Eindhoven University of Technology

MSc. Thesis

In partial fulfillment of the requirements for the degree of Master of Science in Operations Management and Logistics

Formalization and improvement of ambulance dispatching in Brabant-Zuidoost

Author
N.B.J.M. Theeuwes (0803292)

Supervisors
prof. dr. ir. G.J. van Houtum (TU/e)
drs. B.P.M.T. Gadet (GGD BZO)
dr. Y. Zhang (TU/e)
dr. C. Fecarotti (TU/e)
dr. ir. M.A.A. Boon (TU/e)

Eindhoven, Tuesday 7th May, 2019

Eindhoven University of Technology
School of Industrial Engineering
Series Master Theses Operations Management and Logistics

Key words: Ambulance management, EMS, ambulance dispatching, machine learning, decision tree induction, discrete event simulation

Abstract

The performance of ambulance services in the Netherlands has been consistently below the nationally-set target throughout the last years. Combining this with the ongoing increase in demand for these services and the severe shortages of ambulance personnel, stresses the need for steps to be taken towards improved efficiency. In this thesis we develop an alternative dispatch policy with the objective to improve the on-time performance of highly urgent ambulance requests, by capturing current dispatch decisions and building upon them through four potential enhancements. Current dispatch practices in the Dutch EMS region Brabant-Zuidoost are captured using decision tree induction and a unique post-processing phase, resulting in a formal model that is both concise and able to accurately predict current dispatch decisions. Subsequently, four potential enhancements to the current dispatch process are formulated, based on a combination of insights from current practices, discussions with dispatch agents and available literature. These four potential enhancements are evaluated, both individually and combined, in an advanced simulation that is able to realistically capture actual ambulance dynamics. Results show that complementing the current dispatch policy with *consistently redispatching* ambulances from a less urgent to a more urgent request and *reevaluation* of active dispatch decisions upon service completion of an ambulance yields a significant improvement of the on-time performance of highly urgent ambulance requests of 0.77 percent points. Contrary to measures that increase available ambulance capacity to improve performance, adjusting the operational dispatch process to better utilize the existing capacity is virtually free and instantaneous. A similar performance gain as a result of enhancing the current dispatch policy is expected for other (Dutch) EMS regions.

Executive Summary

The performance of ambulance services in the Netherlands being consistently below the nationally-set target (Ambulancezorg Nederland, 2018), combined with the ongoing increase in demand for these services (Kommer & Zwakhals, 2016) and the severe shortages of ambulance personnel (Ambulancezorg Nederland, 2019), stresses the need for steps to be taken towards improved efficiency. Advances in ambulance logistics will contribute towards the provision of sufficient emergency medical care, given the available resources.

Problem Statement
Even more extreme than at a national level in the Netherlands, in the emergency medical services (EMS) region of Brabant-Zuidoost the fraction of highly urgent (A1) requests with a response time of less than 15 minutes has been consistently below the nationally-set target of 95% throughout the last years, while performance of moderately urgent (A2) requests has consistently exceeded its target of 95% with a response time of less than 30 minutes. These statistics suggest that there is potential in improving the performance of A1 requests at the expense of performance of A2 requests by adapting dispatch policies accordingly. However, literature on operational ambulance management has mainly been focused on relocation policies, aimed at repositioning ambulances to improve preparedness for dispatches to requests arriving in the near future. In both the design and the evaluation of these relocation policies, it is predominantly assumed that ambulances are dispatched according to the 'closest-idle' policy, regardless of the urgency of the ambulance request. The limited number of studies exploring alternative dispatch policies are often of theoretical nature, failing to evaluate the effect of these alternative dispatch policies on performance in a realistic(ally sized) system.

Research topic
The objective of this thesis is to improve the on-time performance of A1 requests in the EMS region of Brabant-Zuidoost through improvement of the dispatch policy by building upon current practices. Hereto, current dispatch practices are captured, after which four potential enhancements to the process are formulated. Contrary to the development of an improved dispatch policy from scratch, these enhancements complement, rather than replace, current dispatch practices. This ensures that practical considerations are incorporated in the developed dispatch policy, and that it is in line with the way in which dispatch agents currently work, which is expected to foster adoption in practice. This approach ensures that the resulting policy can be implemented quickly without the need for (major) software changes.

Approach and results
While existing studies, aiming to improve performance through alternative dispatch policies, either alter the commonly-assumed 'closest-idle' dispatch policy or develop a dispatch policy from scratch, this thesis formally captured the way in which dispatch decisions are currently made with the goal of using this policy as a practically relevant basis to build upon by extending it with additional or adapted decision rules. A combination of decision tree induction and a unique post-processing phase resulted in a formal model that is both concise and able

to accurately predict current dispatch decisions. The resulting model enriches the commonly assumed closest-idle dispatch policy through the use of penalty values that reflect the risk associated with certain ambulance characteristics, such as its status, region and time until the end of its shift.

Based on a combination of insights from the capturing effort, discussions with dispatch agents, and available literature, four potential enhancements to the current dispatch policy were formulated: *consistently redispatching* ambulances to highly urgent (A1) requests, *reevaluating* active dispatch decisions upon service completion of an ambulance, dispatching the ambulance resulting in *minimum coverage reduction*, and *postponing dispatches* to less urgent requests in case of limited ambulance availability.

Subsequently, a realistic simulation was developed that is able to accurately capture the complex dynamics of a life size ambulance system to evaluate these potential enhancements to the current dispatch policy within a reasonable computation time. Existing studies evaluating alternative dispatch policies generally resort to simplifying modeling choices and assumptions in the development of a simulation, mainly relating to the size of the problem and the dynamicity of request arrivals and characteristics. The limitations of the simulations used in these studies were identified and solved in our developed simulation, such that it is able to accurately deal with the dynamic arrival of ambulance requests of multiple urgency levels, dynamic ambulance capacity, realistic relocation decisions and a wide range of practical considerations. Lastly, the captured current dispatch process allowed us to be the first to evaluate alternative dispatch policies by comparing the simulated performance to that of a benchmark that resembles current practices. The development of this advanced simulation model, combined with the use of a practically relevant benchmark, allowed us to draw accurate conclusions regarding the expected effect of the proposed enhancements on actual performance in practice.

Using the developed simulation, the effect of the four potential enhancements to the current dispatch policy was evaluated. We showed that significant improvement to the on-time performance of highly urgent (A1) ambulance requests can be obtained by enhancing the dispatch process. More specifically, for the EMS region of Brabant-Zuidoost, this measure can be improved by 0.77 percent points through enhancing current dispatch practices by *consistently redispatching* ambulances that are on its way to a less urgent request to a more urgent request and *reevaluating* active dispatch decisions upon service completion of an ambulance, such that this ambulance can be dispatched instead if this leads to a significant improvement of response time. This improvement comes at the expense of a decrease of 0.33 percent points of the on-time performance of A2 requests, easily keeping it above its threshold target of 95% with a response time of less than thirty minutes. Both enhancements encourage dispatching an ambulance from 'the field', i.e. an ambulance that is not at a station, making them especially beneficial for (postal code) areas that cannot be reached in time from any, or most, ambulances stations, such as those near the region borders. Lastly, simulation results illustrated that an equivalent increase of the on-time performance of highly urgent requests would require the addition of over seven extra eight-hour shifts on a weekly basis. Adjusting the operational dispatch process to better utilize available capacity is both virtually free and instantaneous, contrary to the expansion of capacity through the addition of ambulance shifts. Enhancing the dispatch policy is expected to yield a similar performance gain in other EMS regions, especially those in the Netherlands.

Contents

List of Figures

List of Tables

Glossary

A1 request: Ambulance request of the highest urgency. Requires an ALS ambulance and the performance target is 95% of requests with a response time of less than 15 minutes.

A2 request: Ambulance request of moderate urgency. Requires an ALS ambulance and the performance target is 95% of requests with a response time of less than 30 minutes.

Answering agent: Agent in dispatch center who executes the triage procedure.

ALS ambulance: Advanced life support ambulance, required for A1, A2, and B1 requests.

AZN: Ambulancezorg Nederland, representing all Dutch EMS regions.

B1 request: Ambulance request for non-urgent transportation of a non-stable patient, requiring an ALS ambulance and generally ordered well-beforehand.

B2 request: Ambulance request for non-urgent transportation of a stable patient, requiring a BLS ambulance and generally ordered well-beforehand.

Base station: Station where ambulance shifts can start and should end.

BLS ambulance: Basic life support ambulance, required for B2 requests.

BZO: Brabant-Zuidoost, the EMS region this thesis focuses on.

BNO: Brabant-Noord, neighbouring- and sharing a dispatch center with BZO.

BO: Brabant-Oost, regions BZO and BNO together.

Chute time: Time between the moment an ambulance is dispatched and it departs.

Dispatch agent: Agent in dispatch center who dispatches and relocates ambulances.

Dispatch proposal: List of available ambulances ranked on driving time, see Section 2.3.

EHGV: Request without transportation to a hospital (Dutch: Eerste Hulp Geen Vervoer).

EMS: Emergency Medical Services.

On-time: Request with a response time less than the urgency-dependent threshold.

RAV: Organization responsible for regional dispatch center and ambulance fleet.

Response time: Time from request arrival to ambulance arrival at the request location.

RIVM: National Institute for Public Health and Environment, in the Netherlands.

Standby station: Station where an ambulance can stand idle, but no shifts start.

Triage: A procedure that determines the urgency of a request in a dispatch center.

Urgency: Determines the required ambulance type and performance target of a request.

1 | Introduction

The performance of ambulance services in the Netherlands being consistently below the nationally-set target (Ambulancezorg Nederland, 2018), combined with the ongoing increase in demand for these services (Kommer & Zwakhals, 2016) and the severe shortages of ambulance personnel (Ambulancezorg Nederland, 2019), stresses the need for steps to be taken towards improved efficiency. Advances in ambulance logistics will contribute towards the provision of sufficient emergency medical care, given the available resources.

Bélanger, Ruiz and Soriano (2018) provide an overview of decision problems related to *emergency medical services* (EMS) management on a strategic, tactical, and operational level. They state that the research focus has recently shifted from strategic and tactical problems, such as determining optimal ambulance station locations and fleet size, to the more dynamic operational problems related to EMS management. Such operational problems in EMS literature include both ambulance dispatching and ambulance relocation, with the aim of maximizing the fraction of ambulance requests with a *response time* below a certain threshold time, or the *on-time performance*. Here, response time is defined as the time between the moment an ambulance request arrives at a dispatch center and the moment the ambulance arrives at the request location (Henderson, 2011). The threshold time of a request depends on the urgency level it has been assigned and is set nationally. Operational decisions on dispatching and relocation problems in a region are both made by a dispatch agent in a dispatch center.

1.1 Problem statement

On a national level in the Netherlands, the fraction of highly urgent ambulance requests, so-called *A1 requests*, with a response time of less than 15 minutes has been consistently below the nationally-set target of 95% throughout the last years, with a performance of 92.4% in 2017 (Ambulancezorg Nederland, 2018). On the other hand, performance of moderately urgent requests, called *A2 requests*, has consistently exceeded its target of 95% of requests with a response time within 30 minutes, with 96.1% of requests being served on-time. In the EMS region of 'Brabant-Zuidoost', similar performance was observed, though more extreme. In this region, only 91.7% of highly urgent A1 requests were on-time in 2017, while for less urgent A2 requests a performance of 97.6% was achieved in the same year. These statistics suggest that in Dutch EMS regions, such as Brabant-Zuidoost, there is potential in improving the performance of A1 requests at the expense of performance of A2 requests by adapting dispatch policies accordingly.

However, literature on operational ambulance management has mainly been focused on relocation policies. Such relocation policies are designed in an attempt to reposition ambulances as to improve preparedness for dispatches to requests arriving in the near future. Relocation movements are generally initiated upon reduced preparedness resulting from a decrease in the number of idle ambulances, i.e. due to the dispatch of an ambulance. While such relocation policies are designed to tackle the preparedness reduction caused by a dispatch decision, at-

tention for the actual dispatch procedure in these studies is limited. In both the design and the evaluation of these relocation policies it is predominantly assumed that ambulances are dispatched according to a 'closest-idle' policy, regardless of the urgency of the incident the ambulance is dispatched to (Theeuwes, 2018).

Yet, it has been shown in literature that this policy is suboptimal when minimizing the fraction of late requests (Jagtenberg, Bhulai & van der Mei, 2017). Our observations in the dispatch center of the EMS region of Brabant-Zuidoost have shown that dispatch agents often deviate from this policy in an attempt to improve performance. A variety of factors, shaped by experience and expertise, cause dispatch agents to deviate from this commonly assumed policy. This suggests that, at least in this region, the limiting nature of this policy is recognized in practice.

Moreover, the limited number of studies exploring alternative dispatch policies are often of theoretical nature, failing to evaluate the effect of these alternative dispatch policies on performance in a realistic(ally sized) system. Furthermore, most studies do not distinguish between the urgency of requests, let alone intend to improve performance of more urgent requests at the expense of less urgent requests given the available ambulance capacity. These limitations of the (scarce) current literature on alternative dispatch policies, prevent one from drawing sound conclusions regarding the potential of these alternative dispatch policies in practice.

1.2 Research topic

Aringhieri, Bruni, Khodaparasti and Van Essen (2017) state that knowledge obtained by dispatchers in practice can be very useful in the development of reliable dispatching policies. The objective of this thesis is to improve the *on-time* performance of highly urgent (A1) requests of the EMS region of Brabant-Zuidoost (BZO) by capturing and building upon current dispatch practices. Here, building upon current practices entails extending it with a limited number of additional or adapted decision rules that are expected to contribute towards our objective of improving the *on-time* performance of urgent requests. Contrary to the development of an improved dispatch policy from scratch, these building blocks complement, rather than replace, current dispatch practices. This ensures that practical considerations are incorporated in the developed dispatch policy, and that the developed policy is in line with the way in which dispatch agents currently work, which is expected to foster adoption in practice. Lastly, this approach ensures that the resulting policy can be implemented quickly without the need for (major) software changes.

Decision makers are often not completely aware of the reasoning behind their expert judgments, making it hard for them to verbally express their decision process (Lafond, Tremblay & Banbury, 2013). However, mental decision models can be formally approximated through decision analysis techniques using statistical models or machine learning algorithms. Following Maghrebi, Sammut and Waller (2013), decision tree induction is applied to capture the current dispatch policy. The transparent and interpretable nature of the resulting decision tree allows us to gain insights into the captured dispatch process such that it can be built upon, i.e. enhancing it with the objective to improve performance. Four possible enhancements to the current dispatch policy are formulated, based on a combination of insights from the capturing effort, discussions with dispatch agents, and available literature. The potential of the four possible enhancements to the current dispatch process is assessed using a realistic,

discrete event-based simulation study. Besides using the captured current dispatch policy as a basis to build upon with the objective to improve performance of highly urgent (A1) requests, it is used as a benchmark in the developed simulation. The use of a benchmark that resembles current practices enables us to accurately draw conclusions regarding the potential of the evaluated improvements in practice. This approach is summarized in Figure 1.1.

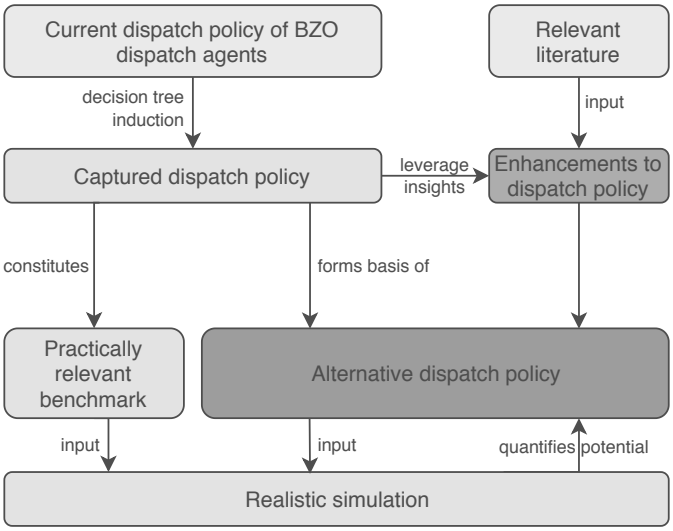

Figure 1.1: Overview of thesis approach

Summarizing, the main contribution of this thesis is fourfold:

- We are the first to approach the development of an (alternative) ambulance dispatch policy by capturing current dispatch practices and using it as a practically relevant basis to build upon. While existing studies, aiming to improve performance through alternative dispatch policies, either alter the commonly-assumed 'closest-idle' dispatch policy or develop a dispatch policy from scratch, this thesis formally captures the way in which dispatch decisions are currently made with the goal of using this policy as a basis to build upon by extending it with additional or adapted decision rules. Furthermore, not only is this the first attempt to formally capture current ambulance dispatching decisions using machine learning, also a unique post-processing phase is applied resulting in a formal model that is both concise and accurate.
- We formulate four potential enhancements to the dispatch process based on a combination of insights from the capturing effort, discussions with dispatch agents, and available literature. These enhancements are formulated as building blocks onto the current dispatch policy, such that they complement rather than replace current dispatch practices, either individually or combined. These four potential enhancements entail *consistently redispatching* ambulances to highly urgent (A1) requests, *reevaluating dispatch decisions* upon service completion of an ambulance, dispatching the ambulance resulting in *minimum coverage reduction*, and *postponing dispatches* to less urgent requests in case of limited ambulance availability.
- We develop a realistic simulation that is able to accurately capture the complex dy-

namics of ambulance systems to evaluate these potential enhancements to the captured dispatch policy. While existing studies evaluating alternative dispatch policies resort to the use of simulations of highly theoretical nature, both in terms of size and assumptions, in this thesis an advanced simulation model is developed that can deal with a real life size problem within a reasonable computation time. The developed simulation is able to realistically deal with ambulance requests of multiple urgency levels (including non-urgent transports), dynamic ambulance capacity, realistic relocation decisions, and practical considerations such as the end of ambulance shifts. Furthermore, the simulation is able to accurately reflect ambulance request patterns through a request generation process that is both stochastic and dynamic in terms of the arrival times and request characteristics. Lastly, the captured current dispatch process allows us to be the first to evaluate alternative dispatch policies by comparing simulated performance to that of a benchmark that resembles current practices. The development of this advanced simulation model, combined with the use of a practically relevant benchmark, allows us to draw accurate conclusions regarding the expected effect of the proposed enhancements on actual performance in practice.

- We quantify the effect of the four potential enhancements to the dispatch policy by using the developed simulation. We show that significant improvement to the on-time performance of highly urgent (A1) ambulance requests can be obtained by enhancing the dispatch process. More specifically, for the EMS region of Brabant-Zuidoost, this measure can be improved by 0.77 percent points through enhancing current dispatch practices by *consistently redispatching* ambulances that are on its way to a less urgent request to a more urgent request and *reevaluating* active dispatch decisions upon service completion of an ambulance, such that this ambulance can be dispatched instead if this leads to a significant improvement of response time. Results show that this improvement to the on-time performance of highly urgent (A1) requests comes at the expense of a decrease of 0.33 percent points of the on-time performance of A2 requests. Contrary to intuitive measures to improve performance, such as increasing ambulance capacity, adjusting the operational dispatch process to better utilize available capacity is both virtually free and instantaneous. Similar effects on performance are expected for other EMS regions, specifically those in the Netherlands.

1.3 Outline of thesis

The structure of the remainder of this thesis is as follows: Chapter 2 provides a background to this research, consisting of information on the Dutch emergency medical services (EMS) system in the Netherlands, specific characteristics of the BZO region, and an overview of relevant literature on both the application of machine learning to capture decision processes, and dispatch policies in ambulance management. Subsequently, Chapter 3 goes into the formalization of the current dispatch policy in the BZO region and lists the proposed enhancements to this process. Chapter 4 presents the implemented simulation and the evaluation results of the alternative dispatch policies. Lastly, Chapter 5 contains the conclusion of this work, as well as suggestions for further research.

2 | Background information

This section provides a background to this research. First, the structure of the emergency medical services (EMS) system in the Netherlands is described in Section 2.1, after which an outline of the EMS process is given in Section 2.2. Section 2.3 further elaborates on the dispatch proposal, on which dispatch decisions are based. Section 2.4 goes into relevant performance measures and Section 2.5 presents some statistics and trends of EMS services in the Netherlands, after which Section 2.6 discusses some specific characteristics of the Dutch EMS region Brabant-Zuidoost (BZO), which is the focus of this research. This chapter concludes with Section 2.7 providing a brief overview of relevant literature concerning both using machine learning to capture decision processes and dispatch policies in ambulances management.

2.1 EMS structure

Emergency medical services (EMS) in the Netherlands are organized on a regional level. According to the *temporary law ambulance care* (Dutch: Tijdelijke Wet Ambulancezorg), which came into effect in 2013 and was recently extended until 2021, the Netherlands is split into 25 EMS regions, one of which is Brabant-Zuidoost. In each of these regions an umbrella organization called a RAV (Dutch: regionale ambulancevoorziening) is responsible for managing its own regional dispatch center and ambulance fleet, which may be either privately or publicly owned. Each region contains a number of *ambulance stations* at which ambulances are positioned, the location of which is based on a national reference framework of the National Institute for Public Health and Environment (Dutch: RIVM) (Kommer & Zwakhals, 2016). The mandate for the RAV of each region from 2021 onwards is expected to be granted through a tender process, which increases the need to meet performance targets, see Section 2.4.

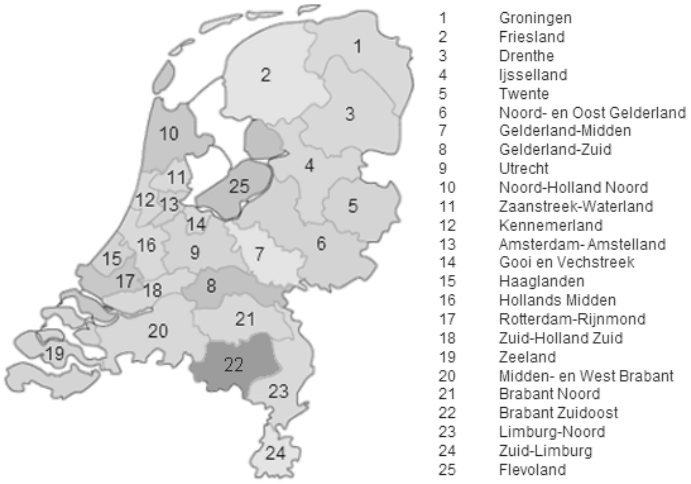

Figure 2.1: Overview of the 25 EMS regions in the Netherlands

2.2 EMS process

The EMS process starts with a request for ambulance care. This request can either be issued by a civilian calling the national emergency services number (112 in Europe) or by a medical professional (e.g. a general practitioner) directly calling the regional dispatch center. Incoming calls to the national emergency services number are forwarded to the required regional emergency service center in the region corresponding to the location of the emergency.

2.2.1 Triage

In case a call is placed by a civilian, the agent answering the call in the regional dispatch center determines the condition of the patient through a *triage procedure*. This triage procedure consists of a dynamic system showing the agent which questions to ask. Based on the resulting patient condition, the system assigns an urgency to the request. This urgency determines both the required ambulance type and the performance target of the request. While the answering agent can decide to manually increase the urgency level of a request based on factors not captured by the triage procedure, it is not possible to decrease. In case a call is placed by a medical professional, he or she may determine the urgency of the request and the triage procedure may be skipped. While patients are only allowed to request medical assistance for urgent emergencies, resulting in an urgency level of either *A1* or *A2*, medical professionals can also request non-urgent medical transport, which is either of stable (urgency *B2*) or non-stable (urgency *B1*) patients.

Table 2.1: Overview of urgency levels used in the Dutch EMS system

Urgency level	Definition	Required ambulance	Performance target
A1	Acute threat to vital functions of the patient	ALS ambulance	95% with response time < 15 min.
A2	No life threatening condition, but possibly (severe) injuries	ALS ambulance	95% with response time < 30 min.
B1	Non-urgent transportation request (ordered transportation)	ALS ambulance	None
B2		BLS ambulance	None

As indicated in Table 2.1, different types of ambulances can be distinguished. Advanced life support (ALS) ambulances contain all means to diagnose a patient and start treatment, while Basic life support (BLS) ambulances are meant to transport stable patients between their home and a hospital or between hospitals. ALS ambulances are required for emergencies with an A1 or A2 urgency, as well as for non-urgent transportation requests of non-stable patients, i.e. with a B1 urgency.

2.2.2 Dispatching

If the request for an ambulance is honored, the request and the corresponding urgency is forwarded to a *dispatch agent*, while the *answering agent* continues to provide the caller with first aid assistance if necessary. In case of an urgent (A1 or A2) call, the dispatch agent receives a *dispatch proposal* from the system, based on which the agent selects an ambulance to respond to the request. The crew of the selected ambulance receives a notification, which

includes the request's location, urgency and patient condition, and which may be updated during traveling based on newly obtained information by the answering agent.

