, S., & Couto, A. (2018). Emergency Medical Service Response: Analyzing Vehicle Dispatching Rules. *Transportation Research Record*, *2672*(32), 10–21. https://doi.org/10.1177/0361198118781645

Amorim, M., Ferreira, S., & Couto, A. (2020). Corrigendum to "How do traffic and demand daily changes define urban emergency medical service (uEMS) strategic decisions?: A multi-period survival approach" [JTH 12 (2019) 60–74] (Journal of Transport & Health (2019) 12 (60–74), (S2214140517307429), (10.1016/j.jth.2018.12.001)). https://doi.org/10.1016/j.jth.2019.05.009

Arnetz, B. B., Goetz, C. M., Arnetz, J. E., Sudan, S., Vanschagen, J., Piersma, K., & Reyelts, F. (2020). Enhancing healthcare efficiency to achieve the Quadruple Aim: An exploratory study. *BMC Research Notes*, *13*(1). https://doi.org/10.1186/s13104-020-05199-8

Aytug, H., & Saydam, C. (2002). Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study. www.elsevier.com/locate/dsw

Banks, J., II, J. S. C., Nelson, B. L., & Nicol, D. M. (2010). *Discrete-Event System Simulation, 5th New Internatinal Edition*. Pearson Education.

Behmanesh, E., & Pannek, J. (2021). A Comparison between Memetic Algorithm and Genetic Algorithm for an Integrated Logistics Network with Flexible Delivery Path. *SN Operations Research Forum*, *2*. https://doi.org/10.1007/s43069-021-00087-8

Berwick, D. M., Nolan, T. W., & Whittington, J. (2008). The Triple Aim: Care, Health, And Cost. *Health Affairs*, *27*(3), 759–769.

Boutilier, J. J., & Chan, T. C. (2020). Ambulance emergency response optimization in developing countries. *Operations Research*, *68*(5), 1315–1334. https://doi.org/10.1287/opre.2019.1969

Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*(3), 451–463. https://doi.org/10.1016/S0377-2217(02)00364-8

Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, *32*(1), 101–118. https://doi.org/10.1007/BF01942293

Comber, A. J., Sasaki, S., Suzuki, H., & Brunsdon, C. (2011). A modified grouping genetic algorithm to select ambulance site locations. *International Journal of Geographical Information Science*, *25*(5), 807–823. https://doi.org/10.1080/13658816.2010.501334

Deng, Y., Zhang, Y., & Pan, J. (2021). Optimization for locating emergency medical service facilities: A case study for health planning from China. *Risk Management and Healthcare Policy*, *14*, 1791–1802. https://doi.org/10.2147/RMHP.S304475

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (Second Edition, Vol. 53). Springer.

Erkut, E., Ingolfsson, A., & Erdogan, G. (2008). Ambulance location for maximum survival. *Naval Research Logistics*, *55*(1), 42–58. https://doi.org/10.1002/nav.20267

Garg, P. (2009). *A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm* (tech. rep. No. 1).

Guimarães, M. M., & Vinicius Cruzeiro Martins, F. (2018). A multiobjective approach applying in a Brazilian emergency medical service.

Harish Dayapule, D., Raghavan, A., Tadepalli, P., & Fern, A. (2018). *Emergency Response Optimization using Online Hybrid Planning* (tech. rep.). https://tinyurl.com/yd22wrjo

Hermansen, A. H., & Mengshoel, O. J. (2020). Spatio-Temporal Prediction of Emergency Medical Demand for Reduced Ambulance Travel Time.

Hermansen, A. H., & Mengshoel, O. J. (2021a). Forecasting Ambulance Demand using Machine Learning: A Case Study from Oslo, Norway.

Hermansen, A. H., & Mengshoel, O. J. (2021b). Machine Learning for Spatio-Temporal Forecasting of Ambulance Demand. A Norwegian Case Study. (June).

Jagtenberg, C. J., Bhulai, S., & van der Mei, R. D. (2017). Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health Care Management Science*, *20*(4), 517–531. https://doi.org/10.1007/s10729-016-9368-0

Jain, R. (2017). *Introduction to Simulation* (tech. rep.). Washington University. Saint-Louis. http://www.cse.wustl.edu/~jain/cse567-17/

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Kaveh, M., & Mesgari, M. S. (2019). Improved biogeography-based optimization using migration process adjustment: An approach for location-allocation of ambulances. *Computers and Industrial Engineering*, *135*, 800–813. https://doi.org/10.1016/j.cie.2019.06.058

Kochetov, Y. A., & Shamray, N. B. (2021). Optimization of the Ambulance Fleet Location and Relocation. *Journal of Applied and Industrial Mathematics*, *15*(2), 234–252. https://doi.org/10.1134/S1990478921020058

Leknes, H., Aartun, E. S., Andersson, H., Christiansen, M., & Granberg, T. A. (2017). Strategic ambulance location for heterogeneous regions. *European Journal of Operational Research*, *260*(1), 122–133. https://doi.org/10.1016/j.ejor.2016.12.020

Marla, L., Krishnan, K., & Deo, S. (2021). Managing EMS systems with user abandonment in emerging economies. *IISE Transactions*, *53*(4), 389–406. https://doi.org/10.1080/24725854.2020.1802086

McCormack, R., & Coates, G. (2015). A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival. *European Journal of Operational Research*, *247*(1), 294–309. https://doi.org/10.1016/j.ejor.2015.05.040

Mengshoel, O. J., Desai, R., Chen, A., & Tran, B. (2013). Will We Connect Again? Machine Learning for Link Prediction in Mobile Social Networks. http://snap.stanford.edu/data/.