The ambulance departs towards the request location as soon as possible. The time between the moment the ambulance is dispatched and the moment it departs for the request location is called the *chute time*. Only in case of a highly urgent (A1) request, the ambulance is allowed to use optical and sound signals and to exceed speed limits. After arriving at the emergency's location, the patient is treated. If deemed necessary by the ambulance crew, the patient is transported to a hospital. After transferring the patient to the correct department in the hospital, the ambulance declares itself *idle* again. If no transport of the patient is necessary, a so-called *EHGV* (Dutch: Eerste Hulp, Geen Vervoer) request, the ambulance is idle after treatment. Subsequently, the dispatch center either sends the ambulance back to any of the stations or immediately dispatches it to a new request.

On the other hand, non-urgent medical transports are requested for a specific time up to one week in advance. Due to its non-urgent nature, B-level requests are often made well in advance, allowing these rides to be scheduled. However, requests with B1 urgency still require an ALS ambulance because the patient to be transported is unstable and may need medical care during the transport. Since requests with B2 urgency are served by BLS ambulances, these can be planned independently from requests with other (higher) urgencies. However, it may occur that BLS capacity is not sufficient for B2 requests, in which case they are served by ALS ambulances. Non-urgent medical transports are generally dispatched around its requested time whenever sufficient ALS ambulances are available.

2.2.3 Relocating

Besides determining to which station to send an ambulance after it completes service of a request, the dispatch center has the possibility to relocate idle ambulances between stations to improve coverage of the region. However, the shift of an ambulance crew starts and ends at the same station: their *base station*.

Throughout this process ambulances communicate their status to the dispatch center each time it changes. The possible statuses and transitions between them are indicated in the process overview in Figure 2.2.

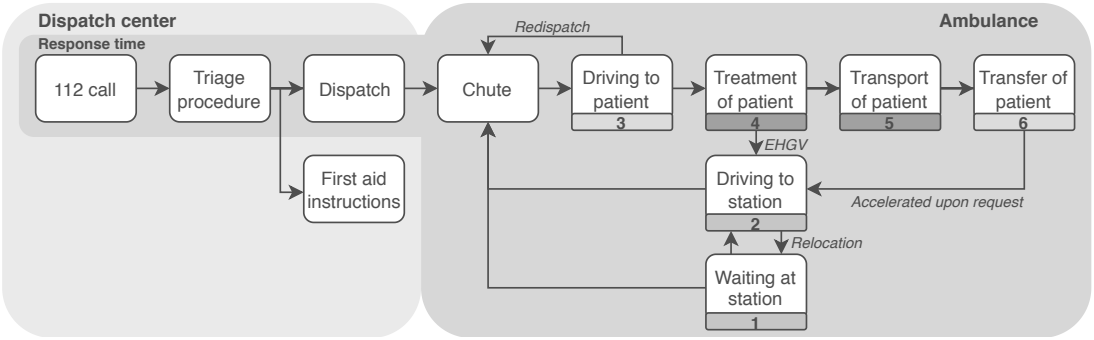

Figure 2.2: EMS (default) process overview, including ambulance statuses

2.3 Dispatch proposal

As described in Section 2.2, a dispatch agent is aided in the dispatching process by a dispatch proposal. In the region Brabant-Zuidoost (BZO) the standard dispatch proposal algorithm of the national dispatch system (Dutch: Geïntegreerd Meldkamer Systeem (GMS)) is used to generate such proposals for urgent (A1 or A2) incidents.

In the dispatching proposal algorithm of BZO the set of ambulances available to be dispatched to a request depends on its urgency. Regardless of the request's urgency this set includes all idle ambulances, i.e. those driving to, or waiting at, a station. Besides idle ambulances, this set includes ambulances which have already been dispatched to a less urgent request, but did not arrive at that request's location yet. Lastly, this set includes ambulances that have arrived at a hospital and are busy transferring a patient. While these ambulances are not idle yet, they might be requested to accelerate the transfer process such that they can be dispatched to a new ambulance request. Since dispatch agents have the possibility to request assistance from neighbouring EMS regions, these ambulances are also included in the dispatch proposal.

An ordered list of all ambulances available for dispatch is created based on an increasing driving time to the request location. Since driving times are rounded to whole minutes a tie might occur, in which case the tied ambulances are ordered based on the (as the crow flies) distance to the request location. If ambulances are still tied, e.g. if their current location is the same, the tied ambulances are ordered based on the time that passed since their status was last updated. This ensures that the longest waiting ambulance at a station receives a higher ranking than an ambulance that just arrived, in an attempt to smooth workload.

2.4 Performance measures

The main performance measure RAVs are evaluated on is the fraction of requests with a response time within a certain threshold time, which differs per urgency level: *the on-time* performance. Table 2.1 shows the performance target for each urgency level. The response time lasts from the moment a call is answered by the regional dispatch center and the moment an ambulance arrives at the patient's location (Henderson, 2011). This implies that the response time is not only made up by the driving time of the dispatched ambulance, but also by the processes taking place in the dispatch center and the chute time (i.e. the time between notifying the dispatched ambulance and the moment the ambulance actually departs to the emergency location). Figure 2.2 shows all steps making up the response time.

2.5 Statistics and trends

All RAVs in the Netherlands are jointly represented by an organization called Ambulancezorg Nederland (AZN). Each year AZN publishes the figures of ambulance care in the Netherlands, both at national and regional level. Table 2.2 shows some statistics of EMS operations in the Netherlands and how they developed over time (Ambulancezorg Nederland, 2018).

The most evident trend in ambulance care figures is the growth in demand. Especially the number of urgent requests (i.e. A1 and A2 urgency) has been growing consistently over the past year. This is mostly caused by demographic factors, such as overall growth and aging of the Dutch population, but also changes in Dutch medical care play a role. Increasing pressure

on hospitals causes patients to be discharged earlier, while they might still need additional care. Lastly, the introduction of stricter triage protocols has also impacted the number of urgent deployments (Ambulancezorg Nederland, 2017).

Moreover, Table 2.2 shows that the fraction of A1 requests with a response time under 15 minutes has been consistently below its target of 95% on a national level. The performance target of A2 requests, on the other hand, was met during each of the past five years, showing that improvement of performance is especially required for A1 requests. Furthermore, it can be seen that the growth in demand for ambulance care exceeds the increase in the number of ambulances. While the total number of honored requests increased by 14.7% over the past five years, the number of ambulances only increased by 6.2%. This limited increase in resources is mainly caused by the extreme shortage of personnel (Ambulancezorg Nederland, 2019).

Table 2.2: EMS statistics in the Netherlands (2013 - 2017)

	2017	**2016**	**2015**	**2014**	**2013**
Nr. of ambulances	790	780	752	755	744
Total budget (M€)	592	565	551	500	486
Total nr. of honored requests	1,313,103	1,313,251	1,253,294	1,190,320	1,144,780
Nr. of A1 requests	611,193	632,875	610,152	579,784	541,164
A1 with RT <15 min. (%)	92.4	93.4	93.4	93.4	92.6
A1 mean RT (min:sec)	09:41	09:26	09:25	09:29	09:39
Nr. of A2 requests	364,421	340,056	310,190	288,924	274,907
A2 with RT <30 min. (%)	96.1	96.6	96.6	96.7	96.1
A2 mean RT (min:sec)	15:07	14:52	14:55	14:56	15:26
Nr. of B requests	337,489	340,320	332,952	321,612	328,709

2.6 Region Brabant-Zuidoost

The Dutch EMS region of Brabant-Zuidoost (BZO) is the focus of this research. This region is located in the south of the Netherlands (see Figure 2.1, region 22). In 2017 the requests in this region amounted to just over four percent of the national total. Furthermore, performance measures show that BZO consistently performs worse in terms of response time targets for A1 requests compared to national performance, while performance is consistently better for A2 requests (Ambulancezorg Nederland, 2018). This shows that focus should shift towards improving performance of A1 requests.

Table 2.3: EMS statistics in the region Brabant-Zuidoost (2013 - 2017)

	2017	**2016**	**2015**	**2014**	**2013**
Total nr. of honored requests	52,874	52,601	50,252	46,983	43,646
Nr. of A1 requests	25,122	26,473	26,139	23,932	19,391
A1 with RT <15 min. (%)	91.7	92.8	93	92	94
A1 mean RT (min:sec)	09:42	09:35	09:36	09:55	09:27
Nr. of A2 requests	15,164	13,507	11,983	11,040	11,929
A2 with RT <30 min. (%)	97.6	98.0	97.3	97	97
A2 mean RT (min:sec)	14:42	14:14	14:24	14:59	15:46
Nr. of B requests	12,588	12,621	12,130	12,011	12,326

Figure 2.3 shows a schematic overview of the region Brabant-Zuidoost. A distinction can be made between base stations and standby stations. While shifts start and end at base stations, dispatch agents may decide to send an idle ambulance to one of the standby stations at any time during its shift. This is done to improve coverage of the region. Lastly, the four hospitals in the region are shown on the map, since this is where ambulances often become idle again.

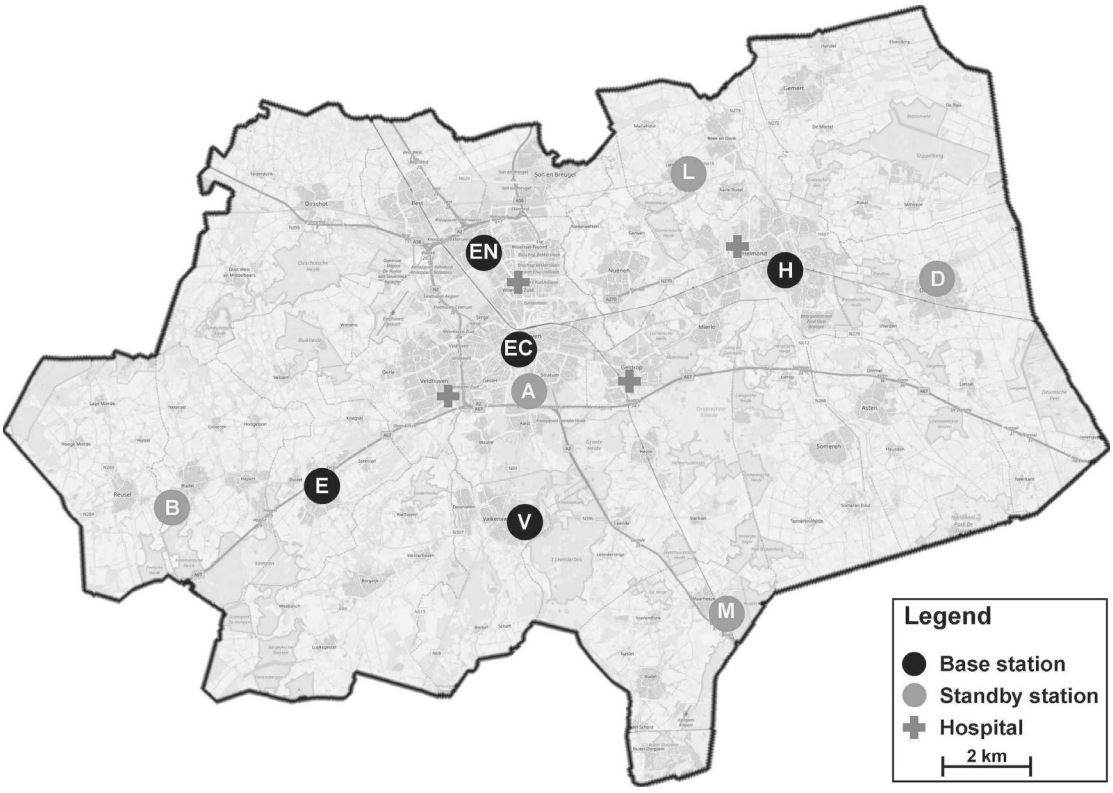

Figure 2.3: Map of the EMS region Brabant-Zuidoost
H: Helmond, EN: Eindhoven Noord, EC: Eindhoven Centrum, E: Eersel, V: Valkenswaard, A: Aalsterweg, B: Bladel, M: Maarheeze, D: Deurne, L: Lieshout

2.7 Overview of relevant literature

In this section a brief overview of literature relevant to our research is provided. First, Section 2.7.1 discusses existing studies that use machine learning techniques to capture decision processes from behavioral data, after which Section 2.7.2 provides an overview of available literature on alternative dispatch methods in ambulance management.

2.7.1 Capturing decision processes using machine learning

With the ever-increasing amount of data, techniques that can discover relevant information from this data become increasingly important. Machine learning techniques are aimed at discovering knowledge by learning structural patterns from data (Mitchell, 1999). While machine

learning techniques are mostly used to identify hidden data patterns with the objective to support future decision making, relatively few studies apply machine learning to behavioral data with the objective to capture the intuition and experience embedded in expert decisions.

For example, Kim and Han (2003) state that while numerous studies have applied learning techniques to quantitative financial databases with the objective of predicting bankruptcy, actual risk assessment processes still require bankruptcy predictions by experts due to the value of their subjectivity. The authors propose a method to discovery bankruptcy decision rules from experts' qualitative decisions using a learning method based on a genetic algorithm. Similarly, Shaw and Gentry (1988) applied inductive learning to develop an expert system that mimics the thought process of a lending officer at a bank, as a first step towards the automation of the process of evaluating business loans. Studies capturing expert decisions or judgments cover a variety of industries and machine learning techniques, ranging from simple regression to model judgment of military conscripts by interviewers (Ganzach, Kluger & Klayman, 2000), to decision tree induction to capture coding decisions of teachers (Lin, Hsieh & Chuang, 2009), and to a more advanced random forest aimed to mimic expert assessment of the quality of medical scans (Menze, Kelm, Weber, Bachert & Hamprecht, 2008).

The limited number of studies applying machine learning to model expert decisions from behavioral data generally seem to have the captured expert knowledge as the ultimate goal of their efforts, mostly to automate decision making. Maghrebi, Sammut and Waller (2015) did a feasibility study of automating the process of determining the order of concrete deliveries. The authors employ a range of machine learning techniques to match expert decisions with the objective of decreasing dependency on human resources. Note that decision data is generated by developing a simulation model that presents decisions to an expert. Isaac and Sammut (2003) also state that experts rely on highly developed tacit skills, which they can often not explicitly describe. They propose machine learning tools for the acquisition of this knowledge, which they call 'behavioral clones'. They demonstrate the ability of a combination of a decision tree learner and linear regression to reproduce fly manoeuvres in an aircraft simulation. Lafond, Roberge-Vallières, Vachon and Tremblay (2017) compare the ability of three machine learning techniques in capturing human classification behavior using a simulated naval air defense task. Results show that decision trees are able to capture the concerned decision process best, which the authors expected since they are well suited to represent human cognition under time pressure. Applications of the captured decision process that are discussed are in the areas of training and decision support.

However, capturing expert decisions with the objective to support future decisions implicitly assumes that the captured expert knowledge is optimal, or at least neglects the fact that insight into current practices provides a good opportunity for the identification and evaluation of improvement of the decision making process. Lafond et al. (2013) propose applying a learning technique to functionally mirror expert mental models. Their objective is to improve decision quality by recognizing when a decision maker is deviating from his usual decision patterns, since this might indicate probable errors. However, this application still assumes the captured policy to be the correct, or desired one. The authors do acknowledge this limitation, but leave improving of the decision support model for future research. As described in Chapter 1, we will capture the way in which dispatch decisions are currently made by dispatch agents in Brabant-Zuidoost with the objective of using the captured current dispatch practices as a starting point to improve decision making, since it is expected that this contains valuable

domain-specific expert knowledge and insight into practical considerations.

Donnot, Guyon, Schoenauer, Panciatici and Marot (2017) recognize the limitations of directly applying learned expert decisions, which is why they propose a hybrid decision support system. First they apply a deep neural network to historic decision data to mimic human decisions in the prevention of violating power flow limits in a power plant, so-called remedial actions. Subsequently, however, their decision support system uses simple simulation to evaluate the effect of each action proposed by the captured decision model before suggesting it to the decision maker. While this approach does not actually improve on the captured decisions, it does distinguish between bad and good decisions and only uses the good ones to support future decision making. Furthermore, X. Li and Olafsson (2005) apply machine learning to production data to capture the way in which experts schedule jobs. One of the potential benefits of this approach listed by the authors is gaining structural knowledge that could lead to new rules to improve scheduling performance. However, the authors leave this to future research.

From the discussed studies that capture a decision making process it can be concluded that, to the best of our knowledge, there are no studies which have captured expert decisions with the objective to use the resulting model as a basis to improve upon. While some authors recognized that expert decisions are likely to contain valuable expertise and domain-knowledge but are not necessarily optimal and might be improved upon, no steps were taken to derive and apply insights concerning potential improvements from the captured policies. Secondly, most of the discussed studies did not derive decision policies from historic data, but rather generated this data by presenting experts with an artificial (simulated) task. However, we expect decisions derived from historic decision data to resemble actual decisions made in practice more closely, since in this case the experts were not aware that their decisions were being monitored, and an artificial task might not contain all variables influencing decision making that are present in practice.

Lastly, in capturing ambulance dispatch decisions, we will apply a post-processing phase which combines knowledge from both the domain and literature with the learned model to further improve the accuracy of the resulting model, as well as make it more concise. To the best of our knowledge, no other studies apply such a post-processing phase after capturing a decision policy using machine learning techniques.

2.7.2 Alternative dispatch policies

In ambulance management literature it is often assumed that to each request the closest idle ambulance is dispatched (Jagtenberg et al., 2017). Here, the closest idle ambulance concerns the ambulance that is currently not serving another request, i.e. with status 1 or 2 in Figure 2.2, which can reach the request location fastest. Furthermore, generally, ambulance regions are studied in isolation, without interaction with ambulances from neighbouring regions, which means that these ambulances are not considered in determining the closest idle ambulance to dispatch. Performance improvements are often sought through the identification of optimal relocation policies in isolation, without searching for an optimal dispatching policy or evaluating how the performance of such a relocation policy depends on the used dispatching policy. However, while the closest-idle policy is easy to implement and provides a quick decision, it is not necessarily the most beneficial policy for performance. This section will provide a brief

overview of the limited available literature on alternative dispatch policies and the extent to which its results were evaluated such that conclusions can be drawn regarding expected performance in practice. Refer to the complete structural literature review for a description of the used search, selection, and extraction procedure, as well as an overview of literature on relocation policies (Theeuwes, 2018).

Literature on dispatch policies in ambulance management can be subdivided in *offline* and *online* methods. While offline dispatch policies determine which ambulance to dispatch in each of a finite set of (simplified) system states in a preparatory phase, online dispatch policies concern real-time decisions based on the actual system state at the decision moment. In practice, offline dispatch policies are often applied in EMS systems which are not able to track the real-time locations of ambulances. In more modern countries/regions such as in the Netherlands, however, dispatch agents resort to online dispatch policies instead, which is why this section focuses on such online dispatch policies.

Jagtenberg et al. (2017) propose a dispatching heuristic, based on a well-known coverage location problem formulation called MEXCLP (Daskin, 1983). For each request, the heuristic considers all ambulances which can reach the request's location within a predetermined time limit. Of these ambulances the marginal coverage that each ambulance provides to the region is computed, after which the ambulance with the smallest marginal coverage is dispatched. This heuristic led to improvements of up to 18% in terms of the fraction of requests with a response time less that the predetermined time limit, but this came at a cost of an increase of 37% in the mean response time. Note, however, that this heuristic assumes all idle ambulances to reside at its base station, implying not only instantaneous movements between request locations and base stations, but also the lack of a more dynamic relocation policy. In practice, a relocation policy is generally in place, which is able to return ambulances to stations different from its base station after serving a request, as well as relocate idle ambulances between stations to improve coverage of the region. Such a relocation policy is expected to affect the benefit of taking into account marginal coverage in dispatch decisions to on-time performance. Yet this interaction is not addressed by the authors.

Similarly, Lee (2011) applies a measure of preparedness to dispatch the idle ambulance that maximizes the minimum preparedness over all demand zones at each decision moment. The author evaluates his dispatch policy in a highly simplified system, consisting of a square 5x5 grid with a deterministic travel time of one minute for each edge. Results shows that dispatching based on this preparedness function results in a significant increase of the mean response time compared to the closest-idle policy. However, no performance measures regarding the fraction of requests with a response time below a certain time threshold are evaluated.

Alternatively, Majzoubi, Bai and Heragu (2012) introduce the possibility to reroute ambulances transporting patients with a low urgency such that it can pick up one more patient, in an attempt to minimize travel costs, as well as a penalty for not meeting the given response time target. Since this model strongly focuses on transporting patients to the hospital, and the ability of an ambulance to transport multiple patients, it is more applicable to large-scale emergency situations rather than everyday ambulance logistics. However, the rerouting of ambulances is an interesting concept to evaluate in circumstances closer to Dutch EMS practices.

Furthermore, Lee (2014) suggests the 'Parallelism' policy, which considers not only idle, but

also busy ambulances in dispatching decisions. An assignment is made between ambulances and waiting requests based on the expected response time from each ambulance to each request, which may include the expected remaining service time to a currently served request if an ambulance is busy. Subsequently, only the idle ambulances are actually dispatched to the requests they were assigned to, while requests that were assigned to a busy ambulance remain in the queue, since a better assignment option may arise before the busy ambulance becomes idle. The author reasons from a situation in which there are generally multiple simultaneous requests to which an ambulance needs to dispatched, such as large-scale emergencies. Translating this concept to everyday Dutch EMS practices implies that this policy may result in no ambulance being dispatched because a currently busy ambulance is expected to be able to respond quicker. However, this is undesirable for highly urgent requests, since this decision is based on highly uncertain expected values, meaning that this might lead to a very long realized response time for this request. To prevent this risk, the policy might be adapted to always dispatch an idle ambulance to a (highly urgent) request even if the policy assigns a busy ambulance, since this ambulance can always be canceled if the busy ambulance indeed completes its current service such that it can reach the request's location quicker. This approach decreases dependency on the realization of highly variable expected values.

Lim, Mamat and Braunl (2011) propose two alternative dispatch policies, of which the first has some resemblance to that of Majzoubi et al. (2012), and the second to that of Lee (2014). Firstly, they propose to allow an ambulance, that is on its way to a request, to be rerouted to a request with a higher urgency, if this results in a smaller response time, which they call 'reroute-enabled dispatching'. Interestingly, dispatch proposals in the BZO region already provide this option to dispatch agents. Furthermore, Lim et al. (2011) considers reassignment of a highly urgent request to a different ambulance that has just completed serving a request, if this improves its response time, which they call 'free ambulance exploitation'. Both alternative dispatch policies were evaluated individually, as well as combined, in a hypothetical EMS region consisting of a 16x16 grid, in which ambulance travel speed, treatment time, and transfer time at the hospital are all assumed to be deterministic and static. Resulting performance is evaluated in terms of the mean response time to requests of each urgency level. It was found that, in terms of the mean response time to highly urgent requests, both dispatching policies are beneficial, with 'reroute-enabled dispatching' slightly outperforming 'free ambulance exploitation'. The 'free ambulance exploitation' policy is applied even if the expected travel time of the recently freed ambulance is only marginally shorter than the remaining travel time of the ambulance that was initially dispatched. This leads to quite frequent cancellations of these initially dispatched ambulances, as well as a significant increase in the mean response time of requests of a lower urgency, both of which might not be desirable. One might consider limiting both by only considering exploitation of a free ambulance if the resulting response time gain is significant, or even only if it makes a difference in an ambulance arriving on-time or not. Extensive experiments in a realistic (simulation) environment are necessary to evaluate the effect of these adaptions.

Lastly, there are a number of authors studying dispatch policies jointly with relocation policies. Andersson and Värbrand (2007) introduce the preparedness measure as later used by Lee (2011) for both relocation and dispatch decisions. The authors only deviate from the closest-idle policy for less urgent requests by dispatching the ambulance resulting in the highest minimum preparedness over all demand zones. Resulting performance of this dispatch policy, combined with their relocation heuristic, is evaluated in a simulation. However, many unreal-

istic assumptions were made, including static input variables and instant relocations, meaning that the travel time for relocations is set to zero. Furthermore, no benchmark is introduced to compare performance of the proposed policies with. Similarly, for each less urgent request Gendreau, Laporte and Semet (2001) solve their relocation problem and dispatch the ambulance that results in the best coverage after these relocations. Furthermore, these authors also allow for an ambulance on its way to a less urgent request to be redispatched to a highly urgent request under certain conditions. However, due to the extensive relocation problem that needs to be solved multiple times for each dispatch decision to a less urgent request, infeasibilities might arise in case of two requests arriving in quick succession. Schmid (2012) and Nasrollahzadeh, Khademi and Mayorga (2018) apply approximate dynamic programming to find alternative dispatch policies. Unfortunately their results do not show insight into the resulting policies.

In conclusion, research into dispatch policies different from the commonly assumed closest-idle policy is quite limited. Some interesting concepts have been introduced, such as taking into account the coverage reduction resulting from each dispatching option, allowing ambulances to be dispatched while it is on its way to a less urgent request, and the exploitation of recently freed ambulances. Essentially, the first of these diverts focus from optimizing the response time of the current request to optimizing the response time of requests expected to arrive in the near future. The second explores the extension of the closest-idle policy with a larger set of ambulances to consider, i.e. not only free ambulances. The third concept challenges the, seemingly, fixed decision moments at which dispatch decisions are made, i.e. not only at the arrival of a new request, but also at the appearance of alternative dispatch options such as recently freed ambulances.

While all three of these concepts are interesting alternatives to the commonly assumed closest-idle policy, their potential has not yet been evaluated in realistic(ally sized) problems. The mentioned existing studies generally evaluate the proposed alternative dispatch policies through a simulation in which many simplifying modelling choices and assumptions are made, mainly relating to the size of the problem and the dynamicity of request arrivals and characteristics. For example, Lee (2011, 2014) simulates a hypothetical square grid of 25 vertices with a fixed driving time for all edges. They do not distinguish between urgency levels and assume a general distribution for treatment and transfer times, a static number of ambulances, and ambulances remaining idle at a request location after service completion. Lim et al. (2011) simulate a larger-sized system, but a hypothetical square grid nevertheless. They assume ambulance stations to be evenly spread, which includes the unrealistic positioning of stations near the borders of the hypothetical region. While the authors do distinguish between two types of urgency levels, they assume constant driving times, a fixed spatial distribution of requests, a static and deterministic treatment time of ten minutes for each request, as well as each request requiring transport to the (same) hospital with a fixed transfer time. Jagtenberg et al. (2017) do simulate the actual EMS region of Utrecht, but assume a static relocation policy causing ambulances to always return to its base station after service completion. Furthermore, the authors assume static request arrivals, as well as static ambulance capacity, treatment and transfer times. Additionally, they assume that in case of hospitalization, the nearest hospital is always selected. Before conclusions can be drawn regarding the expected performance benefits of alternative dispatch policies in practice, extensive experiments in a realistic environment are necessary to evaluate the effect of these dispatch policy adaptions.

3 | Formalization of dispatch process

Capturing current dispatch practices allows us to improve this process by building upon it, in which the use of the current routine as a foundation is likely to foster practical relevance and adoption among dispatch agents. Furthermore, the current dispatch routine can be used as a realistic benchmark in the evaluation of these potential improvements in a simulation.

Particularly, the objective of this chapter is to determine which ambulance is dispatched to a request, given the corresponding dispatch proposal, and why a dispatch agent might decide to deviate from dispatching the closest idle ambulance. Here, we implicitly assume that, for any dispatch decision to be made, a dispatch proposal is generated and one of the ambulances in the proposal is dispatched. Section 3.1.5 elaborates on the robustness of this assumption.

Following the framework of applying learning algorithms to production data by X. Li and Olafsson (2005), this formalization effort consists of two phases:

- Preprocessing of data, including aggregation and feature construction (Section 3.1)
- Model induction and interpretation (Section 3.2)

Next, Section 3.3 elaborates on the results and insights gained from this effort and applies a unique post-processing phase, after which in Section 3.4 four potential enhancements to the dispatch policy are proposed.

3.1 Preprocessing

To be able to capture how a dispatch agent decides which ambulance to dispatch using a dispatch proposal, all information available to the agent at the decision moment, which might affect the decision, needs to be taken into account. As in most cases of knowledge discovery, this data is not readily available and considerable transformations of the available data are required. The desired output of this phase is a set of *instances*. Here, an instance refers to an independent example which can be learned from, i.e. a dispatch decision that was made in the past. In the resulting instance file, which is used as input to the induction effort, each row corresponds to an instance and each column represents a piece of information that might have affected the decision that was made, a *feature*. Additionally, the instance file includes a column which specifies the resulting class for each instance, i.e. the decision that was made.

Based on our observations, we distinguish three categories of information that are available to the dispatch agent and might influence a dispatch decision. The instance file to be constructed needs to include features capturing these categories, which are related to:

- The request to which an ambulance needs to be dispatched
- The dispatch proposal corresponding to the request
- Coverage of the region, i.e. preparedness to respond to future demand

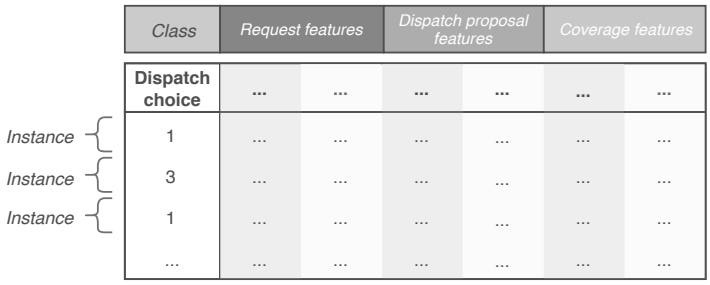

Figure 3.1: Illustration of the instance set format

Figure 3.1 illustrates the instance set format which is the required output of this preprocessing phase. Several sources of raw data are available in various formats, which need to be aggregated, after which relevant features need to be constructed from the aggregated data. As we introduce features throughout this chapter, an example instance set will be filled with possible values for illustrative purposes.

3.1.1 Available data

Four main data logs were used as raw data from which the instance file is constructed. Figure 3.2 displays these four logs and the relation between them using a UML database model. Firstly, a log of requests, including the ambulance that was dispatched, is available. Secondly, all dispatch proposals that were generated upon request by a dispatch agent are saved as a text file containing fifteen ambulances, ranked according to the underlying algorithm (see Section 2.3). Thirdly, a change log of the status of each ambulance is available, from which the status of an ambulance at any time can be derived. Lastly, a similar change log is available of the position of each ambulance; the coordinates of each ambulance are logged every hundred meters if it is moving or every five minutes if the ambulance is standing still.

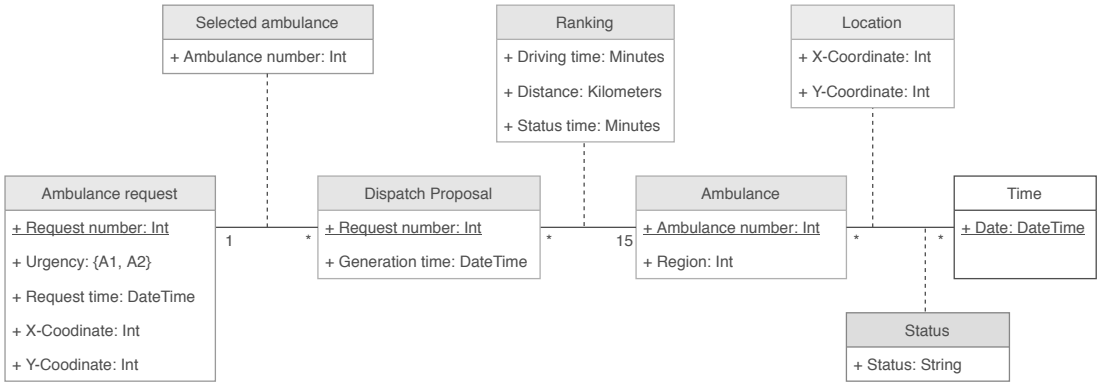

Figure 3.2: Overview of available GMS data: Request data (blue), generated dispatch proposals (green), status logs (red), and location logs (orange)

Data was obtained for the months September and October 2018 from GMS. All data logs include entries concerning ambulances from the BZO region, as well as from EMS region

Brabant-Noord (BNO). The dispatch processes for these two regions are separated, but located in the same building, meaning that data logs are shared. Together, these regions are called Brabant-Oost (BO). Data on ambulances from other regions than BO is only included in the above-mentioned logs if they are positioned in the BO regions, plus a radius of ten kilometers. Only data on requests (dispatches) in the BZO region is used, but data on external ambulances is relevant since they might be dispatched to these incidents as well.

3.1.2 Instance class

For each instance, or dispatch decision, the resulting class needs to be determined. By matching the ambulance that was dispatched to the corresponding dispatch proposal, it can be deducted which option was selected by the dispatch agent. This results in fifteen possible classes, one to fifteen. However, the number of samples for each of these classes are strongly unbalanced, due to the fact that ambulances in the dispatch proposal are ordered based on their driving time to the incident and the main performance measure strongly depending on this driving time. To ensure a sufficient number of samples of each class to be available, such that the inducted decision tree is trained on all possible classes, classes five to fifteen are combined to form one class, which we call '5+'. We are especially interested in an agent's reasons for deviating from sending the closest idle ambulance, which are expected to become apparent by distinguishing between the first few options of a dispatch proposal and which further justifies this decision to combine classes five to fifteen. The resulting class distribution in the final instance set, after instance selection (see Section 3.1.5) is shown in Figure 3.3.

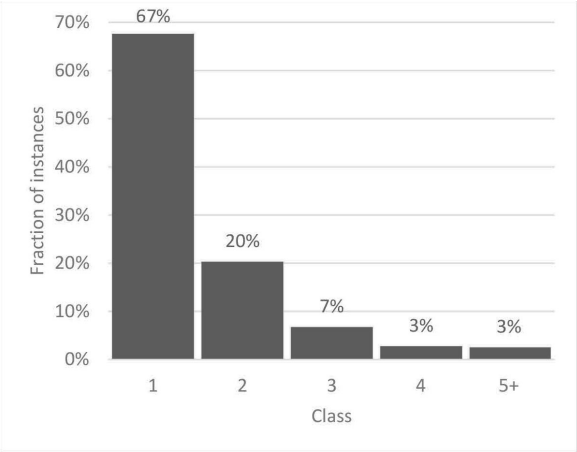

Figure 3.3: Class distribution in the final instance set

3.1.3 Basic instance features

As described above, the features of each instance should reflect the information that was available to the agent at the moment the dispatch decision was made. Since dispatch agents are dedicated to making dispatch decisions, which happens under time pressure, we can assume that all information presented to a dispatch agent is considered to be relevant to such a decision. Capturing all these pieces of information in a set of features requires the process of *feature engineering*, which entails the use of domain knowledge in transforming the available

data into relevant features. In this section the features deducted from the available data are listed and elaborated upon. We distinguish between features relating to the request to which an ambulance needs to be dispatched, features related to the (options in) the dispatch proposal, and features capturing the coverage of the region, which reflect the ability to respond adequately to future demand.

Request features
At the moment a dispatch decision needs to be made generally not all information regarding the request is known yet. An answering agent typically forwards the request for an ambulance to the dispatch agent as soon as its location and urgency are known. This ensures an ambulance can be dispatched as soon as possible, while further information regarding the patient's condition may become evident after dispatch. Therefore, only limited information regarding the request is available at the moment a dispatch decision needs to be made.

- **Urgency**: Naturally, the urgency of the request, i.e. 'A1' or 'A2', to which an ambulance needs to be dispatched is relevant to the dispatch decision, since response time targets depend on this characteristic. This information is supplied to the dispatch agent jointly with the dispatch request. For each historic instance, we obtain the urgency of the corresponding request from the request log.
- **Passed time**: The arrival of the call for emergency assistance marks the start of the response time of the concerned request, which determines the main performance measure. Therefore, at the moment a dispatch decision is made the time that has passed since the call arrived is relevant, since this affects the time left until the response time target. For each instance, the passed time (in minutes) can be obtained by subtracting the time the call arrived from the time the dispatch proposal was generated.

Consider Figure 3.4 for an illustrative, partial, instance set consisting of four instances. Both example values of the class (i.e. dispatch choice) and the request-related features (i.e. urgency and passed time) are provided. As we add features, this example set will be updated.

Class	*Request features*	
Dispatch choice	**Urgency**	**Passed time**
1	A1	2
3	A2	1
1	A1	2
5+	A2	3

Figure 3.4: Example instance set including four instances, the class and request-related features

Dispatch proposal features
For each of the fifteen ambulances in a dispatch proposal, several pieces of information are listed (see Section 2.3). To represent these pieces of information, the following features should be included in the instance set, with i referring to the ith option in a dispatch proposal, $i \in \{1, 2, 3, 4, 5\}$. Only features referring to the first five options are relevant to the dispatch decision, since instances to which one of the options five through fifteen is dispatched are

combined into one class, 5+. This modelling choice also reduces the dimensionality of the instance set, thereby decreasing the risk of overfitting.

- **Driving time of** i: The driving time of each option to the request location is the main characteristic that determines the ranking of options in the dispatch proposal, meaning that the driving time of i is smaller than the driving time of j if $i < j$ for any given instance. The driving time of each option is logged in each dispatch proposal, meaning that it can easily be obtained.

- **Status of** i: While the dispatch proposal algorithm excludes ambulances that are not available for dispatch to the incident at hand (see Section 2.3), the status of the provided options may still vary. Ambulances may be idle at a station, driving towards a station, transferring a patient at a hospital or even currently serving a less urgent request. While the status of each option of the dispatch proposal is displayed to the agent upon its generation, these are not explicitly logged for each proposal. However, the status of each option can be deducted by matching the time the proposal was generated to the most recent status change using the status change log of the concerned ambulance.

- **Idle status indicator of** i: Some statuses might be regarded as equal in dispatch considerations. For example, both ambulances that are free on the road and those free at a station are considered to be idle, i.e. directly available for dispatch. Alternatively, an ambulance that is transferring a patient at a hospital, or on its way to a less urgent request, might require some additional time before it can start driving towards a newly assigned request. Therefore, a binary feature indicating whether an ambulance is directly available is included.

- **Status time of** i: Similarly, the time since the status of each option last changed is displayed with the dispatch proposal during the dispatch process. Furthermore, ties in ranking the available ambulances based on their driving time and distance to the incident are broken based on this characteristic, meaning that this value for each option can be deducted from the logged dispatch proposals.

- **Remaining shift time of** i: A dispatch agent has insight in the time the current shift of each ambulance ends. Shifts typically last eight hours and need to start and end at the base station of the concerned ambulance. This implies that a dispatch agent might prefer a lower ranked option if an ambulance's shift is almost ending and the request to which an ambulance needs to be dispatched is located far from its base station. An ambulance is eligible to be listed in a dispatch proposal as long as its status is not set to 'off duty' (Dutch: Buiten Dienst (bd)). This means that an ambulance which' shift has ended but has not yet arrived at its base station can still be dispatched, and will thus be driving in overtime. Besides overtime, a shift may also start early if the crew is already present. Therefore, the end of shift, or remaining shift time, of each option can be determined by identifying the shift of the concerned ambulance closest to the time the dispatch proposal was generated. This may be a shift in the past, meaning that the ambulance is driving in overtime and thus that the remaining shift time is negative, or a shift that has (officially) not started yet, meaning that the remaining shift time exceeds eight hours. Unfortunately, shifts itself are also not explicitly logged, meaning that they have to be deducted from the status change logs. Shifts can only be derived for ambulances of the BO regions (BZO and BNO) since only all their status changes are

available. Section 3.1.6 elaborates on the manner in which missing values are handled.

- **Own ambulance indicator of** i: The region each ambulance belongs to can be deducted from its ID number, e.g. BZO ambulance IDs start with '22'. Whether an ambulance is a BZO ambulance or not might be relevant to a dispatch decision since BZO dispatch agents can directly dispatch their own ambulances, while dispatching an ambulance of another region requires requesting permission by telephone. Such a permission request takes time and might be denied because of limited availability of idle ambulances in the concerned region. Furthermore, after dispatching an ambulance belonging to another region, the dispatch agent does not have any control over, or insight in the status of, this ambulance anymore. Therefore, a binary feature is included indicating whether an ambulance belongs to the own region (BZO) or not (other regions).

- **Region BZO & BNO (BO) indicator of** i: Since the dispatch agents of region BZO and BNO are located in the same room, some of the disadvantages regarding delay and communication might not be present when dispatching an ambulance of this region. Therefore, another region indicator feature is included which indicates whether a dispatch option belongs to either of the BO regions, or not.

Refer to Figure 3.5 for the example instance set to which the above-mentioned features related to the dispatch proposal have been added. For clarity and space purposes no values are shown for the formerly added request-related features, as well as for the dispatch proposal-related features for dispatch options other than the first.

Class	*Request features*	*Dispatch proposal features*							
		Option 1							*Options 2-5*
Dispatch choice	**...**	**Driving time #1**	**Status #1**	**Idle #1**	**Status time #1**	**Rem. shift #1**	**Own ambu #1**	**BO ambu #1**	**...**
1	...	7	vr	1	8	9	0	1	...
3	...	8	ab	0	7	8	0	0	...
1	...	11	op	1	31	14	1	1	...
5+	...	5	ar	0	4	6	1	1	...

Figure 3.5: Example instance set including four instances, the class, and features relating to the request and dispatch proposal (cont. example)

Coverage features

Lastly, besides information regarding the request and each option listed in the dispatch proposal, the dispatch agent has a screen with a map of the region at his disposal, on which all ambulances are displayed at its current location, in a colour indicating its status. This allows him to get a quick impression of the extent to which the region is prepared for future requests. The choice of which ambulance to dispatch is a decision that needs to be made quickly, meaning that the dispatch agent can only regard the map momentarily to get an impression of the region's preparedness. Therefore, simple measures are defined as a proxy for conclusions a dispatch agent may draw by looking at the map of the region. The first two of these relate to the overall preparedness of the region at the moment the dispatch proposal is generated.

- **Number of available ambulances**: The current available capacity of the region might affect decision making, since this is an indication of the preparedness of the region for ambulance request arriving in the near future. The number of idle vehicles is determined using the status change log, by counting the number of ambulances that are on duty and are either free on the road or at a station. Additionally, based on discussions with BZO's dispatch agents, we add the number of ambulances that are on duty and are transferring a patient at a hospital. These ambulances are expected to become idle in the very near future, and are therefore also considered to contribute to the preparedness of the region. They may even be requested to accelerate the transfer process if they are needed for dispatch to a very urgent incident.
- **Single coverage**: Besides the number of available vehicles, their location is also visible on the map, as well as relevant. If all available vehicles are located at the same location, a large part of the region can still not be reached within a reasonable time span. Therefore, we introduce the measure of *single coverage*, which refers to the fraction of the region that is covered, i.e. can be reached within twelve minutes of driving time by at least one ambulance (Bélanger et al., 2018). Here, we select twelve minutes of driving time as the coverage criterion, since the target response time for the most urgent calls corresponds to fifteen minutes of which generally three minutes are reserved for answering and dispatching (Jagtenberg et al., 2017; Van Barneveld, 2016). We divide the region in areas corresponding to the 4-digit postal code areas (e.g. 5613) and use driving times between the centroids of these postal code areas. These driving times were obtained through the National Institute for Public Health and Environment (Dutch: RIVM) and are based upon realized driving speeds of ambulances to highly urgent (A1) calls. The driving times corresponding to rush hour are used such that the resulting coverage reflects the most conservative view. The exact location of each available ambulance is derived from the position log and mapped onto a postal code area based on the closest centroid (distance as the crow flies between relative coordinates using Pythagoras), after which all postal code areas that can be reached within twelve minutes are marked as 'covered'. The fraction of postal code areas that is covered by at least one ambulance constitutes the single coverage feature.

Based on conversations with dispatch agents, only ambulances of the BZO region are considered when determining the number of available ambulances and the single coverage of the region at a given moment, since dispatch agents do not have control over ambulances of other regions, even if they are currently located in (proximity of) the BZO region. These ambulances may be relocated elsewhere by other dispatch agents at any moment.

Additionally, two features are introduced for each option i in the dispatch proposal relating to the reduction in preparedness that would be caused by dispatching that particular ambulance. It might for example be the case that dispatching a certain ambulance leads to a large part of the region being left uncovered, while another option is currently only covering areas which remain covered by another ambulance after dispatch. Both measures are based on the notion of single coverage.

- **Absolute coverage reduction of** i: The effect on preparedness for future demand resulting from dispatching a certain ambulance can be expressed in terms of the reduction of single coverage. That is, for each dispatch option the single coverage of the region is determined excluding that particular ambulance. Subsequently, the difference between

the current single coverage and the single coverage excluding the dispatch option is computed. This measure reflects the absolute reduction of single coverage resulting from dispatching a particular ambulance.

- **Percentual coverage reduction of** i: Similarly, the percentual coverage reduction is computed for each dispatch option by dividing the absolute coverage reduction of option i by the general single coverage before dispatch.

Figure 3.6 provides a simple demonstration of these four coverage related features. Here, a small example region is split into twenty areas. There are currently three available vehicles and for each of them it is indicated which areas they cover, i.e. which areas they can reach within twelve minutes. It can be seen that two areas are covered by both ambulance two and three, but this does not affect the (single) coverage measure. For each ambulance the coverage resulting from dispatching it is shown. Furthermore, the coverage reduction relative to the situation before dispatch is computed, both in terms of absolute and percentual reduction.

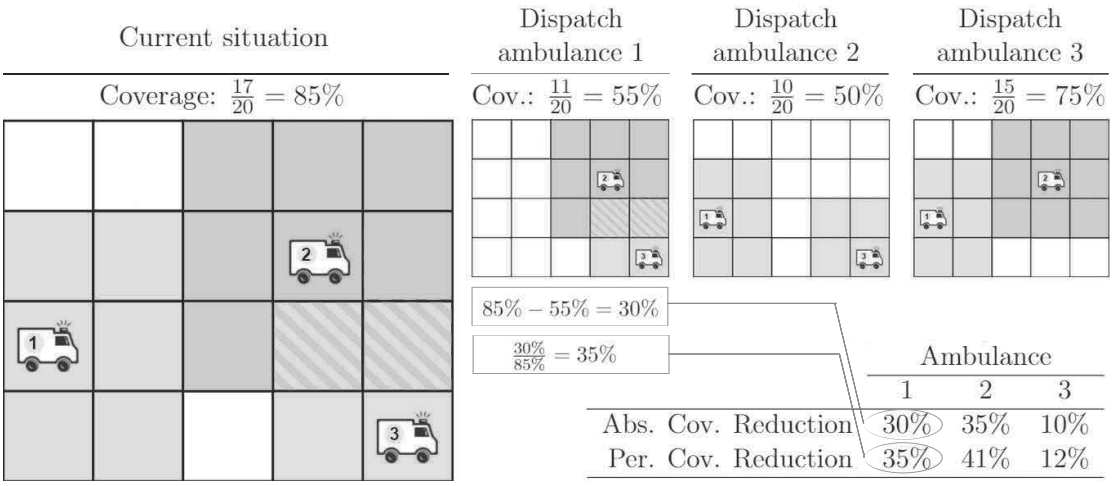

Figure 3.6: Simple coverage demonstration with three available ambulances

Refer to Figure 3.7 for the example instance set to which the above-mentioned features related to coverage have been added. Note that besides the general coverage features, only values of those relating to the first dispatch option are shown for clarity purposes.

Class	*Request features*	*DP features*	*Coverage features*				
			General		*Option 1*		*Options 2-5*
Dispatch choice	**...**	**...**	**Nr. idle**	**Coverage**	**% cov. red. #1**	**Abs. cov. red #1**	**...**
1	...	...	4	58%	0%	0%	...
3	...	...	6	72%	0%	0%	...
1	...	...	7	87%	5.1%	4.4%	...
5+	...	...	3	49%	0%	0%	...

Figure 3.7: Example instance set including four instances, the class, and features relating to the incident, dispatch proposal, and coverage (cont. example)

Table 3.1 provides an overview of all features introduced in this section. Including dispatch option-specific features for options one through five leads to a total of 49 features for each instance. For each feature the data type is indicated according to the classification of Han, Pei and Kamber (2011), as well as the data sources used to construct these features.