Mengshoel, O. J., & Riege, J. (2022). *Understanding the Cost of Fitness Evaluation for Subset Selection : Markov Chain Analysis of Stochastic Local Search* (No. *1*), GECCO. https://doi.org/10.1145/3512290.3528689

Neri, F., & Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, *2*, 1–14. https://doi.org/10.1016/j.swevo.2011.11.003

Olsen, S., Kjøllesdal, J. K., Berlac, P. A., & Bárðarson, L. (2018). *The Nordic Emergency Medical Services: Project on Data Collection and Benchmarking* (tech. rep.).

Roa, J. C. P., Escobar, J. W., & Moreno, C. A. M. (2020). An online real-time matheuristic algorithm for dispatch and relocation of ambulances. *International Journal of Industrial Engineering Computations*, *11*(3), 443–468. https://doi.org/10.5267/j.ijiec.2019.11.003

Sariyer, G., Ataman, M. G., Akay, S., Sofuoglu, T., & Sofuoglu, Z. (2017). An analysis of Emergency Medical Services demand: Time of day, day of the

week, and location in the city. *Turkish Journal of Emergency Medicine*, *17*(2), 42–47. https://doi.org/10.1016/j.tjem.2016.12.002

Sasaki, S., Comber, A. J., Suzuki, H., & Brunsdon, C. (2010). Using genetic algorithms to optimise current and future health planning - the example of ambulance locations.

Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, *219*(3), 611–621. https://doi.org/10.1016/j.ejor.2011.10.043

Schultes, D., Sanders, P., & Möhring, R. (2008). *Route Planning in Road Networks* (Doctoral dissertation).

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. http://arxiv.org/abs/1712.01815

Song, J., Li, X., & Mango, J. (2020). Location Optimization of Urban Emergency Medical Service Stations: A Hierarchical Multi-objective Model with a New Encoding Method of Genetic Algorithm Solution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12473 LNCS*, 68–82. https://doi.org/10.1007/978-3-030-60952-8{_}7

Strand, G.-H., & Bloch, V. V. H. (2009). Statistical grids for Norway. www.ssb.no

Tassone, J., & Choudhury, S. (2020). A Comprehensive Survey on the Ambulance Routing and Location Problems. http://arxiv.org/abs/2001.05288

The Norwegian Directorate of Health. (2021). AMK - Tid fra AMK varsles til ambulanse er på hendelsessted [EMCC - Time from EMCC is notified until the ambulance is at the scene]. https://www.helsedirektoratet.no/statistikk/kvalitetsindikatorer/akuttmedisinske-tjenester-utenfor-sykehus/tid-fra-amk-varsles-til-ambulanse-er-pa-hendelsessted

The Norwegian Labour Inspection Authority. (2020). *Arbeidstilsynets tilsyn og veiledning i ambulansetjenesten i 2018-2019 [The Norwegian Labour Inspection Authority's supervision and guidance in the ambulance service in 2018-2019]* (tech. rep.). https://www.arbeidstilsynet.no/globalassets/om-oss/forskning-og-rapporter/rapporter-fra-tilsynsprosjekter/arbeidstilsynets-tilsyn-og-veiledning-i-ambulanse-tjenesten-i-2018-2019.pdf

Toregas, C., Swain, R., Revelle, C., & Bergman, L. (1971). *Guidelines for the Practice of Operations Research* (tech. rep. No. 6). https://about.jstor.org/terms

Walsh, H. J. (2019). Impact of standby points on the workday of ambulance personnel A study on motivation, work environment and meaningful work of ambulance personnel in Oslo and Akershus counties. http://www.duo.uio.no/

Wang, J., Wang, Y., & Yu, M. (2020). A multi-period ambulance location and allocation problem in the disaster. *Journal of Combinatorial Optimization*. https://doi.org/10.1007/s10878-020-00610-3

Weise, T. (2009). Global optimization algorithms: Theory and Application. *Self-Published Thomas Weise*, *361*.

Wolpert, D. H., & Macready, W. G. (1997). *No Free Lunch Theorems for Optimization* (tech. rep. No. 1).

Yang, W., Su, Q., Huang, S. H., Wang, Q., Zhu, Y., & Zhou, M. (2019). Simulation modeling and optimization for ambulance allocation considering spatiotemporal stochastic demand. *Journal of Management Science and Engineering*, *4*(4), 252–265. https://doi.org/10.1016/j.jmse.2020.01.004

Yue, Y., Marla, L., Krishnan, R., & John Heinz, H. (2012). An Efficient Simulation-Based Approach to Ambulance Fleet Allocation and Dynamic Redeployment. www.aaai.org

Zhang, Z., Liu, M., & Lim, A. (2015). A memetic algorithm for the patient transportation problem. *Omega (United Kingdom)*, *54*, 60–71. https://doi.org/10.1016/j.omega.2015.01.011

Zhen, L., Wang, K., Hu, H., & Chang, D. (2014). A simulation optimization framework for ambulance deployment and relocation problems. *Computers and Industrial Engineering*, *72*(1), 12–23. https://doi.org/10.1016/j.cie.2014.03.008

Schjølberg & Bekkevold

Simulation and Optimization of Emergency Medical Services in Oslo and Akershus

NTNU
Norwegian University of
Science and Technology

Oslo
University Hospital