Table 3.1: Overview of basic features with $i \in \{1, 2, 3, 4, 5\}$ being dispatch proposal options

No.	Feature	Symbol	Data type	Source			
				Request log	Dispatch proposal	Status log	Position log
1	Urgency	U	Ordinal: {A1, A2}	x			
2	Passed time	P	Numeric (Min.)	x	x		
3-7	Driving time of i	D_i	Numeric (Min.)		x		
8-12	Status of i	S_i	Nominal: {1,2,3,6}		x	x	
13-17	Idle status indicator of i	SI_i	Binary		x	x	
18-22	Status time of i	ST_i	Numeric (Min.)		x		
23-27	Remaining shift time of i	RS_i	Numeric (Min.)		x	x	
28-32	Own ambulance indicator of i	$Rown_i$	Binary		x		
33-37	Region BZO & BNO indicator of i	Rbo_i	Binary		x		
38	Number of idle ambulances	I	Numeric			x	
39	Single coverage	Cov	Numeric (%)			x	x
40-44	Percentual coverage reduction of i	PCR_i	Numeric (%)		x	x	x
45-49	Absolute coverage reduction of i	ACR_i	Numeric (%)		x	x	x

3.1.4 Composite instance features

Now that features representing the information available to a dispatch agent when making a dispatch decision are constructed by aggregating multiple data sources, we consider the construction of composite features. Decision tree induction, the technique that will be used for knowledge discovery, is not able to combine feature values (e.g. the driving time of dispatch option one and that of option two) and draw conclusions from the relation (e.g. the difference) between them. A dispatch agent, however, might infer information relevant to a dispatch decision based on the relation between two feature values. Feature construction is the process of inferring or creating additional features to discover missing information about the relationships between features (Liu & Motoda, 1998).

Therefore, new features are constructed, representing meaningful relations between the existing, basic, features. Generally, such composite features are constructed by performing a logical operation on basic features. Naturally, it only makes sense to include resulting features that are meaningful, e.g. while the difference between the driving time of option i and the driving time of option $j, i \neq j$ makes sense and has a intuitive unit (minutes), adding the urgency of the request to the coverage reduction of any option i does not have any meaning.

Considering the possible combinations between any multiple number of basic features results in the following meaningful composite features:

- **Difference between driving times of options i and $i + 1$, $i \in \{1, 2, 3, 4\}$**: Upon making a dispatch decision there may be reasons for an agent to deviate from dispatching a closer ambulance over an ambulance that is further from the request. However, these reasons might become irrelevant if the difference in driving time between this option and the subsequent one is too large. For example, if the shift of ambulance A is almost over, a dispatch agent might be inclined to consider dispatching another ambulance,

B, despite its driving time to the incident being longer. However, if the difference in driving time between these two options is too large, the resulting increase in expected response time (and thus increased risk of exceeding the response time threshold) might not outweigh the benefit of preventing overtime of ambulance A.

- **Difference between coverage reduction of options i and $i+1$, $i \in \{1,2,3,4\}$:** Similarly, the difference between the coverage reduction of two subsequent options might be relevant to a dispatch decision. While a large reduction in coverage as a result of dispatching a certain ambulance might lead an agent to reject this option, this argument becomes invalid if dispatching any of the other considered options lead to a comparable coverage reduction. This composite feature is included for both the percentual, as well as for the absolute coverage reduction measure.
- **Expected response time of option i:** As discussed in Section 2.4, the fraction of requests with a response time less than the threshold corresponding to its urgency is the main performance measure in the Dutch EMS system. Figure 2.2 shows that the response time comprises the triage procedure, the dispatch process, chute time, and the driving time to the request. Therefore, a composite feature referring to the expected response time of each dispatch option i is included, which is made up of the basic feature *Passed Time*, chute time, and the driving time to the request of i. Historic data shows that the mean chute time is 47 seconds. However, since driving times in the dispatch proposal are rounded to complete minutes, we set the expected chute time to one minute.

Refer to Figure 3.8 for the example instance set to which the above-mentioned composite features have been added. Again, note that only values of those features relating to the first dispatch option are shown for clarity purposes.

Class	*Request features*	*DP features*	*Coverage features*	*Composite features*				
				Option 1				*Options 2-5*
Dispatch choice	**...**	**...**	**...**	**Δdriving t. #1&#2**	**Δ% cov. red. #1&#2**	**Δabs. cov. red. #1&#2**	**Exp. RT #1**	**...**
1	...	...	...	1	0%	0%	10	...
3	...	...	...	3	-1.8%	-2.4%	10	...
1	...	...	...	2	-2.5%	-3.2%	14	...
5+	...	...	...	3	0%	0%	9	...

Figure 3.8: Example instance set including four instances, the class, all basic features, and the composite features (cont. example)

Table 3.2 shows an overview of the composite features that were constructed in addition to the basic features of Table 3.1, leading to a total of 66 features for each instance.

Table 3.2: Overview of composite features with $i \in \{1,2,3,4,5\}$ being dispatch proposal options

No.	Feature	Symbol	Data type	Operation	
50-53	Driving time difference*	ΔD_i	Numeric (Min.)	$D_{i+1} - D_i$	$i \in \{1,2,3,4\}$
54-57	Perc. coverage reduction difference*	ΔPCR_i	Numeric (%)	$PCR_{i+1} - PCR_i$	$i \in \{1,2,3,4\}$
58-61	Abs. coverage reduction difference*	ΔACR_i	Numeric (%)	$ACR_{i+1} - ACR_i$	$i \in \{1,2,3,4\}$
62-66	Expected response time of i	E_i	Numeric (Min.)	$P + 1 + D_i$	$\forall i$

*between subsequent dispatch options

3.1.5 Instance selection

Using the data described in Section 3.1.1, an initial instance set is constructed with each instance referring to a dispatch decision that was made for an A1 or A2 incident within the region of Brabant-Zuidoost. The class of each instance reflects the dispatch decision that was made according to Section 3.1.2. Furthermore, each instance contains a value for each basic feature and for each composite feature as described in Sections 3.1.3 and 3.1.4.

This results in a total of 7547 instances. However, not all instances are used in the formalization procedure for a variety of reasons. First of all, in selecting instances to be included in the final instance set, we consider the official measurement plans of the National Institute for Public Health and Environment (RIVM, 2010). At the end of each year the performance of each EMS region is computed and published by this institute according to these plans. Since this measure is what each EMS region is evaluated upon, we apply the same criteria in selecting dispatch instances as done in these plans. This implies the following exclusions:

- Dispatches of non-ALS or BLS vehicles (specialized vehicles, e.g. helicopter)
- Subsequent dispatches to a multi-dispatch incident (only the first counts)
- Patient transfers between hospitals, cancelled, false, and stand-by dispatches

Applying these filters prescribed by the RIVM's measurement plans results in a reduced number of 5931 instances. From the remaining instances it can be observed that some dispatch decisions were made without the generation of a dispatch proposal and that to some incidents an ambulance was dispatched which does not appear in the corresponding dispatch proposal. Specifically, it was found that two out of twenty regular dispatch agents (99+ dispatches in September and October 2018) structurally refrain from requesting a dispatch proposal before making a dispatch decision, while they are obliged to do so. Furthermore, in case an ambulance was dispatched which did not appear in the corresponding dispatch proposal it is assumed that the dispatch agent had information at his disposal which was not captured by the system. These observations lead to the following additional exclusion criteria:

- Dispatch decisions that were made without the use of a dispatch proposal
- Dispatches of vehicles that do not appear in the corresponding dispatch proposal

The above-mentioned exclusion criteria lead to 4506 instances.

3.1.6 Missing values

Before decision tree induction methods can be applied to the final instance set, it should be decided how missing values are handled. In our case, the reason for missing feature values is the fact that some data is only available for ambulances of the own region (or controlled from the own dispatch center, i.e. of region BNO). We adopt the simple technique of using a measure of central tendency for the missing feature values as described in Han et al. (2011), such that it is unlikely for extreme decisions to be made based on these artificial data values:

- Replacing the missing status time (ST_i) of ambulances from regions other than BZO or BNO by the mean status time (4.4% of feature values)

- Replacing the remaining shift time (RS_i) of ambulances from regions other than BZO or BNO by the mean remaining shift time (12.6% of feature values)

It should be noted that the status time of ambulances from regions other than BZO and BNO (BO) are currently logged as '999' if its status last changed outside of the BO regions (plus a buffer of ten kilometers). However, some ambulances from BO also have a status time of 999, since this is the largest possible value. Therefore, only those status times with value 999 of ambulances from regions other than the BO regions should be replaced. Since status changes of vehicles from regions other than the BO regions are only logged whenever they are in proximity of the BZO or BNO region, the remaining shift time of these ambulances cannot be estimated accurately from the status change log.

3.1.7 Summary statistics after preprocessing

This section provides some insight in the final instance set. Figure 3.9 shows the distribution of the (expected) driving time of each of the five options included in each instance. Naturally, the mean driving time increases with each option. Furthermore Figure 3.10 shows the distribution of the status of the ambulances that were selected to be dispatched. This figure shows that most dispatched are either free on the road or at a station. Less than thee percent of the dispatched ambulances are cases of redispatches, i.e. dispatches of ambulances which were on their way to a less urgent incident.

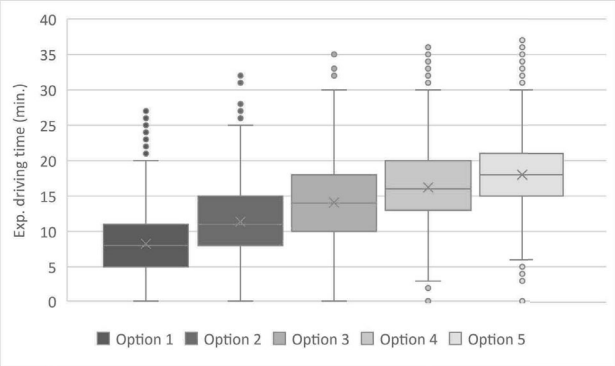

Figure 3.9: Driving time of each dispatch option

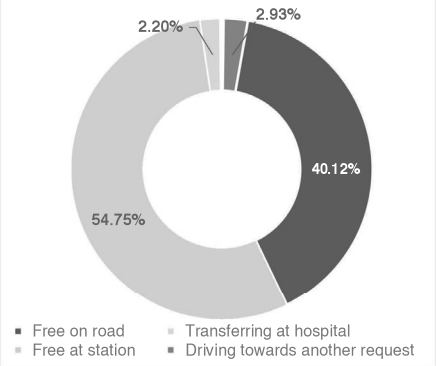

Figure 3.10: Status of dispatched vehicles

Furthermore, a correlation analysis is conducted to identify those features which correlate most with the class to be predicted, i.e. the dispatch decision. All categorical (ordinal and nominal) features are transformed to numeric features by creating binary dummy features, one for each possible value. The seven features with the largest absolute correlation with the objective class, i.e. dispatch decision, are shown in Figure 3.11. Large correlation to the objective class is a likely indicator for feature importance. It can be seen that the highest ranking features are those of dispatch option one, especially those relating to its status and region. Furthermore, the driving time of low ranking options, four and five, are present in the top ten, which can be explained by noting that these driving times are good indications of the driving times of all higher ranking options due to the fact that options are ranked based on their driving time.

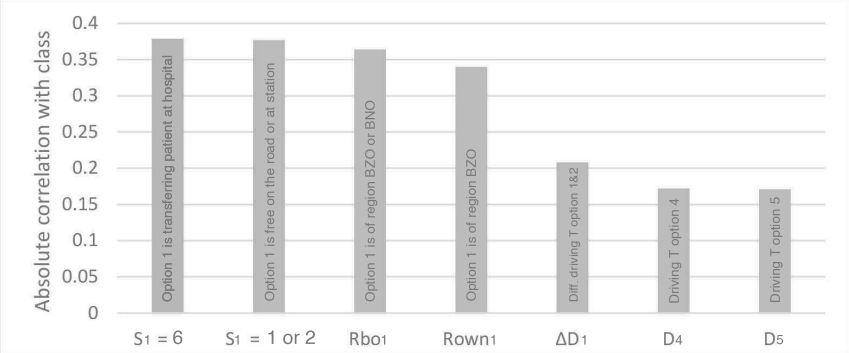

Figure 3.11: Top seven feature correlations with objective class; green and red indicate positive and negative correlations respectively

3.2 Decision tree induction

After the instance set has been constructed, a learning algorithm is applied to induct a decision tree that approximates the current dispatch process. A decision tree is selected to represent the current dispatch process due to its transparent and intuitive nature. While well-known machine learning representations such as neural networks are often described as black boxes, decision trees are easily interpretable (Kotsiantis, Zaharakis & Pintelas, 2007). Liu, Gegov and Cocea (2017) emphasize that 'black box' approaches should be used for predictive modelling to create a mapping from inputs to outputs, while 'white box' approaches are relevant in knowledge discovery, such that the underlying reasons for the mapping can be interpreted. The possibility to interpret the resulting knowledge representation allows us to gain insight into the current dispatch routine, which can be leveraged both as a foundation to build upon to ensure practical relevance of an improved dispatch process, as well as a benchmark in the evaluation of these potential improvements.

A decision tree consists of a root node, internal nodes, branches, and leaf nodes, see Figure 3.12. The decision process starts at the root node. Both at the root node and at each internal node a specific feature of an instance is tested, after which the instance is routed down the tree along the branch corresponding to the test's outcome. A leaf node represents a probability distribution over a set of classes, such that when an instance reaches a leaf node, the class corresponding to that instance can be predicted according to the specified probabilities. Tested features can be both numeric and categorical (Witten, Frank, Hall & Pal, 2016).

This section describes the approach that was used to induct a decision tree from the constructed instance set. Subsection 3.2.1 describes the selected learning algorithm, including the tuning of parameters of this algorithm. Next, Subsection 3.2.2 elaborates on the performance measures used to tune parameters of the learning algorithm and evaluate its outcome. Subsection 3.2.3 goes into the imbalancedness of the instance set at hand.

3.2.1 Experimental setup

For the decision tree induction effort an implementation of the well-known CART (Classification and Regression Trees) algorithm (Breiman, Friedman, Olshen & Stone, 1984) in Python

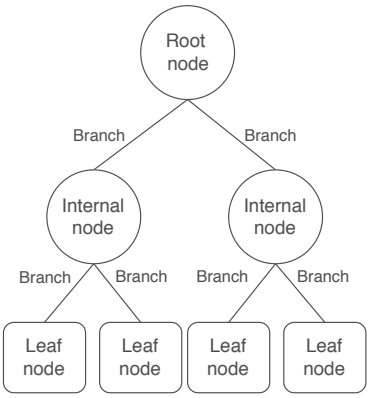

Figure 3.12: Schematic example of a decision tree

is used, called *Scikit-learn* (Pedregosa et al., 2011). The CART algorithm adopts a greedy approach, in which a decision tree is constructed according to a top-down, recursive divide-and-conquer method. The basic idea of this approach consists of recursively selecting a feature to place at a node and constructing one branch for each possible option. Subsequently a new node is created at the end of each of the constructed branches, for which the procedure is repeated. Note that at the end of each branch only a subset of the training set is relevant, namely the subset of instances actually reaching the concerned branch based on its attribute values, which is why this approach is said to divide-and-conquer (Witten et al., 2016).

At each node, the selection of a feature, based on which the subset reaching that node is split, consists of finding the feature that minimizes the resulting *impurity* of each resulting partition. The two most common impurity measures are *Entropy*, which selects the feature leading to the highest information gain, i.e. the most homogeneous node, and the *Gini Index*, which selects the feature that minimizes the probability of a weighted guess being incorrect.

Before inducting a decision tree from an instance set, this set needs to be split into a training and a test (or validation) set. Evaluation of the tree's performance using the same data set as it was trained on will lead to an overestimation of performance. Despite countermeasures, a decision tree tends to *overfit* the data on which it was trained, meaning that the model reflects (some of) the errors or noise in the training data. Validating the decision tree using a separate test set ensures performance of the trained decision tree is evaluated fairly.

Furthermore, several parameters can be set prior to training a decision tree on the specified training set. They might affect performance of the resulting tree, which is why we aim to identify those parameter values resulting in the best performance. Combinations of values for the following parameters are tested, after which the optimal parameter set is selected:

- **Feature selection method**: {Entropy, Gini Index}
- **Maximum tree depth**: {1,2,3,4,5,6}
- **Minimum instances at leaf node**: {10,15,20,25,50,75,100,150,200,250}

This results in a total of $2 * 6 * 10 = 120$ possible combinations. The maximum tree depth is not allowed to exceed six depths, to ensure interpretability of the resulting decision tree.

For each of the 120 possible parameter sets a decision tree should be fitted and evaluated. To ensure reliability and robustness of the evaluated performance of each parameter set, we implement a technique called *k-fold cross-validation*. This technique partitions the training data (70% of complete instance set) into k equally sized subsets. Subsequently, the learning method is applied and evaluated k times, each time reserving a different subset for testing and using the remaining $k-1$ subsets of the data for training. Using this technique, each subset is used exactly $k-1$ times for training and once for testing, making maximum use of the available data. Furthermore, we *stratify* each of the k subsets, which ensures that random sampling is done such that each class is evenly represented in each subset, meaning that each subset is representative of the underlying decision making process. We set $k = 10$, following the recommendation by Han et al. (2011) who state that stratified 10-fold cross-validation generally leads to relatively low bias and variance in performance. For each of the 120 possible parameter sets the training data is split into ten stratified subsets, after which ten decision trees are trained and subsequently tested with a different subset used for testing in each iteration. The performance of each of the ten trees is averaged to obtain the performance of the concerned parameter set. Refer to Figure 3.13 for an illustration of this stratified 10-fold cross-validation method.

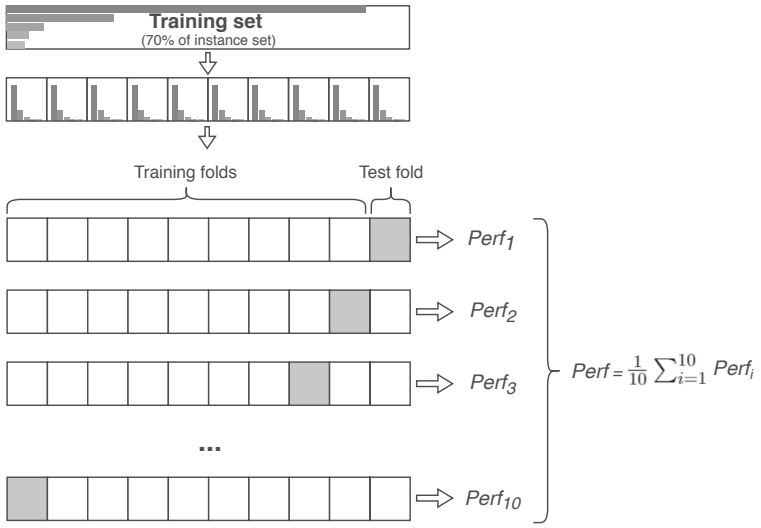

Figure 3.13: Illustration of stratified 10-fold cross-validation, applied to each parameter set

The parameter set with the best mean performance is selected as optimal for the problem at hand. Now that the optimal parameter values have been determined a decision tree can be trained using the complete training set (70% of complete instance set) and these parameter values. Subsequently, the resulting decision tree is evaluated using the test set (30% of complete instance set). Note that this test set has not been used up to this point, meaning that the decision tree to be evaluated does not depend on any of the instances in this test set, allowing for an independent performance evaluation. The complete machine learning approach is listed in Algorithm 1.

Some parallels can be drawn between the problem at hand and so-called *ranking problems*, which refer to the application of machine learning techniques to train a model in ranking tasks

Algorithm 1: Algorithm for parameter optimization and decision tree induction

```
1: Split data set in training set (70%) and test set (30%)
2: for each possible parameter setting do
3:     Split training set in ten stratified subsets
4:     for each of the ten subsets do
5:         Set concerned subset as 'parameter test set'
6:         Train decision tree on union of remaining nine subsets
7:         Test resulting tree on 'parameter test set'
8:     Compute mean performance of current parameter set
9: Select parameter set resulting in best mean performance
10: Train decision tree using best parameter set and entire training set (70% of data)
11: Test resulting tree on entire (unseen) test set (30% of data)
```

H. Li (2011). However, the problem at hand is of an easier nature since we are only interested in the dispatch option that is selected, i.e. the option that would be ranked highest in a ranking problem. In the domain of ranking problems, three approaches are distinguished: a pointwise, a pairwise, and a listwise formulation. In case of a pointwise approach, the group structure of ranking is ignored and a model is trained to be able to predict the ranking of a single option, regardless of the other options it is ranked against. Alternatively, a pairwise approach transforms each list of n options that need to be ranked to $\binom{n+1}{2}$ pairs of options and predicts their relative order. From a complete set of relative rankings the resulting ranking may be deducted. However, such a pairwise approach also ignores the underlying group structure, meaning that it might result in conflicting results (e.g. $a > b$, $b > c$, and $a < c$). Lastly, a listwise approach may be applied, which addresses the ranking problem by considering the entire group of options to be ranked at once. The approach we adopt resembles the listwise approach, since we prefer a global view of all options in a dispatch problem and because of the intuitive adaption of this approach to a classification problem in which the resulting dispatch decision, i.e. highest ranking option, is the class.

3.2.2 Performance evaluation

In the description of the applied learning algorithm above, we refer to 'performance' multiple times. However, we still need to define performance. The most common performance measure in machine learning is *accuracy*, which is defined as the percentage of instances in the test set for which the class (i.e. dispatch decision) is predicted correctly by the decision tree. However, this measure is misleading in case of an imbalanced instance set (Han et al., 2011). Here, a data set is said to be imbalanced if classes in the instance set are clearly unevenly distributed. For such instance sets the accuracy of a decision tree is not very insightful, as predicting each instance to belong to the majority class might already result in a high accuracy value. From Figure 3.3 it can be concluded that the instance set at hand is imbalanced.

For imbalanced instance sets the *confusion matrix* provides more insight into the performance of a decision tree. A confusion matrix shows the number of instances predicted to belong to each class relative to their actual class. An illustrative example is shown in Figure 3.14a.

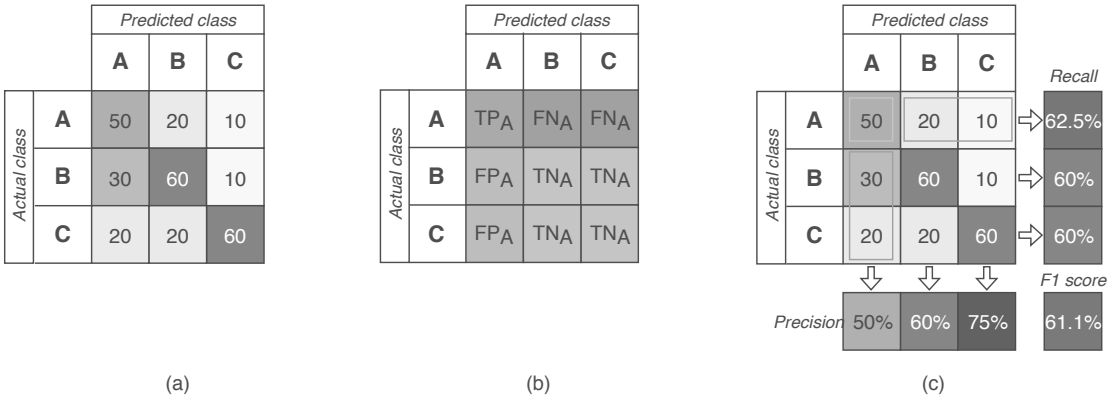

Figure 3.14: (a) Example of confusion matrix for a three-class problem; (b) Indication of TP, TN, FP, and FN values for class A; (c) Recall and precision values for all classes, and weighted F1-score

In case of binary classification it is relatively straightforward to distinguish the number of *true positives* (TP), *true negatives* (TN), *false positives* (FP), and *false negatives* (FN). In case of a multiclass problem, however, these measures can be computed for each class separately. Refer to Figure 3.14b, in which these measures are indicated for class A. Given the confusion matrix, we define the measure of *Recall* and the measure of *Precision*. While recall measures, for each class, the correctly predicted percentage of instances that actually belong to that class (Equation 3.1), precision measures, for each class, the correctly predicted percentage of instances that were predicted to belong to that class (Equation 3.2).

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3.1}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{3.2}$$

Figure 3.14c shows the recall and precision values for each class. In this Figure, the values of the confusion matrix that were used to compute the recall and precision for class A are indicated using green and red.

Generally, increasing recall leads to a reduction of precision and vice versa. This calls for a measure capturing the balance between recall and precision. This is exactly the nature of the F1-score. The F1-score for each class is computed as follows:

$$F1score_i = \frac{2 * recall_i * precision_i}{recall_i + precision_i} \tag{3.3}$$

To compute the overall F1-score of a decision tree, including all classes, one can use either the *unweighted* or the *weighted* average of the F1-score of each class. In case of the weighted F1-score, the F1-score of each class is weighted by the number of instances belonging to that class. The F1-score is a commonly used performance measure in case of an imbalanced instance set. We use the weighted F1-score both in the evaluation of decision trees trained during the identification of the optimal parameter set, as well as to evaluate the final decision tree trained using the complete training set and tested using the complete test set. In Figure 3.14c the weighted F1-score is shown.

Besides the weighted F1-score, performance of the trained decision tree will also be evaluated in terms of the *Weighted Mean Error*. In most machine learning problems the distance between two classes is meaningless (e.g. when classifying animals). However, for the problem at hand, if the actual dispatch decision was dispatch of the first option, predicting dispatch of the third option is actually *more wrong* than predicting dispatch of the second option. When using the learned model as a benchmark for the current dispatch policy in a simulation, simulated performance is more likely to resemble actual performance if wrongly predicted dispatch decisions are "closer" to the actual dispatch decision. Therefore, specifically for this problem we define the following additional performance measure:

$$WME = \sum_{d=0}^{k-1} d \sum_{i,j \in \{1,2,\ldots,k\}:|i-j|=d} m_{i,j} \tag{3.4}$$

Here, k equals the number of possible classes and $m_{i,j}$ are cells in the confusion matrix, where rows and columns are indicated by i and j respectively. Naturally, while we strive towards a dispatch prediction model with a weighted F1-score that is as high as possible, we prefer the mean distance to actual class to be as low as possible.

To place the performance of the resulting decision tree into perspective, its performance will be compared to the dispatch policy that is commonly assumed in literature, the *closest-idle policy*. However, in literature this policy generally does not include the additional dispatch options that are available to BZO's dispatch agents, namely ambulances that are not completely idle but nevertheless available to (certain) incidents and external ambulances that belong to other regions (see Section 2.3). Therefore, we define two dispatch policies to which the performance of our fitted dispatch policy will be compared:

- **The limited closest-idle policy**: corresponding to the policy that is commonly assumed in literature, i.e. dispatching the highest ranking option in the dispatch proposal that is completely idle (on the road or at station) and belongs to the own region
- **The extended closest-idle policy**: corresponding to the commonly assumed policy but adapted to include the additional available dispatch options, i.e. always dispatching option one in the dispatch proposal

3.2.3 Data set imbalance

As described in Section 3.2.2 our instance set is imbalanced, which is why we adopt a performance measure fit for imbalanced data rather than the common measure of accuracy. Several methods exist to balance instance sets, e.g. under- or oversampling, possibly using advanced techniques such as SMOTE. These methods are useful whenever the class of interest, i.e. of which it is important that it can be predicted correctly, is (one of) the minority class(es).

However, our objective of capturing the current dispatch process is to identify which ambulance is actually dispatched. Therefore, the larger sized classes are of greater interest than the smaller ones. By definition, this relative interest in correctly predicting each class is reflected in the class distribution. The fact that the frequency of each class in the instance set reflects its importance of being correctly classified, eliminates the need for balancing the instance set. In line with the importance of the learned decision tree being able to correctly predict each class

reflecting the class distribution, the performance measure used to evaluate the decision tree should also reflect this relative importance, which is why the weighted F1-score is used.

3.3 Induction results and insights

Figure 3.15a, b, and c show the confusion matrices and summarizing performance measures for the dispatch model learned through Algorithm 1, the limited closest-idle policy and the extended closest-idle policy respectively.

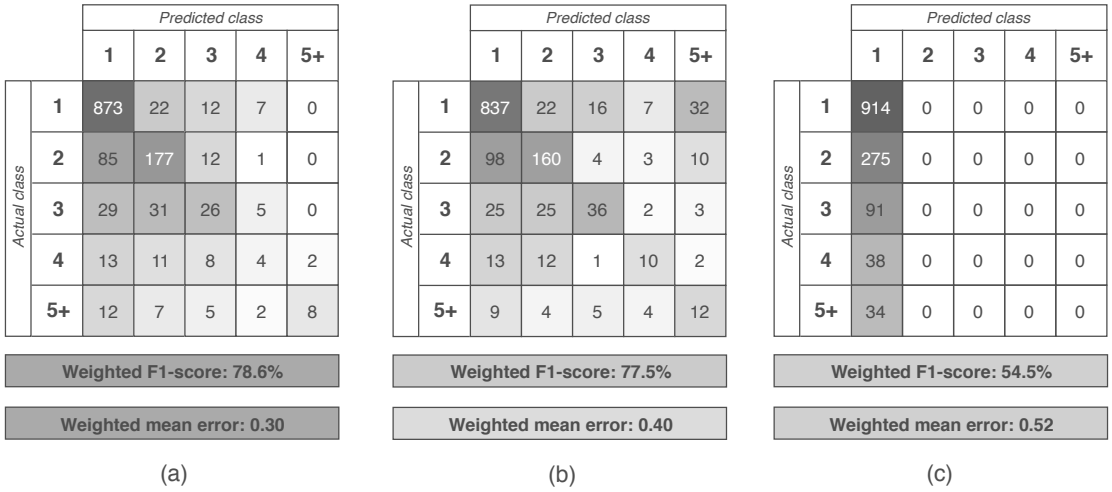

Figure 3.15: Confusion matrices and summarizing performance measures for (a) the learned dispatch model, (b) the limited closest-idle policy, and (c) the extended closest-idle policy

Figure 3.15 shows that the learned dispatch model outperforms both interpretations of the closest-idle policy, in terms of the weighted F1-score, as well as the weighted mean error. However, while the difference in performance between the learned model (a) and the extended closest-idle policy (c) is quite significant, the improvement in predicting performance of the learned model (a) relative to the basic, limited closest-idle policy (b) is less apparent. This observation leads us to believe that BZO's dispatch agents generally make limited use of the additional dispatch options available to them.

This insight is confirmed by studying the learned decision tree, depicted in Figure 3.16, in more detail. There are several clear 'decision paths' to be distinguished in this decision tree, which have been highlighted in Figure 3.16. These highlighted decision paths indicate the dominant dispatch decision that was made for those instances following the concerned path. Note that some of these paths, and the insights derived from them, can be regarded as more dominant, or important, than others due to the larger number of samples following that path. Generally, nodes that are higher up in the decision tree are reached by a larger number of samples, stressing their importance above those that are closer to the tree's leaf nodes. The weight of each path indicates the number of samples following that path.

The main reasons that might lead a dispatch agent to deviate from dispatching the highest ranking dispatch option (i.e. option 1) quickly become clear from the splits on the most dom-

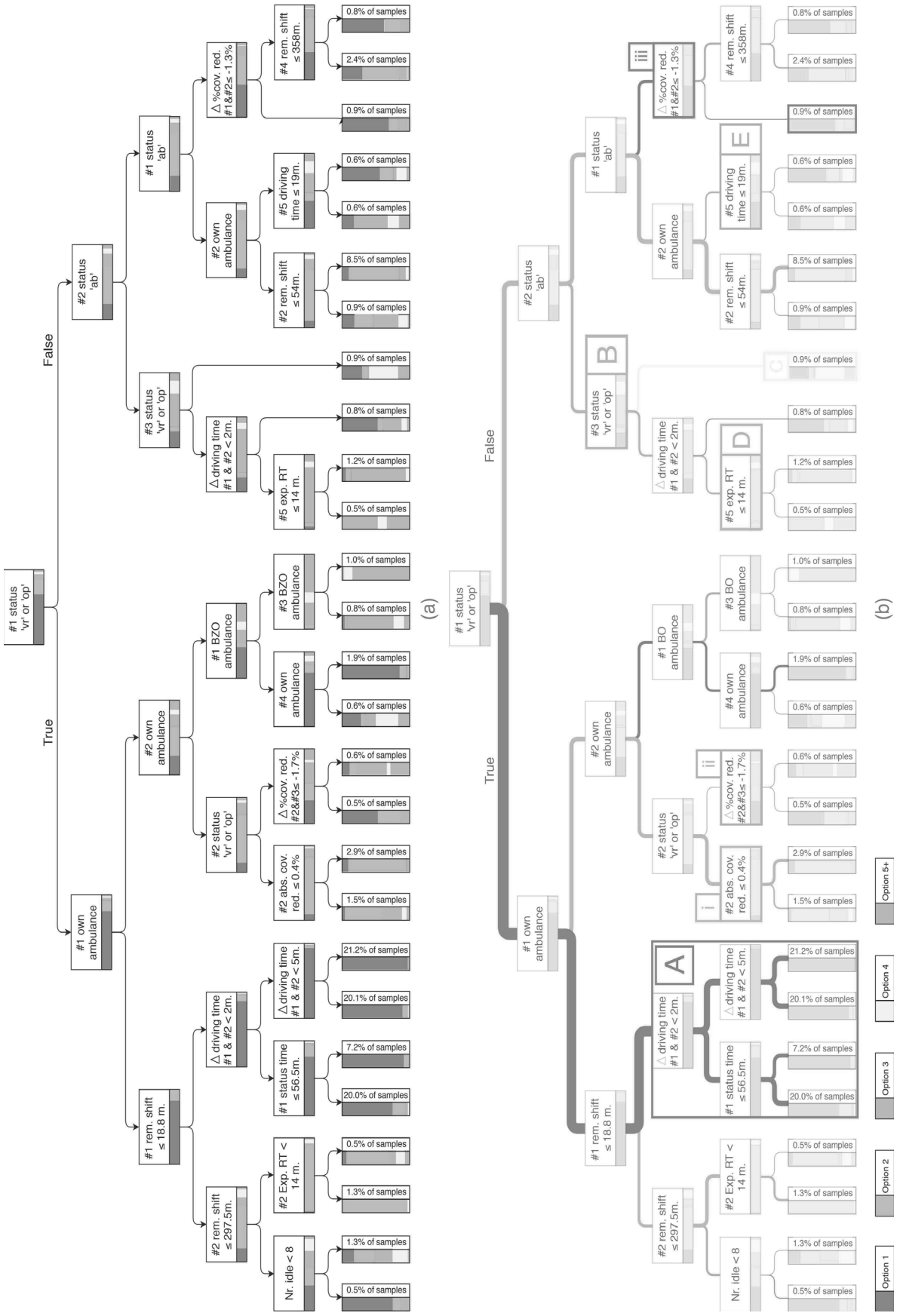

Figure 3.16: Visualization of learned dispatch decision model with colors indicating the class distribution of the instances reaching each node: (a) complete model and (b) model including highlighted decision paths indicating its dominant dispatch decision

inant path (leading to [A]). These main reasons include this highest ranking ambulance:

- **Not being immediately available for dispatch**: due to its status. For example, the ambulance is transferring a patient at a hospital and might require some time to finish this transfer, or it is on its way to a less urgent request, meaning that it might require some time to be relieved from its current request and redispatched to the new request.
- **Not belonging to the own region**: meaning that the concerned dispatch center needs to be requested to dispatch it, which takes time and might be denied.
- **Nearing the end of its shift**: causing a risk of overtime if it is dispatched.

The first two of these reasons confirm that dispatch agents make limited use of the additional dispatch options available to them. Possibly, this is the case because these issues add a potential delay to the indicated driving time to the request. Such a potential delay adds a degree of uncertainty to the ambulance's expected driving time, causing the dispatch agent to consider deviating from this option. Naturally, the potential delay is only relevant if the difference between the driving time of that option and the subsequent option is less than this expected delay. This is reflected by the node at the top of node group [A], as well as at several other nodes in the tree. Note that the first two of these reasons to deviate from the highest ranking ambulance are also reflected in Figure 3.11, which indicates those features which have the highest (absolute) correlation with the resulting dispatch decision.

It can be deducted that, if there are enough reasons to deviate from the highest ranking ambulance option, the subsequent option is considered. However, the same reasons to deviate seem to hold for this option, e.g. see the path leading to node [B], where option 3 is considered due to the status of option 2, and that same path eventually leading to leaf node [C], where option 4 is considered due to the status of option 3.

However, subsequent options cannot be considered indefinitely, since the driving time to the request increases with each option. Naturally, despite the dispatch agents being risk averse and preferring subsequent options if there is a potential delay for the closest option, the selected option should still be able to arrive on-time. Since the driving time increases with each option, the driving time, or expected response time, of the furthest option we consider, option 5, is a good indication of whether previous options are able to arrive on-time. This is why multiple nodes testing for the closeness of option 5 to the incident are present in the decision tree, see nodes [D] and [E]. It can be seen that if the closeness of option 5 is sufficiently small, generally lower ranked options are selected for dispatch than when this is not the case.

This is also why the learned model performs significantly better than the limited closest-idle policy in terms of its weighted mean error. In case of sufficient available capacity, dispatch agents clearly prefer risk averse dispatch options. However, while the learned model recognizes that in case of scarcity the dispatch agent is required to choose an ambulance to be dispatched among 'bad' risky options, the limited closest-idle policy keeps considering subsequent options until a risk-free (completely idle and own region) option is found. In other words, while the performance of the fitted model is similar to the limited closest-idle policy for the majority of dispatch decisions to be made, i.e. in case of sufficient capacity, it strongly outperforms this commonly assumed policy in case of scarce capacity. This ability of the fitted model is especially relevant since dispatch decisions made under scarce capacity are precisely where the expertise and human judgment of the dispatch agents can make a difference.

3.3.1 A penalty-based dispatch model

However, the fitted dispatch policy is quite complex. Combined with the fact that a simple model such as the limited closest-idle policy is able to predict dispatch decisions quite well in case of sufficient ambulance capacity, but performs very bad in case of limited capacity due to its inability to consider 'bad' options, leads us to propose a concise, **penalty-based model** to represent the dispatch decisions made by BZO's dispatch agents. In line with the three main reasons to deviate from dispatching an ambulance that were deducted from the fitted model, penalty terms are defined based on an ambulance's status, region and time until the end of its shift to reflect the potential delay or risk associated with the value of these features.

For each ambulance option, its total time penalty, is determined based on its status, region and remaining shift time, after which it is added to its driving time. Next, the dispatch option with the lowest driving time plus total penalty is dispatched. This approach reflects dispatching agents' preference for a completely idle ambulance from the own region, but ensures that in case of scarce capacity still one of the 'bad' options is selected for dispatch.

Similar to the machine learning effort, these penalty terms are fitted on the training data (70% of total instance set), such that they result in a maximum weighted F1-score. This is done through an exhaustive search of integer penalty values. The performance of the resulting penalty model is evaluated on the test data (remaining 30% of instance set). Figure 3.17 shows the fitted penalty values, the confusion matrix and performance measures. It is shown that both the weighted F1-score and the weighted mean error have improved even further compared to the fitted decision tree. The penalty-based model is presented in Algorithm 2.

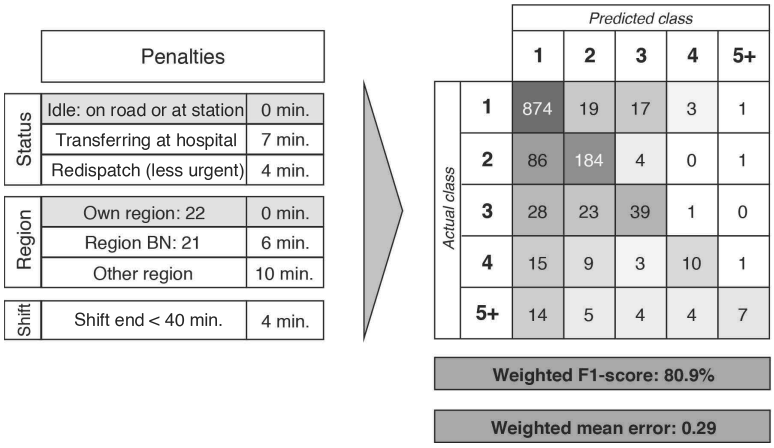

Figure 3.17: Fitted penalty values (on training data) and performance (on test data) of the penalty-based model

In conclusion, insights from our fitted dispatch decision prediction model were used to enrich the commonly assumed closest-idle dispatch policy using penalty values reflecting the risk associated with certain ambulance characteristics. The result of this postprocessing phase is a concise model that has significantly greater resemblance to the actual dispatch decisions made by BZO's dispatch agents compared to the policy that is generally assumed in literature. This provides us with a policy to build upon to improve the current dispatch process, as well as with

Algorithm 2: Algorithm of the penalty-based dispatching model

for each dispatch option in the dispatch proposal i **do**
    Penalty$_i = 0$
    if ambulance is transferring a patients at a hospital **then**
        Penalty$_i$ = penalty$_i$ + 7 (min.)
    else if ambulance is on its way to a less urgent request **then**
        Penalty$_i$ = penalty$_i$ + 4 (min.)
    if ambulance is of BNO region **then**
        Penalty$_i$ = penalty$_i$ + 6 (min.)
    else if ambulance is from neither BZO nor BNO region **then**
        Penalty$_i$ = penalty$_i$ + 10 (min.)
    if shift of ambulance ends within 40 minutes **then**
        Penalty$_i$ = penalty$_i$ + 4 (min.)
    Penalized driving time of i = driving time of ambulance i + penalty$_i$
Dispatch ambulance with smallest penalized driving time

a benchmark that is close to current practices, such that the improved dispatch model can be evaluated fairly in a simulation. Figure 3.18 shows an overview of the applied formalization approach, including the postprocessing phase resulting in the final penalty-based model.

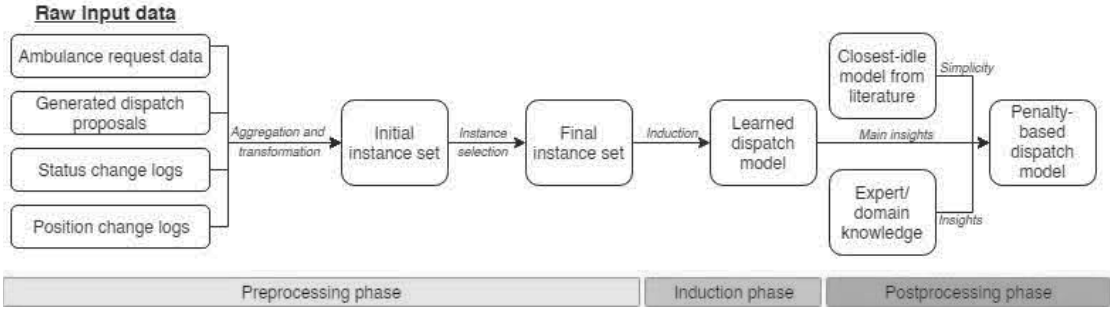

Figure 3.18: Overview of total approach to capture current dispatch practices

3.4 Potential enhancements to the dispatch policy

Besides using the formalization of the current dispatch decision process as a benchmark to evaluate a potentially improved dispatch decision process, this formalization is also used to build such improvements upon. Using the current dispatch process as a basis for improvements ensures that practically relevant considerations are included in the improved decision process. Furthermore it increases the probability of the resulting dispatch process fitting the way in which dispatch agents currently work, and thus the probability of the improved process being adopted in practice. Based on a combination of insights from the captured current dispatch policy, discussions with dispatch agents and the limited amount of available literature on alternative dispatch policies, four potential enhancements to the current dispatch process were defined. The effect of these individual potential enhancements, as well as possible combina-

tions, on performance will be evaluated using a simulation. In this section we will introduce the proposed enhancements of the dispatch process. Figure 3.19 shows an illustrative example of each of the four potential enhancements to the dispatch process.

Consistent redispatching
From the formalized current dispatching process, it can be seen that a dispatch option that is not completely free (on the road or at a station), is considered to be risky due to a potential delay. While a potential delay is difficult to avoid if the ambulance is busy transferring a patient at a hospital, it might be avoided in case of redispatching an ambulance that is currently on its way to a less urgent request. Discussions with dispatch agents have shown that considerations for avoiding redispatching include the additional work of relieving the ambulance of its current request assignment, as well as dispatching another ambulance to this request it was originally assigned to. Furthermore, being redispatched too frequently is undesirable from the perspective of the ambulance crew. However, it is interesting to evaluate the potential performance improvement of consistently redispatching an ambulance whenever it is the best dispatch option, since the performance improvement might outweigh the disadvantages and might even lead to system adaptions to mitigate some of these inconveniences. The enhancement is similar to 'reroute-enabled dispatching' as proposed by Lim et al. (2011), though those authors did not evaluate this policy in a realistic(ally sized) EMS system.

Reevaluation of dispatch decision
Furthermore, currently dispatch decisions are only made upon arrival of a new request. A dispatch decision is made by selecting the best option from those ambulances that are available at that moment. However, the system of ambulances is very dynamic and during the time the dispatched ambulance is driving towards the request, another ambulance may complete serving another request. This, newly idle, ambulance may in fact be a better dispatch decision than the ambulance that is already on its way. Especially if the dispatched ambulance might exceed the response time threshold for the concerned request, while the newly idle ambulance is able to reach the request location on-time, reevaluation of the dispatch decision might contribute towards improving performance. Contrary to the 'Parallelism' dispatch policy of Lee (2014), the consideration of a busy ambulance only after it has completed service prevents dependency on the realization of highly variable treatment times. To prevent reevaluated dispatch decisions resulting in only a marginal difference in response time, as is the case for the 'free ambulance exploitation' policy of Lim et al. (2011), a reevaluated dispatch decision will only lead to the recently freed ambulance being dispatched instead of the current one if this leads to a response time improvement of at least one minute for highly urgent (A1) requests, or a direct improvement of the on-time performance for less urgent (A2) requests.

Minimum coverage reduction dispatching
The current dispatch process is predominantly focused on ensuring that the response time of the ambulance request at hand does not exceed its threshold, by assessing the driving time and potential delay factors of the available options. The extent to which dispatch agents consider the preparedness of the region for the arrival of requests in the (near) future, e.g. by dispatching the ambulance that causes minimum coverage reduction among those that are able to arrive on-time, is very limited. Nodes [i], [ii], and [iii] in Figure 3.16 are only reached by a small number of instances and are almost equivalent to testing whether coverage reduction is zero, meaning that the concerned dispatch option is likely to be located at the exact same location as another. Taking into account the coverage reduction involved with each dispatch

option that is able to arrive to the request on-time and selecting the one with the minimum value might improve the resulting preparedness of the region for ambulance requests arriving in the near future. While the on-time performance is the main performance measure for both A1 and A2 requests, A1 requests are potentially life-threatening. Combining this with the fact that both Jagtenberg et al. (2017) and Lee (2011) showed that taking into account the coverage reduction of each dispatch option results in a significant increase of the mean response time, leads us to apply this enhancement only to less urgent (A2) requests. The authors of both studies did not distinguish between requests of different urgencies.

Postpone A2 dispatches

Lastly, it is interesting to note that the urgency of the request to which an ambulance needs to be dispatched does not appear in the dispatch decision prediction model, despite the main performance measure depending on it. This does not mean that BZO's dispatch agents do not take the request's urgency into account at all. However, it does indicate that the dispatch decision process does not heavily depend on it. Especially the fact that the on-time performance of less urgent (A2) requests is well above its target of 95%, while the on-time performance of urgent (A1) incidents is consistently below its target of 95% within fifteen minutes, leads us to believe that there is potential in improving the performance of A1 requests at the expense of performance of A2 requests by adapting the dispatch policy accordingly. Due to the response time target of 30 minutes there is often a large number of ambulances that are able to reach an A2 request location on-time, while the number of eligible ambulances is often much smaller for A1 requests because of the smaller response time target of 15 minutes. Especially when there is only a small number of idle ambulance available for dispatch, say three or less, it might be beneficial to postpone dispatching an ambulance to an A2 request to keep ambulances available for the case that one or multiple A1 requests arrive. This enhancement was proposed by dispatch agents themselves.

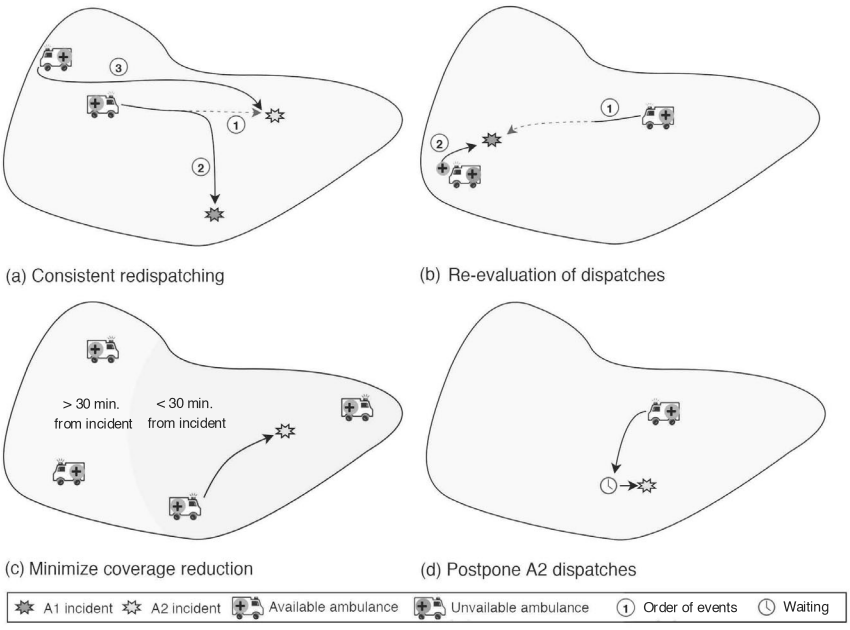

Figure 3.19: Illustrative examples of potential dispatch enhancements

4 | Improving the dispatch process

To evaluate the potential of the proposed enhancements to the dispatch process, the BZO ambulance services are simulated. Section 4.1 describes the setup of this simulation, after which the results are presented in Section 4.2. Section 4.2.4 places the results into perspective.

4.1 Simulation setup

Ambulance movements in the BZO ambulance region are simulated using discrete-event simulation (DES). This type of simulation allows for evaluation of the system only at relevant (decision) moments in time. Figure 4.1 shows the general architecture of this simulation, including the queue of events, which are ordered on the time of occurrence. First, the main modeling choices and assumptions underlying this simulation will be listed, after which the details of the simulation will be outlined.

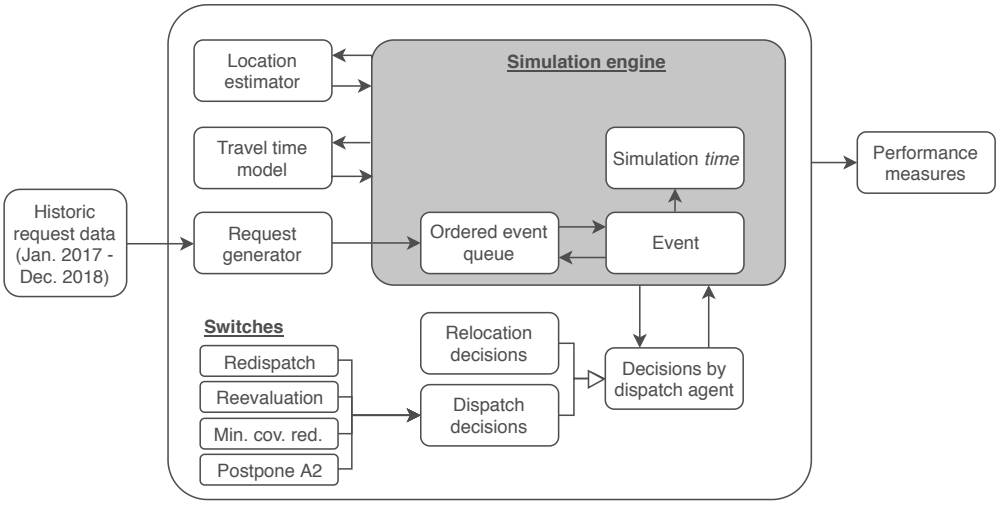

Figure 4.1: General architecture of the simulation

In this simulation the BZO region is aggregated into 138 subregions, corresponding to 4-digit postal codes. Locations of ambulance stations, hospitals, and requests are mapped onto the centroid of its postal code. While our focus is on the performance of urgent (i.e. A1 and A2) requests, all types of requests that are served by ALS vehicles are simulated to be able to capture all dynamics in the utilization of ALS capacity. These include all non-urgent transports of non-stable patients (B1), as well as the occasional non-urgent transport of stable patients (B2) in case of insufficient capacity of BLS ambulances. Furthermore, driving times between each pair of postal codes are assumed to be deterministic, as supplied by the *driving time model* of the RIVM (non-public). This driving time model estimates driving times under

the assumption of driving with siren, i.e. to an A1 request, and distinguishes between driving times during the rush hour (weekdays 6:30 - 9:30h and 15:00 - 19:00h), nighttime (19:00 - 6:30h), and daytime (otherwise). Driving times without the use of the siren, i.e. to an A2 or B1/B2 request and for relocations, are obtained by multiplying the driving times from the RIVM's model by a factor 1.45. This factor was estimated by comparing the driving times from the model with those obtained from Google Maps, i.e. driving times realized when the speed limit cannot be exceeded and traffic lights cannot be ignored.

Lastly, the interaction with neighbouring EMS regions is excluded from the simulation due to its complexity. This choice is justified by the fact that these external ambulances are rarely dispatched and that approximately the same number of such external ambulances are dispatched to requests in the BZO region as vice versa (<2% of requests).

Besides these modeling choices, a number of assumptions was made. The most important of those are listed below, and will be elaborated upon in the relevant sections presenting the simulation details. Any other assumptions that were made in the design of the simulation, are discussed in subsequent sections as well.

- Ambulance requests are assumed to arrive dynamically according to a Poisson process with a time-dependent rate depending on fifteen minute time slots throughout a week, see Section 4.1.4. Every week request arrivals follow this same pattern, meaning that the effects of holidays and trends over time are neglected.
- Ambulance shifts are assumed to start according to a static weekly shift roster, which is based on realized shifts, i.e. corrected for illness or holidays, see Appendix A.
- Ambulances are assumed to move from origin to destination 'as the crow flies', i.e. in a direct line, with constant speed according to the total driving time as given by the driving time model, such that the current location of a driving ambulance can be approximated, see Section 4.1.13.
- All dispatch decisions, also for non-urgent transport requests (B1/B2) in case of sufficient ambulance capacity, are assumed to be made according to the penalty-based model in Algorithm 2, see Section 4.1.13.
- Upon each change in the number of available ambulances, i.e. due to a dispatch or service completion, available ambulances are assumed to be relocated according to the region's compliance table, see Section 4.1.12.

4.1.1 Relevant entities and events

There are two main entities of which unique instances are created throughout the simulation and of which information is stored, namely **requests** and **ambulances**. Upon arrival, a request instance is created with attribute values such as its urgency, location, treatment time at the scene, whether a hospital visit is required, and if so, to which hospital and the required transfer time. At the start of each shift ambulance instances are created according to the shift roster and are located at its base station. Throughout its shift, which lasts eight hours, an ambulance instance may be dispatched to arriving requests. While serving a request, the ambulance's status, origin, destination, and driving time are updated regularly.

Seven types of events are defined. Figure 4.2 displays these event types and the possible

transitions between them from the perspective of an ambulance. An ambulance always **starts its shift** at its base station, which is also where it has to **end its shift**. Upon **arrival of a new request**, an ambulance might be dispatched. As long as the ambulance has not yet arrived at the request's location, it might be redispatched to another request with a higher urgency. Upon **arrival at the request location**, the next event for this ambulance is scheduled, which is, depending on whether the concerned request requires transportation to a hospital, either the arrival of the ambulance at the request's hospital, or the completion of service at the request's location. Upon **arrival at the hospital**, a service completion event is scheduled according to the transfer time required for the request. However, before this **service completion** event takes place, the ambulance may already be dispatched to another request from the hospital. Note that the ambulance requires at least ten minutes to transfer the patient, meaning that the ambulance may not be able to depart for the new request right away. Each time the number of free ambulances changes, either due to an ambulance being dispatched, or an ambulance completing service, it is determined whether relocation movements need to be initiated. Ambulances may be dispatched already before **arriving at a station**.

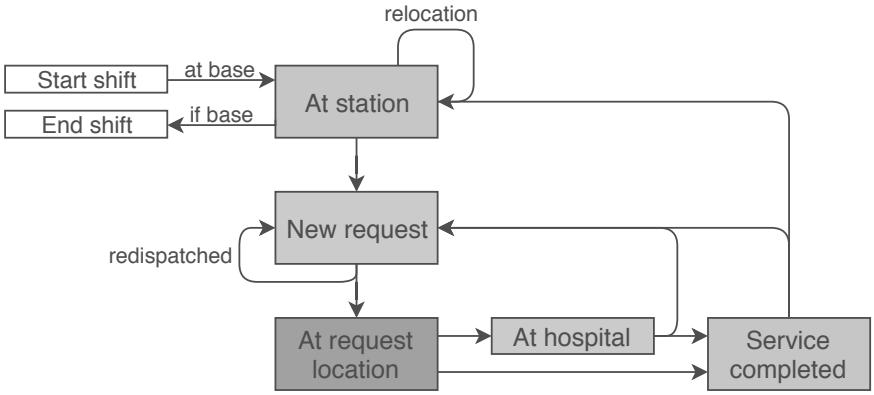

Figure 4.2: Simple representation of relevant simulation events from the perspective of an ambulance

4.1.2 Performance measures

The main performance measures we are interested in are those relating to the response time of urgent requests, i.e. the fraction of ambulances arriving *on-time* and the *mean response time* to both A1 and A2 requests. Therefore, each time an *Arrival at request location* event occurs, the response time, i.e. the difference between the current time and the request's arrival time, is logged. In accordance with the RIVM's measurement plans (RIVM, 2010), in case of an emergency requiring multiple ambulances, only the response time of the ambulance that arrives first contributes to performance measures.

4.1.3 Simulation framework

The setup of the simulation will be presented using the simulation framework as presented in Pseudocode 1. This framework shows that the simulation is run for a given number of runs, in which each run consists of a given number of weeks. Both of these parameters are set

such that sufficiently small confidence intervals can be determined of the relevant performance measures, see Section 4.2. During each run events are scheduled and handled such that actual processes are reflected as accurately as possible.

Pseudocode 1: Simulation framework

```
1: Import relevant data
2: for each run do
3:     Set simtime to Monday morning 7.00h
4:     Generate and schedule the first request arrival                ▷ Section 4.1.4
5:     Set number of weeks to simulate
6:     while simtime is less than the number of weeks to simulate do
7:         Set e ← first event of the ordered event queue
8:         Set simtime ← the time of event e
9:         Set req ← request corresponding to first event (if applicable)
10:        Set ambu ← ambulance corresponding to first event (if applicable)
11:        if Event e is of type New request arrival then
12:            Handle New request arrival event                       ▷ Section 4.1.5
13:        else if Event e is of type Arrival at request location then
14:            Handle Arrival at request location event               ▷ Section 4.1.6
15:        else if Event e is of type Arrival at hospital then
16:            Handle Arrival at hospital event                       ▷ Section 4.1.7
17:        else if Event e is of type Service completion then
18:            Handle Service completion event                        ▷ Section 4.1.8
19:        else if Event e is of type Arrival at station then
20:            Handle Arrival at station event                        ▷ Section 4.1.9
21:        else if Event e is of type Start shifts then
22:            Handle Start shifts event                              ▷ Section 4.1.10
23:        else if Event e is of type End shift then
24:            Handle End shift event                                 ▷ Section 4.1.11
25:    Compute performance measures
26: Compute mean performance measures, including confidence intervals
```

4.1.4 Generating and scheduling request arrivals

To be able to draw sound conclusions regarding performance under different dispatch policies, multiple independent simulation runs are required. Therefore, a distribution is fitted on the arrival time of request arrivals using historic data from the years 2017 and 2018. This data includes all A1, A2, and B1 requests, as well as those B2 requests to which an ALS ambulance was dispatched due to insufficient capacity of BLS ambulances. Generally, requests, regardless of its urgency, occur independently from each other, which is why a Poisson arrival process is assumed. This assumption generally holds for real-world medical emergency systems (Galvao & Morabito, 2008). Figure 4.3 shows the number of requests throughout an average week between January 2017 and December 2018. The arrival intensity is strongly time-dependent throughout the week. Therefore, we model the arrival process as a non-homogeneous Poisson process, which is a Poisson process where the rate is time-dependent. A week is split into $7 * 24 * 4 = 672$ time periods of fifteen minutes, which together form set $\mathcal{P}$, and an arrival intensity $\lambda(p)$ is determined for each time period $p \in \mathcal{P}$.

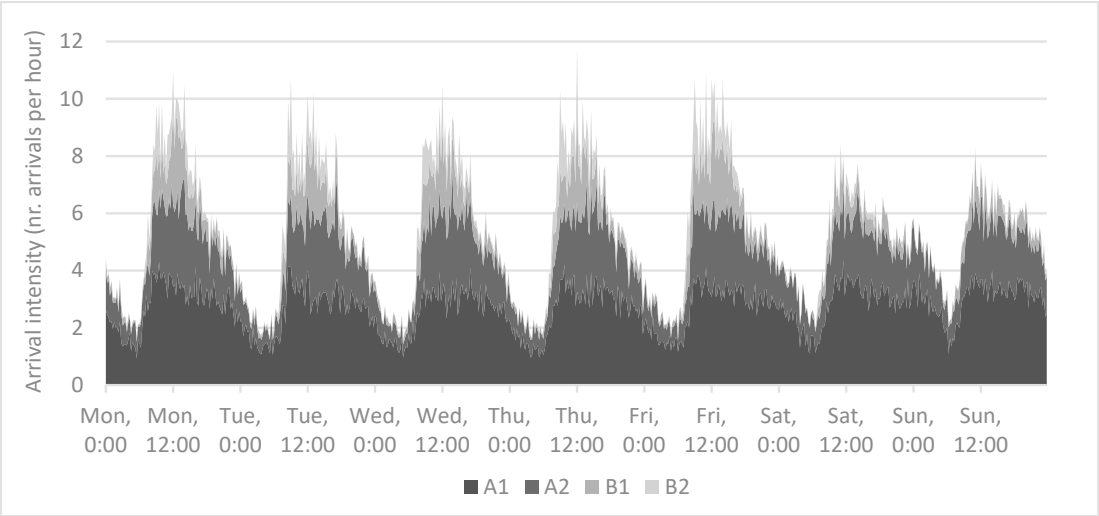

Figure 4.3: Cumulative arrival rate for requests of each urgency in an average week between January 2017 and December 2018

This non-homogeneous Poisson process is simulated using a technique called *thinning* (Boon, van Leeuwaarden, Mathijsen, van der Pol & Resing, 2017). The interarrival time of the next request, t_{int}, is generated according to an exponential interarrival time distribution with an arrival intensity λ_{max} equal to the maximum arrival intensity over all time periods:

$$\lambda_{max} = \max_{p \in P} \lambda(p) \tag{4.1}$$

Each possible request arrival is accepted with probability $\frac{\lambda(p)}{\lambda_{max}}$ (see Section 4.1.5). For each possible request arrival, that is scheduled, a request sample is drawn from the historic data for the corresponding time period $p \in \mathcal{P}$. This approach ensures that time-dependent request characteristics, such as the proportion of each urgency, location patterns, waiting times at hospitals etc., are captured by the simulation. A request instance is created with attributes corresponding to the drawn request sample, which are the following:

- The request's urgency (A1, A2, B1, B2)
- The request's location (postal code)
- The required treatment time at the request location (hours)
- The hospital location (postal code, if applicable)
- The required transfer time in the hospital (hours, if applicable)
- The number of ambulances that is required

A historic request may have required multiple ambulances, for example in case of a reanimation. Information regarding such additional ambulance requests is also included in each request sample. This information includes the required treatment time, hospitalization, hospital location (if applicable), and the required transfer time in the hospital (if applicable) of the additional ambulance request. Furthermore, the sample holds information regarding the

time, after the original request, the request for the additional ambulance arose. If a drawn request sample requires multiple ambulances, additional request arrivals are scheduled. Figure 4.4 shows an example of two request samples.

		Original Request					Additional ambulance request 1				Add. requests 2+	
Time period	#ambulances	Urgency	Location	Treatment time (h)	Hospital location	Transfer time (h)	Δt_1 (h)	Treatment time (h)	Hospital location	Transfer time (h)	Δt_2(h)	...
8	2	A1	5611	0.34	5504	0.18	0.10	0.23	N/A	N/A	N/A	...
342	1	A2	6029	0.49	5707	0.11	N/A	N/A	N/A	N/A	N/A	...

Figure 4.4: Two request samples: in time periods 8 (Monday 01:45 - 02:00h) and 342 (Thursday 13:15 - 13:30h); first requires additional ambulance 6 minutes after the original emergency request

Pseudocode 2: Generating and scheduling a request arrival

1: Draw next random interarrival time t_{int} from distribution with $\lambda_{max} = \max_{p \in P} \lambda(p)$
2: Draw random request sample req from current time period $p \in \mathcal{P}$
3: Schedule request arrival at $simtime + t_{int}$
4: **if** request req requires multiple ambulances **then**
5: **for** each additional ambulance request r **do**
6: Schedule additional request arrival at $simtime + t_{int} + \Delta t_r$

4.1.5 Handling a *New request arrival* event

If the first event of the ordered event queue is of the type *New request arrival*, it should first be determined whether this event is accepted with a probability that reflects the arrival rate of the current time period p. Naturally, if a *New request arrival* event corresponds to an additional ambulance request, this event is accepted if and only if the original request was accepted. If the arriving request is accepted, then it is checked whether at least one ambulance is available to be dispatched, which depends on the request's urgency as follows:

- A1: there is at least one free ambulance or ambulance on its way to a less urgent request
- A2: there is at least one free ambulance
- B1/B2: there are at least five free ambulances during daytime (6:30 - 19:00h) or at least four free ambulances during nighttime (19:00 - 6:30h)

Here, free ambulances include those transferring a patient at a hospital, since they might be requested to accelerate the transfer process if they are required for dispatch. The condition for dispatching an ambulance to a non-urgent transportation request (B1/B2) is based upon discussions with BZO's dispatch agents. If the urgency-dependent condition for dispatch is met, it is determined which ambulance should be dispatched using the chosen dispatch policy, see Section 4.1.13. Subsequently, the selected ambulance is dispatched to the request, which entails updating its attributes, such as its status, destination, and driving time. Lastly, an event is scheduled for the arrival of the ambulance at the request location. Besides the driving time from the ambulance's current location to the request location, the *delay* should also be taken into account when scheduling the arrival event. This delay always includes *chute time*,

which depends on the request's urgency, see Appendix A. In case the ambulance selected to be dispatched is currently busy transferring a patient at a hospital, the delay includes the remainder of the (accelerated) transfer time, see Section 4.1.7. There may be a (now) redundant event in the event queue referring to the dispatched ambulance, e.g. this ambulance arriving at a station after being relocated. This event should be removed from the event queue, since the ambulance changed its course to be dispatched. *End of shift* events do not have to be removed. The ambulance selected for dispatch to an A1 request may be a redispatch, i.e. it was already on its way to a less urgent request. In this case, it is checked whether an alternative ambulance can be dispatched to the initial request, similarly to dispatching to a new request.

If the urgency-dependent condition for dispatch is not met, the new request is added to a waiting list of requests. Note that there is a separate list for urgent (A1 and A2) and for non-urgent (B1 and B2) requests, such that once an ambulance completes serving a request, priority can be given to it being dispatched to waiting urgent requests.

If an ambulance, or possibly multiple ambulances in case of a redispatch, were dispatched, relocation movements are determined and initiated to improve coverage of the region by the remaining free ambulances, see Section 4.1.12. Lastly, regardless of whether the request was accepted and/or whether an ambulance was dispatched, a *New request arrival* event should be generated and scheduled, which is done as outlined in Section 4.1.4.

Pseudocode 3: Handling a New request arrival event

```
1: if request req is an original request then
2:     Accept req with probability λ(p)/λ_max
3: else if original request was accepted then
4:     Accept req
5: if request req was accepted then
6:     if there are ambulances available for dispatch to req's urgency then
7:         Determine ambulance disp to dispatch                          ▷ Section 4.1.13
8:         Dispatch the selected ambulance to request req
9:         Remove event(s) of disp from event queue (except if of type End of shift)
10:        Schedule the Arrival at request location event at simtime + driving time + delay(req, disp)
11:        if this is a redispatch then
12:            if there are free ambulances available then
13:                Determine alternative ambulance alt to dispatch to initial request   ▷ Section 4.1.13
14:                Dispatch the selected ambulance to initial request
15:                Remove event(s) of alt from event queue (except if of type End of shift)
16:                Schedule Arrival at request location event at simtime+driving time+delay(req, alt)
17:            else
18:                Add request req to waiting list corresponding to its urgency
19:    else if There are no ambulances available for dispatch to req's urgency then
20:        Add request req to waiting list corresponding to its urgency
21:    if at least one ambulance was dispatched then
22:        Determine and initiate relocations of remaining free ambulances   ▷ Section 4.1.12
23: Generate and schedule the next request arrival                           ▷ Section 4.1.4
```

4.1.6 Handling an *Arrival at request location* event

If the first event of the ordered event queue is of the type *Arrival at request location*, attributes of the concerned ambulance, such as its status, origin, and driving time, are updated to reflect this. Furthermore, the response time of the concerned request is registered. Note that only the response time of the first ambulance arriving to a request is relevant. Lastly, the next event is scheduled, which is either the arrival of the concerned ambulance at the hospital as given by the request sample, or service completion at the request location.

Pseudocode 4: Handling an Arrival at request location event

Update ambulance attributes
if ambulance is first to arrive to (original) request **then**
    Register response time: $simtime-$ arrival time of (original) request
if request req requires a hospital visit **then**
    Determine driving time between request location and concerned hospital
    Schedule an *Arrival at hospital* event at $simtime + treatment\ time_{req} + driving\ time$
else if request req does not require a hospital visit **then**
    Schedule a *Service completion* event at $simtime + treatment\ time_{req}$

4.1.7 Handling an *Arrival at hospital* event

If the first event of the ordered event queue is of the type *Arrive at hospital*, attributes of the concerned ambulance, such as its status, origin, and driving time, are updated to reflect this. Furthermore, while an ambulance arriving at the hospital still requires time to transfer the patient before it is free, dispatch agents may request the ambulance to accelerate this process if it is needed for dispatch. In formalizing the current dispatch routine, it was found that dispatch agents prefer not to dispatch such an ambulance due to the uncertainty in transfer times and the ability to accelerate these.

Nevertheless, the ambulance arriving at the hospital might be the only (reasonable) option for dispatch, for example in case of requests which arrived prior to the ambulance arriving at the hospital while no ambulances were available for dispatch. In this case, the ambulance arriving at the hospital should be dispatched to the waiting request as quickly as possible. Also, there might be non-urgent transports (B1/B2 requests) waiting for dispatch because there were not enough ambulances available. Including the ambulance that arrived at the hospital might result in enough ambulances being available (in the near future) to justify dispatching an ambulance to the waiting non-urgent transport.

A discussion with a group of dispatch agents yielded the conclusion that the minimum required transfer time, achievable in case of an accelerated transfer process, is ten minutes. Therefore, in case of a dispatch of an ambulance that is is busy transferring a patient, we assume a delay equal to the remainder of these ten minutes, plus the chute time, as follows:

$$delay = max(arrival\ time\ at\ hospital - simtime + 10\ minutes;\ 0) + chute\ time \qquad (4.2)$$

Note, however, that we assume that ambulances will only be requested to accelerate its transfer process for urgent requests (A1 or A2). For non-urgent transports (B1/B2) ambulances are only able to depart after the complete transfer time has passed.

If there are waiting A1/A2 requests, this implies that the ambulance that just arrived at the hospital is the only ambulance available for dispatch. Therefore, no relocations need to be determined and initiated after dispatch of the ambulance to such a waiting request, since by definition there are no remaining free ambulances. On the other hand, if the ambulance that just arrived at the hospital resulted in enough ambulances being available for a waiting non-urgent transport to be dispatched, relocations should be determined. Even though the number of ambulances remained constant, the ambulance at the hospital is not necessarily the one that was dispatched to the B1/B2 request, meaning that relocations may be required.

Pseudocode 5: Handling an Arrival at hospital event

```
1: Update ambulance attributes
2: if there are requests in waiting list for A1/A2 requests then
3:     Dispatch the ambulance to longest waiting A1/A2 request
4:     Schedule the Arrival at request location event at simtime + driving time + delay(req, ambu)
5: else if there are requests in waiting list for B1/B2 requests then
6:     if there are ambulances available for dispatch to B1/B2 requests then
7:         Determine ambulance disp to dispatch                          ▷ Section 4.1.13
8:         Dispatch the selected ambulance to longest waiting B1/B2 request
9:         Schedule the Arrival at request location event at simtime + driving time + delay(req, disp)
10:        Determine and initiate relocations of remaining free ambulances  ▷ Section 4.1.12
11: if ambulance was not dispatched to a waiting request then
12:    Schedule the Service completion event at simtime + transfer time_req
```

4.1.8 Handling a *Service completion* event

If the first event of the ordered event queue is of the type *Service completion*, attributes of the concerned ambulance, such as its status, are updated to reflect this. While an ambulance completing service at a hospital was already available for dispatch during its transfer time since the corresponding *Arrival at hospital* event, an ambulance completing service at the request location becomes available for dispatch at this point. Similar to an ambulance becoming available for dispatch at a hospital, it is checked whether there are any requests waiting for dispatch due to no (not enough) ambulances being available at the time of its arrival. Here, priority is given to urgent requests (A1 or A2) over non-urgent transports (B1/B2).

Pseudocode 6: Handling a Service completion event

```
1: Update ambulance attributes
2: if there are requests in waiting list for A1/A2 requests then
3:     Dispatch the ambulance to longest waiting A1/A2 request
4:     Schedule the Arrival at request location event at simtime + driving time + delay(inc, ambu)
5: else if there are requests in waiting list for B1/B2 requests then
6:     if there are ambulances available for dispatch to B1/B2 requests then
7:         Determine ambulance disp to dispatch                          ▷ Section 4.1.13
8:         Dispatch the selected ambulance to longest waiting B1/B2 request
9:         Schedule the Arrival at request location event at simtime + driving time + delay(inc, disp)
10: if an ambulance was dispatched to a B1/B2 request or no ambulances were dispatched then
11:    Determine and initiate relocations of remaining free ambulances  ▷ Section 4.1.12
```

4.1.9 Handling an *Arrival at station* event

If the first event of the ordered event queue is of the type *Arrival at station*, attributes of the concerned ambulance, such as its status and origin, are updated to reflect this.

Pseudocode 7: Handling an Arrival at station event

```
1: Update ambulance attributes
```

4.1.10 Handling a *Start shifts* event

If the first event of the ordered event queue is of the type *Start shifts*, ambulance instances are created according to the shift roster. The weekly shift roster contains, for each day of the week, the times at which shifts start, and for each time the number of ambulances starting its shift at each base station. Each shift in the BZO region lasts eight hours, which is why, for each ambulance starting its shift, an *End shift* event can be scheduled for the current time plus eight hours. Lastly, a new *Start shifts* event is scheduled according to the shift roster.

Pseudocode 8: Handling a Start shifts event

```
1: for each shift to start according to the shift roster do
2:     Create new ambulance object at indicated base station
3:     Schedule the End shift event at simtime + 8 hours
4: Schedule event of the next round of shift starts according to the shift roster
```

4.1.11 Handling an *End shift* event

Ideally, an ambulance ends its shift eight hours after it started. However, ambulances can only end its shift at the base station where it started it. Therefore, it is checked whether the ambulance corresponding to the *End shift* event is positioned at its base station. If this is the case, the ambulance object is removed from the list of available ambulances and will play no further role in the simulation. However, if the ambulance is not positioned at its base station, it is checked at what time the ambulance's next event is scheduled to occur, which is the earliest moment at which the ambulance might be at its base station. A new *End shift* event is scheduled to occur just after this event.

Due to the dispatch policy avoiding dispatch of ambulances with an almost ending shift (see Section 4.1.13), as well as the relocation policy sending ambulances back to its base station if its shift is almost over (see Section 4.1.12), overtime is being limited.

Pseudocode 9: Handling a End shift event

```
1: if ambu is at its base station then
2:     Remove ambulance object from available ambulances
3: else
4:     Schedule new End shift event just after ambu's next event
```

4.1.12 Relocation policy

While the relocation of ambulances to restore coverage of the region after the number of available ambulances changes, is not the focus of this research, the current relocation policy should be incorporated in this simulation to be able to capture the ambulance dynamics as accurately as possible. During discussions with a group of dispatch agents they stated that relocations in the BZO region are generally done according to a so-called *Compliance table*. A compliance table dictates at which stations the available ambulances should be positioned, depending on the number of available ambulances. The region can either be 'in compliance' if all available ambulances are located according to this table, or 'out of compliance' if they are not. Whenever the system is out of compliance, relocation movements are required to bring the system back into compliance again (Theeuwes, 2018). Table 4.1 shows the compliance table that BZO's dispatch agents adhere to, which is partially nested. The nested property entails that the location set where ambulances should be located given m available ambulances is a subset of the set corresponding to n available ambulances if $m < n$. The (partial) nested property limits the number of required relocations (Sudtachat, Mayorga & Mclay, 2016).

Table 4.1: Compliance table used by BZO agents to determine relocations, incl. four-digit postal codes

Nr. / Avail. Ambulances									
1	A (5644)								
2	A (5644)	H (5705)							
3	A (5644)	H (5705)	EN (5627)						
4	E (5521)	H (5705)	EN (5627)	M (6026)					
5	E (5521)	H (5705)	EN (5627)	M (6026)	EC (5611)				
6	E (5521)	H (5705)	EN (5627)	M (6026)	EC (5611)	D (5751)			
7	E (5521)	H (5705)	EN (5627)	M (6026)	EC (5611)	D (5751)	V (5555)		
8	E (5521)	H (5705)	EN (5627)	M (6026)	EC (5611)	D (5751)	V (5555)	B (5531)	
9	E (5521)	H (5705)	EN (5627)	M (6026)	EC (5611)	D (5751)	V (5555)	B (5531)	L (5735)

*A: Aalsterweg, H: Helmond, EN: Eindhoven Noord, E: Eersel, M: Maarheeze, EC: Eindhoven Centrum, D: Deurne, V: Valkenswaard, B: Bladel, L: Lieshout

This table is used to determine the stations that should be covered each time the number of available ambulances changes, i.e. in case of a dispatch or service completion. Since in case of a large number of available ambulances, the location of the umpteenth available ambulance is less important, no relocations are initiated if there are more than nine ambulances available. If in such a situation a service completion occurs, this ambulance is sent back to its base station. Ambulances that are busy transferring a patient are also included in the determination of relocations, since they will become available in the (very) near future. However, relocations are not actually initiated for these ambulances (yet). Lastly, before determining and initiating relocations, ambulances whose shift is almost ending are sent to its base station, if it is not already (on its way) there, and excluded from the ambulances to be relocated.

While the compliance table dictates at which stations the available ambulances should be positioned, it does not specify how this configuration should be reached, i.e. which relocation movements should be initiated. Through discussions with BZO dispatch agents it was found that they wish to achieve compliance as quickly as possible, such that the fraction of time during which the system is in compliance is maximized. Additionally, dispatch agents wish to limit the number of relocations, since this causes disturbance to ambulance crews. Therefore, ambulances that are at, or on its way to, one of the stations that should be covered as specified

by the compliance table, are not rerouted. The remaining available ambulances are allocated to the remaining, uncovered, ambulance stations in such a way that the maximum driving time is minimized. We define this relocation problem as follows:

$$\begin{aligned} \text{minimize} \quad & \max_{i,j} d_{ij}\, x_{ij} & \\ \text{subject to} \quad & \sum_{j=1}^{n} x_{ij} = 1, & \forall i \in \mathcal{I} \\ & \sum_{i=1}^{n} x_{ij} = 1, & \forall j \in \mathcal{J} \\ & x_{ij} \in \{0,1\} & \forall i \in \mathcal{I}, j \in \mathcal{J} \end{aligned}$$

Here, $\mathcal{I}$ denotes the set of remaining ambulances to be relocated, and $\mathcal{J}$ denotes the set of remaining, uncovered, stations that need to be occupied according to the compliance table level corresponding to the total number of available ambulances. d_{ij} is the time it takes for ambulance $i \in \mathcal{I}$ to arrive at station $j \in \mathcal{J}$ and x_{ij} denotes whether ambulance $i \in \mathcal{I}$ is relocated to station $j \in \mathcal{J}$. This problem is a Linear Bottleneck Assignment Problem (LBAP). We have implemented a solving method for such problems based on Garfinkel's *Threshold algorithm* as described by Burkard, Dell'Amico and Martello (2009). After solving the LBAP, relocations are initiated for the available ambulances according to the solution.

Pseudocode 10: Determining and initiating relocations

```
1: for each available ambulance ambu do
2:     if the shift of ambu should (have) end(ed) within 30 minutes then
3:         if ambu is not at -or on its way to- its base station then
4:             Send ambu to base station
5:             Schedule Arrival atstation event at simtime + driving time
6:     else
7:         Add ambu to set I of ambulances available for relocation
8: Add all ambulances that are busy transferring a patient at a hospital to set I
9: if |I| ≤ 9 then
10:    Look up set of stations J to be covered in compliance table on level |I|
11:    for each ambulance i ∈ I do
12:        if ambulance i is already at, or on its way to, a station j to be covered ∈ J then
13:            Remove ambulance i from set I
14:            Remove station j from set J
15:    for each remaining ambulance i ∈ I do
16:        for each remaining station, j ∈ J do
17:            Determine driving time between i and j
18:    Solve the LBAP
19:    for each x_ij > 0 in LBAP solution do
20:        if ambulance i is available (not busy transferring a patient at a hospital then
21:            Send ambulance i to station j
22:            Schedule Arrival at station event at simtime + driving time
23: else if an ambulance just completed service then
24:    Send ambulance to its base station
25:    Schedule Arrival at station event at simtime + driving time
```

4.1.13 Dispatch policy

Naturally, the dispatch policy that is implemented in this simulation is the result of the formalization effort of Chapter 3. First, however, it should determined which ambulances are available for dispatch to the request at hand, which depends on the request's urgency, as outlined in Section 4.1.5. Subsequently, for each available ambulance, the driving time between its current location and the location of the request at hand is determined, after which relevant penalties are added to it according to the Penalty model of Algorithm 2. Note, however, that since ambulances from neighbouring EMS regions are excluded from the simulation to avoid unnecessary complexity, that lines 7 - 10 of this algorithm are not applicable. After computing the penalized driving time of each of the ambulances available for dispatch to the request at hand, the ambulance with the smallest value is identified and returned to be dispatched.

Pseudocode 11: Determining which ambulance to dispatch (current; based on Alg. 2)

```
1: Determine which ambulances are available for dispatch to req's urgency
2: for each ambulance ambu available for dispatch to req do
3:     Determine driving time between location of ambu and req
4:     Add relevant penalties to driving time to obtain penalized driving time for ambu   ▷ Alg. 2
5: Return ambu for which penalized driving time is smallest
```

An important note that should be made, is that in the determination of the driving time between an ambulance's current location and the request location, this current location should be estimated. The simulation does not include route planning and merely uses driving times between postal codes to schedule arrival events. If an ambulance is standing still, i.e. its origin is equal to its destination, its current location is intuitive. However, if an ambulance is moving, e.g. during a relocation, its location should be estimated at the moment the driving time to a request is determined. For this purpose, we assume that ambulances move 'as the crow flies' and with a constant speed. Using this assumption, the current location of a moving ambulance can be estimated through (the coordinates of its) origin and destination, combined with its driving time, and the time that has passed since it departed from its origin, by using Pythagoras. Subsequently, the estimated coordinates of the ambulance's current location are mapped onto a postal code. This procedure of determining an ambulance's current location is used each time the driving time to a given destination is determined, which includes during the dispatch and during the relocation procedure.

4.1.14 Potential enhancements to the dispatch policy

To evaluate the four enhancements to the current dispatching policy, as listed in Section 3.4, individually, as well as combined with each other, four switches were implemented.

Consistent redispatching
In order to evaluate the performance improvement potential of *consistent redispatching*, i.e. always dispatching an ambulance that is currently on its way to a less- or non-urgent (A2 or B1/B2) request if this is the best dispatch option for a highly urgent (A1) request, simply the penalty on redispatches is removed from the penalty model representing the current dispatch policy.

Pseudocode 12: Consistent redispatching: to replace line 5 and 6 in Alg. 2

```
1: if ambulance is on its way to a less urgent request then
2:     if consistent redispatching switch is on then
3:         Penalty_i = penalty_i + 0 minutes
4:     else
5:         Penalty_i = penalty_i + 4 minutes
```

Reevaluation of dispatch decisions

To determine the potential of reevaluating dispatch decisions upon service completion of an ambulance, the expected response time of the recently freed ambulance to each active request, for which the currently dispatched ambulance has not arrived yet, has to be computed. There might, however, be multiple requests to which the recently freed ambulance is expected to arrive earlier than the currently dispatched ambulance. Therefore, it needs to be established which response time improvements are preferred, such that it can be decided which dispatch decision is actually revised.

Due to our focus on improving the *on-time* performance of A1 requests, i.e. with a response time under the national target of fifteen minutes, priority will be given to requests to which the currently dispatched ambulance will arrive too late, while the recently freed ambulance is able to arrive on-time (*prio 1*). If there are multiple of such requests, priority is given to the biggest absolute response time improvement. If there are no such requests, the same is checked for A2 requests (*prio 2*). Lastly, if there are neither A1, nor A2, requests for which a reevaluated dispatch decision results in an ambulance arriving on-time rather than too late, it is checked whether there are reevaluated dispatch decisions possible which will significantly improve ($>$ one minute) on the response time of an A1 request (*prio 3*). While such a revised dispatch decision does not directly improve the fraction of on-time requests, it does reduce the mean response time, which in turn leads to increased available capacity.

The reevaluation of dispatch decisions should be done upon service completion of an ambulance. However, if there are urgent requests waiting for an ambulance to be dispatched due to no ambulances being available, dispatching the recently freed ambulance to such a waiting request is, naturally, preferred over improving the response time of requests to which an ambulance has already been dispatched.

Pseudocode 13: Reevaluation of dispatch decisions: to be inserted between line 4 & 5 in Pseudocode 6

```
1: if reevaluation switch is on then
2:     for each urgent (A1/A2) request req to which no ambulance has arrived yet do
3:         Determine expected response time of recently freed ambulance to req
4:         if reevaluating dispatch decision results in RT improvement then
5:             if result of reevaluation option is of higher priority than best option found thus far then
6:                 Set req as best reevaluation option found thus far
7:     if a reevaluation option was found then
8:         Dispatch the recently freed ambulance to request req
9:         Schedule the Arrival at request location event at simtime + drivingtime + delay(req, ambu)
10:        Schedule Service completion event at simtime for formerly dispatched ambulance
```

Minimum coverage reduction dispatching

To evaluate the potential of dispatching the ambulance that causes minimum coverage reduction among those that are able to arrive on-time, the single coverage measure, as illustrated in Figure 3.6, is implemented. Since in case of A1 requests every minute counts, this dispatch policy add-on is only applied to A2 requests. After computing the penalized driving time for each ambulance that is available for dispatch to the A2 request at hand, it is identified which of the ambulances are expected to arrive on-time. Subsequently, the percentual coverage reduction resulting from dispatching each of these ambulances is determined. To ensure coverage reduction is not limited at too high cost (i.e. increased response time), the ambulance that can reach the request location fastest is selected among those for which dispatching results in a coverage reduction of less than five percent from the minimum possible value.

Pseudocode 14: Minimum coverage reduction dispatching: to replace line 5 in Pseudocode 11

```
1: if minimum coverage reduction switch is on then
2:     if urgency of req is A2 then
3:         for each ambulance a that is expected to arrive on-time given its penalized driving time do
4:             Determine percentual coverage reduction (PCR_a) resulting from dispatching a
5:         PCR_min ← min_{a∈A} PCR_a
6:         Return a with smallest penalized driving time among those with PCR_a ≤ PCR_min + 5%
7:     else if urgency of reqc is not A2 then
8:         Return ambu for which penalized driving time is smallest
9: else
10:    Return ambu for which penalized driving time is smallest
```

Postpone A2 dispatches

Lastly, to evaluate the potential of postponing A2 dispatches in case of limited capacity, events of the type *Arrival at request location* are rescheduled upon occurrence for A2 requests if conditions are met. These conditions include limited capacity of ambulances, which is set at ≤ 3 based on discussions with dispatch agents, and the A2 request not having exceeded its response time threshold yet. This implementation represents immediately dispatching ambulances to A2 requests, but staying available for redispatch to requests of higher urgency until the response time is about to exceed thirty minutes.

Pseudocode 15: Postpone A2 dispatches (a): to insert between lines 1 & 2 in Pseudocode 4

```
1: if postpone A2 switch is on then
2:     if there are less than four ambulances available then
3:         if req is of urgency A2 and arrived less than 30 min. ago then
4:             Schedule new Arrival at request location event at arrival time of req + 30 min.
5:             Skip rest of event handling
```

As soon as capacity of ambulances is not limited anymore, postponed ambulances do not have to be available for redispatch to requests of higher urgency anymore. Therefore, after the two events at which available ambulance capacity might be increased, i.e. *Arrival at hospital* and *Service completion* events, it is evaluated whether postponed ambulances should be allowed to start treatment, in which case a new *Arrival at request location* event is scheduled.

Pseudocode 16: Postpone A2 dispatches (b): to add at end of Pseudocode 5 and 6

```
1: if postpone A2 switch is on then
2:     if there are (now) four or more ambulances available then
3:         for each postponed A2 request do
4:             Remove next Arrival at request location event from event queue
5:             Schedule new Arrival at request location event at simtime
```

4.2 Results

After implementation of the simulation in Java, it can be used to evaluate the improvement potential of the listed enhancements to the current dispatch policy. First, however, performance of the simulation in the base scenario, representing current practices, is compared to actual values in Section 4.2.1 and a theoretic upper bound on the on-time performance of A1 requests is determined analytically in Section 4.2.2. Lastly, Section 4.2.3 provides insight into performance under different (combinations of) potential enhancements of the dispatch process. All results are based on fifty simulation runs of 52 weeks (run time: four seconds per run), resulting in 95% confidence intervals for the on-time performance with a half width of less than 0.05%. Confidence intervals can be found in Appendix B.

4.2.1 Base scenario

To evaluate the extent to which the simulation captures actual dynamics of the BZO region, the performance measures resulting from simulating the base scenario, representing current dispatch practices, are compared to the actual values of these measures, realized between January 2017 and December 2018. Table 4.2 shows that the simulation results in somewhat better performance for A1 requests and similar performance for A2 requests compared to realized values. There are a number of reasons for the simulation outperforming reality. One main reason is the fact that the ambulance station Eindhoven Noord only moved there from (the less optimal location in) Best in June 2018, while this current set of ambulance stations is assumed for the entire duration of the simulation. Furthermore, as the relocation process has not been formally captured, its implementation in the simulation is based on (extensive) discussions with dispatch agents. The implemented policy results in more frequent relocations than observed in practice, mainly due to relocations being determined and initiated after each change in the number of available ambulances, while in reality multiple of such changes may occur in quick succession with no time in between for initiating relocations. Lastly, while in practice variations arise in dispatch and relocation decisions due to human judgment and differences between dispatch agents, in the simulation both the dispatch and relocation policy are applied consistently. Following the well-known decision making theory of Bowman (1963), the elimination of variance in decision making generally results in improved performance.

Table 4.2: Realized and simulated values performance measures under current dispatch policy

	A1 requests		**A2 requests**	
	On-time (%)	**Mean RT (min:sec)**	**On-time (%)**	**Mean RT (min:sec)**
Realized	92.13	9:33	97.10	14:32
Simulated	93.63	9:02	97.07	13:38

A closer look into the results of the simulation shows that the geographic distribution of urgent requests (A1 & A2) in the simulation greatly resembles reality, see Figure 4.5.

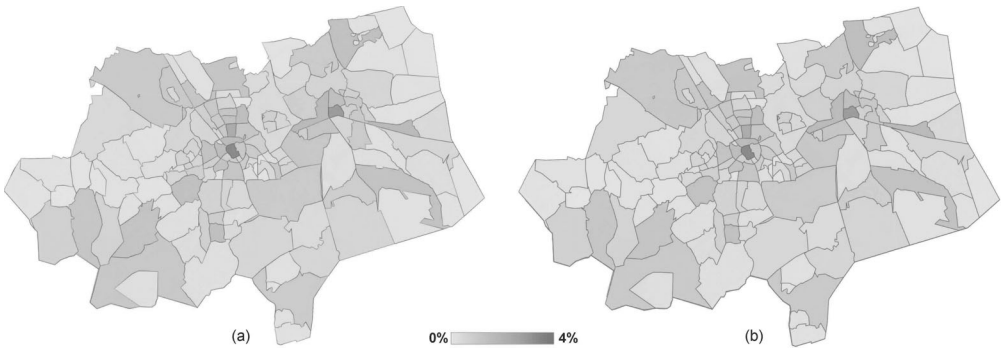

Figure 4.5: Geographic distribution of urgent requests; realized (a) and simulated (b)

In terms of the fraction of requests with a response time (RT) below its target, Figure 4.6 shows a tendency of high performance towards the more centrally located postal codes for both realized, as well as simulated performance. However, this pattern is significantly more apparent in case of the simulated performance, which is mainly driven by four factors. Firstly, the fact that interaction with neighbouring EMS regions is excluded from the simulation is likely to negatively affect performance of postal codes near the region's border. Secondly, due to the use of deterministic driving times in the simulation, variation in performance for a given postal code is more limited than in reality. Thirdly, due to the consistent application of the formalized current dispatch process, the resulting performance is also expected to follow more consistent patterns, since this neglects any variation between dispatch agents that is likely to affect performance in practice. Lastly, while the realized geographic distribution of performance in Figure 4.6a is based on (one 'run' of) two years of data, the simulated distribution in Figure 4.6b is based on fifty one-year runs, which makes it intuitive that performance converges more strongly.

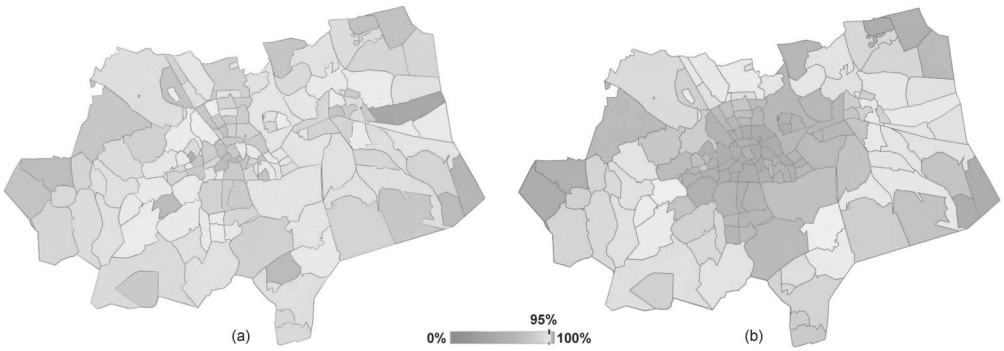

Figure 4.6: Fraction of urgent requests with RT below target; realized (a) and simulated (b)

Despite the simulation resulting in somewhat higher performance and more pronounced geographic performance patterns compared to practice, we expect that the simulation resembles

reality close enough such that the potential of enhancing the dispatch process, compared to the current dispatch policy, can be evaluated.

4.2.2 Analytic upper bound on performance

Before evaluating the effect of the formulated enhancements to the dispatch process, an analytic upper bound on the main performance measure, the fraction of highly urgent (A1) requests that is on-time, is determined. Such an upper bound gives us an idea of the problem at hand by quantifying the extent to which the main performance measure is able to improve. Consider the highly theoretic scenario in which ambulance occupancy is zero, i.e. ambulances that are on shift according to the weekly shift roster are always available. In this case, ambulances are positioned at ambulance stations according to the compliance table, Table 4.1. Refer to Appendix C for an overview of the postal codes that can be reached within a response time of fifteen minutes from each station. Then, the number of available ambulances throughout the week is shown in blue in Figure 4.7. Given the time-dependent distribution of A1 request locations, deducted from the total set of request samples, it can be computed what fraction of highly urgent (A1) requests is expected to be served on-time, i.e. within a response time of fifteen minutes. This expected fraction throughout a week is shown in Figure 4.7 in orange. An upper bound on the on-time performance of A1 requests in this highly theoretical scenario can be computed by computing the weighted average of the expected on-time performance in each time period $p \in \mathcal{P}$ (orange line), where the weight of each time period reflects the expected number of A1 requests in this time period p (see Figure 4.3. This computation results in an expected on-time performance of A1 requests of 99.18%. So, when assuming all ambulances to be available at all times, there is quite some room for improvement in terms of the main performance measure.

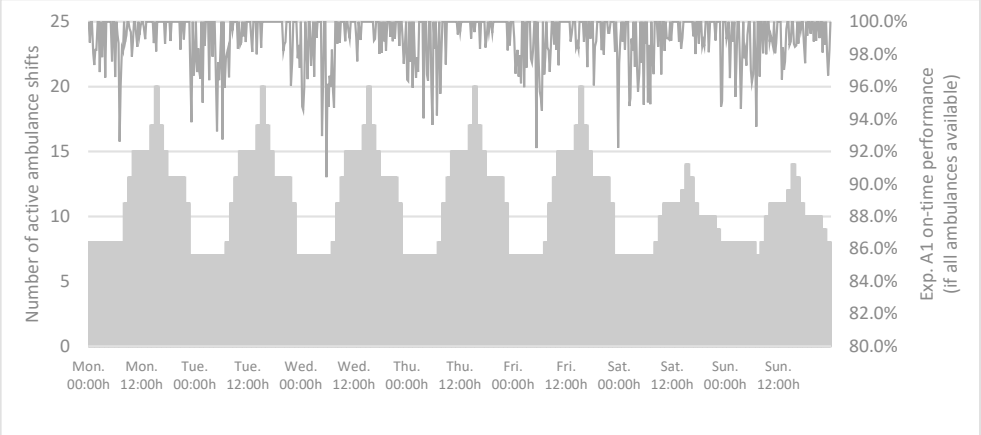

Figure 4.7: The number of active ambulance shifts throughout the week, and the resulting expected A1 on-time performance (assuming all ambulances to be available at all times)

While this analytic upper bound does not depend on the applied dispatch policy, it is based on the assumption that all ambulances that are on shift are available at all times. The simulation can be used to estimate the fraction of time each possible number of available ambulances holds, but this requires the assumption of a dispatch policy. Consider, for example, the

same theoretic scenario in which all available ambulances are always positioned at the station as specified by the compliance table, i.e. relocations are instantaneous such that available capacity is always optimally positioned. Table 4.3 shows, for each number of free ambulances, the fraction of A1 requests that can be reached on-time, i.e. which location can be reached within a response time of fifteen minutes from the stations that are occupied given the number of available ambulances. Furthermore, the table shows the fraction of time the number of free ambulances is actually equal to each of these, obtained from the simulation and assuming dispatching is done according to the captured dispatch policy, i.e. the penalty-based model in Algorithm 2. Multiplying the expected on-time performance for A1 requests in this theoretic scenario with the fraction of time each number of available ambulances holds, results in an expected on-time performance of 97.28%. This number can be regarded as a stricter theoretical upper bound for this performance measure. However, recall that this stricter upper bound depends on the applied dispatch policy. The enhancements to the dispatch policy to be evaluated are likely to shift this stricter theoretical upper bound.

Table 4.3: Expected A1 on-time performance for each number of available ambulances in a theoretical scenario assuming instantaneous relocations

	No. free ambulances									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9+**
% A1 on-time	0.00%	62.82%	82.51%	87.58%	96.09%	97.10%	97.89%	97.89%	98.92%	99.57%
% of time	0.33%	0.58%	1.47%	3.50%	7.04%	10.86%	14.57%	14.39%	10.23%	37.02%

4.2.3 Potential enhancements dispatch process

Table 4.4 shows the resulting performance measures for each (combination of) enhancement(s) to the dispatch process. Besides the main performance measures relating to the response time of urgent requests, the right side of Table 4.4 provides further insight into the effect of each enhancement from which conclusions regarding the effect on ambulance crew disturbance can be deducted. Appendix B provides the 95%-confidence intervals of all these measures.

From the effects on performance caused by each dispatch enhancement individually, it can be concluded that *consistent redispatching* is most beneficial to on-time performance of A1 requests, resulting in a gain of 0.43 percent points (pp). However, for the on-time performance of A2 requests this adaption to the dispatch process is most detrimental. This detrimental effect is mostly caused by the fact that an ambulance is redispatched regardless of whether an alternative ambulance is available for dispatch to the original request, and whether this ambulance is able to arrive on-time. The elimination of the artificial driving time penalty on redispatches leads to almost 2.5 times more dispatches. While under the current dispatch policy on average 3.9 redispatches are initiated each day, this number increases to a little over 9 redispatches per day in case of consistent redispatching. Given the number of shifts on an average day, this implies that an ambulance crew is only redispatched once every four shifts, which does not seem excessive.

Furthermore, *reevaluation* of active dispatch decisions upon service completion of an ambulance also leads to a significant improvement of the fraction of A1 requests that is served on-time, namely 0.41 pp. Not only is this the only enhancement to the dispatch process, of the four evaluated, that does not improve response time performance for A1 requests at the expense of performance for A2 requests, this measure is even improved through prio two

Table 4.4: Resulting performance for potential dispatch enhancements

Redispatch	Reevaluation	Coverage red.	Postpone A2	A1 requests		A2 requests		Nr. redisp./yr	Nr. reeval. P1/yr	Nr. reeval. P2/yr	Nr. reeval. P3/yr	Nr. postponed/yr
				On-time (%)	Mean RT (min:sec)	On-time (%)	Mean RT (min:sec)					
Base				93.63	9:02	97.07	13:38	1425				
x				94.06	8:55	96.15	14:04	3293				
	x			94.04	8:56	97.50	13:34	1413	112	76	635	
		x		93.78	9:01	97.07	14:36	1407				
			x	93.67	9:01	96.39	14:29	1560				1339
x	x			94.40	8:50	96.73	13:58	3269	99	95	578	
x		x		94.17	8:54	96.06	15:04	3361				
x			x	94.08	8:54	95.60	14:53	3425				1286
	x	x		94.11	8:55	97.49	14:32	1390	109	73	621	
	x		x	94.04	8:55	96.82	14:25	1555	108	79	627	1352
		x	x	93.78	9:00	96.39	15:22	1540				1305
x	x	x		94.52	8:49	96.52	14:59	3346	96	90	577	
x	x		x	94.41	8:50	96.11	14:48	3406	96	94	572	1308
x		x	x	94.20	8:53	95.53	15:46	3460				1254
	x	x	x	94.12	8:55	96.81	15:18	1530	105	78	620	1317
x	x	x	x	94.53	8:49	95.96	15:42	3443	94	91	571	1258

*P1, P2, and P3 refer to priority one, two, and three of reevaluation, i.e. A1 request from late to on-time, A2 request from late to one time, and RT improvement of A1 request respectively.

reevaluations with 0.43 percent points. From the number of reevaluations leading to the recently freed ambulance being dispatched, and thus for the currently dispatched ambulance to be redirected, it can be deducted that such a decision is made on average 2.26 times per day. The disturbance to the ambulance crew of this number of redirections is likely to be quite limited.

The *minimum coverage reduction* dispatching enhancement results in some improvement (0.15 pp) to the on-time performance of A1 requests, while keeping this measure for A2 requests approximately at the same level. Naturally, the mean response time to A2 requests does increase, since, of those ambulances able to reach the A2 request on-time, no longer the one which is expected to reach the request location quickest is dispatched necessarily. The *postponing* of dispatches to A2 requests in case of scarce availability, however, does not result in significant improvement to the on-time performance of A1 requests, i.e. the 95%-confidence interval overlaps with that of the base scenario. A likely reason for the limited impact of postponing of A2 requests in case of scarce availability is the fact that these ambulances are not taken into account in positioning the remaining available capacity through relocations. This makes the potential impact of the postponed ambulance very dependent on the location of the concerned A2 request relative to the position of the remaining (scarce) capacity. This is underlined by the fact that the number of redispatches, compared to the base scenario, only

increases by 136 (per year), meaning that of the 1339 postponed ambulances, only 10.1% was actually redispatched to a more urgent request, which is why they were postponed in the first place. Furthermore, this enhancement to the dispatch process is quite detrimental to A2 on-time performance, for similar reasons as in case of the *consistent dispatch* enhancement.

Besides evaluating the effect on performance of each enhancement individually, combinations were also simulated. Adding both the *consistent redispatch* and *reevaluation* enhancement to current dispatch practices yields the largest improvement of the fraction of on-time A1 requests. It can be seen that the performance gain of both of these enhancements individually is quite complementary, as combining these enhancements leads to an A1 on-time performance gain of almost the sum of the individuals performance gains. Furthermore, the mean response time of A1 requests is lower than that of both enhancements individually. Lastly, the fact that the *reevaluation* enhancement is beneficial to the performance of A2 requests mitigates part of the detrimental effect of the *consistent redispatch* enhancement. Combining these two enhancements, however, also leads to a larger number of redirections (resulting from either being redispatched or from a reevaluated dispatch decision), which may cause disturbance to ambulance crews. Yet with an average of approximately eleven redirections per day, this disturbance is likely to be outweighed by the resulting performance gain.

The effect of combining the *minimal coverage reduction* and *postponing A2 requests* enhancements with other(s) is in line with their respective individual performance gain. While addition of the *minimal coverage reduction* enhancement leads to some gain in on-time performance of A1 requests, with limited effects on the on-time performance of A2 requests but an increase in their mean response time, *postponing A2 requests* lead to marginal, often insignificant, improvements to performance of A1 requests, while negatively affecting performance of A2 requests. The best performance for A1 requests, both in terms of *on-time* performance and the *mean response time*, is obtained by combining all four enhancements. The resulting dispatch policy, however, also results in the worst performance for A2 requests, as well as the largest degree of disturbance to ambulance crews.

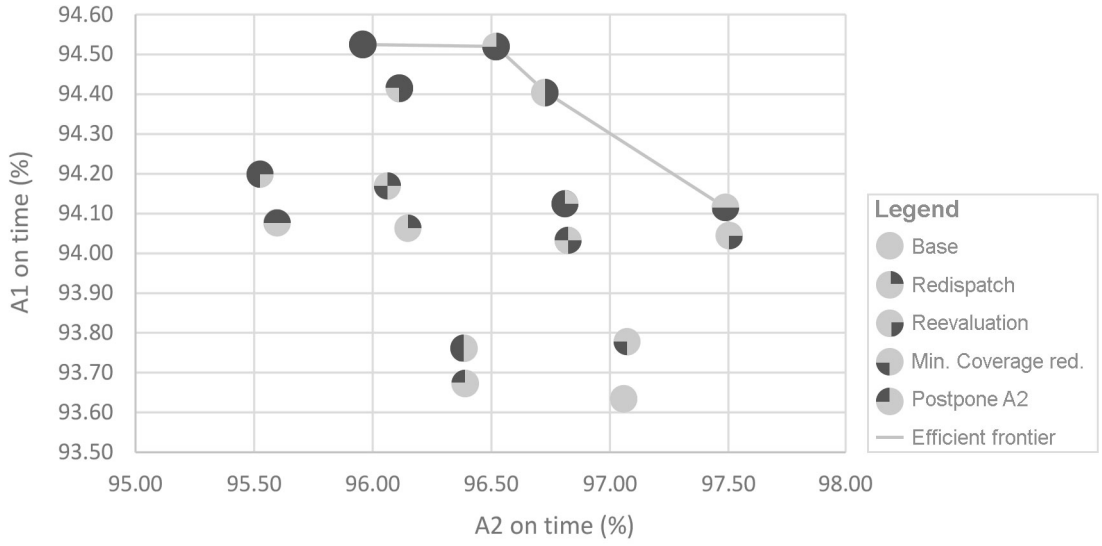

Figure 4.8: Visual representation of performance of (combinations of) dispatch policy enhancement(s)

Ideally, we wish to improve the *A1 on-time* performance as much as possible. However, in selecting the best dispatch policy, the cost, i.e. the reduction in *A2 on-time* performance should also be taken into account. Furthermore, while each of the four possible enhancements are relatively easy to incorporate in the current dispatch process, each additional enhancement adds to dispatch agents' cognitive load, which might inhibit their performance. So, even though Table 4.4 shows that an additional enhancement always leads to better or equal *A1 on-time* performance, the performance gain should outweigh the additional strain on the working memory of dispatch agents. In other words, the selection of the best dispatch policy essentially encompasses a trade off between the gain in *A1 on-time* performance, the reduction of *A2 on-time* performance, and the number of enhancements to be added to the current dispatch policy.

Figure 4.8 visually shows the elements of this trade-off. Concerning the on-time performance of both A1 and A2 requests, the efficient, or Pareto, frontier has been highlighted. The dispatch policies on this frontier are those that cannot be improved upon for either A1 or A2 on-time performance without deteriorating the other. It makes sense to select one of the dispatch policies on this frontier. Taking into account the fact that the primary focus of this research is on improving the *on-time* performance of highly urgent A1 requests, combined with the preference to limit the additional cognitive load imposed on dispatch agents, leads to selection of the ***consistent redispatch* and *reevaluation* enhancements** to the current dispatch policy as the preferred policy. Application of these enhancements is expected to result in an increase of on-time performance of A1 requests equal to 0.77 percent points, at the expense of a decrease of 0.33 percent points in on-time performance of A2 requests. Furthermore, mean response time is expected to decrease slightly for A1 requests, while it increases slightly for A2 requests. An expected number of 3269 ambulances will be redispatched on a yearly basis, constituting an increase of 129% compared to current practices, and approximately 99, 95, and 578 ambulances will be redirected as a result of reevaluated dispatch decisions of prio 1, 2 and 3 respectively. Hence, the total number of redirections, either due to being redispatched or reevaluated, is expected to be equal to 4041, which implies 11.1 redirections per day, or an ambulance being redirected once every three eight-hour shifts.

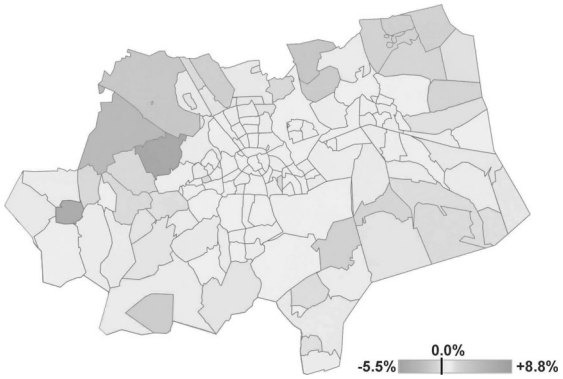

Figure 4.9: Absolute (percentage point) improvement of A1 on-time performance per postal code due to consistent redispatch and reevaluation enhancements to the current dispatch policy

The selected *consistent redispatch* and *reevaluation* enhancements to the current dispatch

policy both contribute to the extent to which the dispatch policy is dynamic. Both enhancements encourage dispatching an ambulance from 'the field', i.e. an ambulance that is not at a station. Therefore, as can be expected, the *consistent redispatching* and *reevaluation* enhancements to the current dispatch policy are especially beneficial for (postal code) areas that cannot be reached in time from any, or most, ambulance stations. Figure 4.9 shows the absolute difference in on-time performance of A1 requests for each postal code. Comparing this figure to Figure 4.6b shows that performance gain is greatest for those areas that perform worst under the current dispatch policy. It should, however, also be taken into account that lower initial performance in these postal code areas provided more room for improvement.

4.2.4 Result in perspective

To place the effect of the selected enhancements to the dispatch policy in perspective, as well as to demonstrate the potential of the simulation in the identification of ways to improve performance, the simulation is used to evaluate the effect of additional ambulance shifts. Using the simulation of the base scenario, an eight-hour shift will be added to the realized weekly shift roster (see Appendix A) iteratively through a greedy neighbourhood search. Here, in each iteration the neighbourhood is defined as the set of current weekly shift rosters with one additional eight-hour shift, where the additional shift is assumed to only be allowed to start at one of the moments shifts are currently starting (leading to a total of 49 options). Each additional shift is set to start (and thus end) at base station Helmond, since this is the base station ranking highest in the compliance table. The effect of the additional shift's base station on results is expected to be limited. Each iteration consists of fifty simulation runs of 52 weeks for each of the 49 possible shift rosters in the defined neighbourhood. Subsequently, the shift roster leading to the largest improvement of the on-time performance of A1 requests is selected greedily, after which the next iteration starts.

Table 4.5: Resulting performance for the iterative addition of eight-hour shifts

	A1 requests		**A2 requests**		
No./add. shifts	**On-time (%)**	**Mean RT (min:sec)**	**On-time (%)**	**Mean RT (min:sec)**	**Additional shift at**
Base	93.63	9:01	97.1	13:37	-
1	93.79	8:59	97.24	13:31	Sunday 07:00h
2	93.91	8:58	97.36	13:28	Saturday 07:00h
3	94.03	8:57	97.44	13:25	Saturday 14:00h
4	94.13	8:55	97.51	13:21	Saturday 23:00h
5	94.22	8:54	97.59	13:20	Friday 08:00h
6	94.31	8:53	97.66	13:16	Friday 23:00h
7	94.39	8:52	97.73	13:14	Saturday 09:00h
8	94.46	8:51	97.81	13:11	Sunday 08:00h

Table 4.5 shows the resulting performance measures for the addition of one to eight shifts. The greedy addition of ambulance shifts led to shifts being added mostly during the weekend, which is mainly driven by the lack of non-urgent transports (B1/B2) during these days, which allow for more flexible utilization of capacity during weekdays. It can be seen that, in terms of the on-time performance of A1 requests, the addition of the two selected enhancements to the current

dispatch process is equivalent to the addition of over seven extra shifts on a weekly basis. Naturally, while the enhancements to the dispatch process result in improved performance for A1 requests at the expense of performance for A2 requests, the addition of ambulance shifts improves both measures. However, while the addition of ambulance shifts is quite costly, time consuming, and difficult due to the severe shortage of medical personnel, the selected enhancements are merely process adaptions, which are essentially free and instantaneous. Furthermore, adding extra ambulance shifts and the enhancements to the dispatch process are not mutually exclusive. For example, the addition of seven weekly ambulance shifts as indicated in Table 4.5, combined with the selected enhancements to the dispatch process, *consistent redispatching* and *reevaluation* result in a performance of 95.04% and 97.32% of requests served on-time for A1 and A2 requests respectively.

5 | Conclusion and further research

The objective of this thesis was to improve the on-time performance of A1 requests in the EMS region of Brabant-Zuidoost through improvement of the dispatch policy by building upon current practices. Hereto, current dispatch practices were captured, after which four potential enhancements to this process were formulated and evaluated using a realistic simulation. This chapter briefly summarizes the conclusions of these efforts in Section 5.1, after which recommendations for further research are discussed in Section 5.2.

5.1 Conclusion

We approached the development of an (alternative) ambulance dispatch policy by capturing current dispatch practices and using it as a practically relevant basis to build upon. While existing studies, aiming to improve performance through alternative dispatch policies, either alter the commonly-assumed 'closest-idle' dispatch policy or develop a dispatch policy from scratch, this thesis formally captured the way in which dispatch decisions are currently made with the goal of using this policy as a basis to build upon by extending it with additional or adapted decision rules. A combination of decision tree induction and a post-processing phase resulted in a formal model that is both concise and able to accurately predict current dispatch decisions. The resulting model enriches the commonly assumed closest-idle dispatch policy through the use of penalty values that reflect the risk associated with certain ambulance characteristics, such as its status, region and time until the end of its shift. Based on a combination of insights from the capturing efforts, discussions with dispatch agents, and available literature, four potential enhancements to the current dispatch policy were formulated: *consistently redispatching* ambulances to highly urgent (A1) requests, *reevaluating* dispatch decisions upon service completion of an ambulance, dispatching the ambulance resulting in *minimum coverage reduction*, and *postponing dispatches* to less urgent requests in case of limited ambulance availability.

Additionally, a realistic simulation was developed that is able to accurately capture the complex dynamics of a life size ambulance system to evaluate these potential enhancements to the current dispatch policy within a reasonable computation time. As discussed in Section 2.7.2, existing studies evaluating alternative dispatch policies generally resort to simplifying modeling choices and assumptions in the development of a simulation, mainly relating to the size of the problem and the dynamicity of request arrivals and characteristics. The developed simulation, however, is able to realistically deal with ambulance requests of multiple urgency levels (including non-urgent transports), dynamic ambulance capacity, realistic relocation decisions, and practical considerations such as the end of ambulance shifts, patient transfers that may be accelerated, and the distinction between base and standby ambulance stations. Furthermore, the simulation is able to accurately reflect ambulance request patterns through a request generation process that is both stochastic and dynamic in terms of the arrival times and request characteristics such as its urgency, location, treatment time, hospitalization prob-

ability, hospital location, transfer time, and the number of required ambulances. Lastly, the captured current dispatch process allowed us to be the first to evaluate alternative dispatch policies by comparing the simulated performance to that of a benchmark that resembles current practices. The development of this advanced simulation model, combined with the use of a practically relevant benchmark, allowed us to draw accurate conclusions regarding the expected effect of the proposed enhancements on actual performance in practice.

Using the developed simulation, we were able to quantify the effect of the four potential enhancements to the current dispatch policy. We showed that significant improvement to the on-time performance of highly urgent (A1) ambulance requests can be obtained by enhancing the dispatch process. More specifically, for the EMS region of Brabant-Zuidoost, this measure can be improved by 0.77 percent points through enhancing current dispatch practices by *consistently redispatching* ambulances that are on its way to a less urgent request to a more urgent request and *reevaluating* active dispatch decisions upon service completion of an ambulance, such that this ambulance can be dispatched instead if this leads to a significant improvement of response time. Given the theoretical upper bound on the on-time performance of A1 requests of 99.18%, the performance gain of 0.77 percent points closed at least 14% of the gap between current performance and this theoretical upper bound. Furthermore, results showed that this improvement to the on-time performance of highly urgent (A1) requests comes at the expense of a decrease of 0.33 percent points of the on-time performance of A2 requests, easily keeping it above its threshold target of 95% with a response time of less than thirty minutes.

Both enhancements encourage dispatching an ambulance from 'the field', i.e. an ambulance that is not at a station, making them especially beneficial for (postal code) areas that cannot be reached in time from any, or most, ambulances stations, such as those near the region borders. As a result of these two enhancements, ambulances are redirected on average once every three eight-hour shifts. Lastly, simulation results illustrated that an equivalent increase of the on-time performance of highly urgent requests would require the addition of over seven extra eight-hour shifts on a weekly basis. Adjusting the operational dispatch process to better utilize available capacity is both virtually free and instantaneous, contrary to capacity expansion through addition of ambulance shifts.

Although this work was focused on quantifying the potential of an adjusted dispatch policy specifically for the EMS region of Brabant-Zuidoost, we expect the resulting dispatch policy to yield similar effects in other EMS regions. Especially in Dutch EMS regions, where historic performance shows a similar opportunity to improve A1 performance at the expense of A2 performance and request intensity and density is comparable to that of the BZO region, a performance gain of similar magnitude is expected.

5.2 Further research suggestions

We suggest four directions for further research into this topic. First of all, this research specifically focused on improving the on-time performance of highly urgent (A1) requests in the EMS region of Brabant-Zuidoost. As mentioned, similar results are expected to hold for other EMS regions, especially those in the Netherlands. It would be interesting to find out to what extend the obtained results are directly applicable to other EMS regions. This might depend on the way ambulances are currently dispatched in such a region, but also on

region-specific characteristics, such as request intensity and density or ambulance capacity relative to demand. For example, in a region where a larger fraction of requests cannot be reached on-time from any of the ambulance stations, enhancements to the dispatch process which encourage dispatching ambulances from the field, such as *consistent redispatching* and *reevaluation* are expected to have more improvement potential.

Furthermore, after establishing that performance can be improved by adjusting the dispatch policy, such as we have done for Brabant-Zuidoost, we suggest to apply the two selected enhancements to the dispatch process in practice. These enhancements were designed as building blocks to complement, rather than replace, current dispatch practices, such that practical considerations are (still) incorporated in the final dispatch policy and it is in line with the way dispatch agents currently work. While this approach is expected to foster adoption in practice and allows for quick implementation without the need for (major) software changes, we recommend to conduct a pilot study to confirm the potential of these enhancements in practice before full adoption. A pilot study in which dispatch agents are asked to manually apply the two selected enhancements to the dispatch process, without supporting system adaptions, might not capture its full performance improvement potential. However, such a pilot study is an opportunity to evaluate the effect of these enhancement on both dispatch agents, in terms of increased cognitive load, and on ambulance crews, in terms of disturbance due to being redirected. While the expected magnitude of both of these aspects was taken into account in the selection of the two final enhancements, they were not explicitly researched. Furthermore, a pilot study allows for mapping the system adaptions that might help leverage the full improvement potential of the improved dispatch process, as well as limit additional workload for dispatch agents. For example, the process of relieving an ambulance of its current request assignment and redispatching it to a new request is currently quite time consuming, and might need some adjustments. Also, to limit the additional cognitive load for dispatch agents, relevant notifications might be needed, such as whenever an ambulance completes service and reevaluation of active dispatch decisions might lead to significant response time improvements.

This research revolved around alternative dispatch policies, assuming the current relocation policy to be a given. While findings in literature on dynamic ambulance management often show that smarter positioning of available ambulances at stations offers greater gains than advanced dispatching rules (e.g. Yue, Marla and Krishnan (2012)), these studies mostly assume no, or a very basic, relocation policy is in place initially. Furthermore, while the formulated enhancements to the dispatch process are complementary to current practices and therefore easily implementable, alternative relocation policies could entail more rigorous process changes. Nevertheless, it would be an interesting direction of research to consider alternative dispatch and relocation policies jointly. Even more so than for alternative dispatch policies, literature considering both processes jointly is very limited and proposed policies are often not evaluated in realistic(ally sized) EMS systems. Therefore, research into joint dispatch and relocation policies to further improve performance is suggested for further research.

Lastly, the approach that was used to formalize current dispatch practices, decision tree induction, is a rather basic machine learning technique. While the transparent and interpretable nature of the resulting decision tree allowed us to gain insights into the captured dispatch process such that it could be built upon, more advanced machine learning techniques exist to better capture the dispatch process. For example, deep learning methods, such as neural

networks, might be applied to better predict the ranking of ambulances to be dispatched, but do not allow insight into the resulting process. Nevertheless, the formalization of an operational decision process such as dispatching is an interesting research topic in itself and further research into alternative, more advanced, machine learning techniques to capture this process could verify whether it is possible to capture it more accurately than through the use of decision tree induction. Additionally, such techniques could be applied to capture the current relocation policy as well. In this thesis, the relocation policy was approximated through (extensive) discussions with dispatch agents, but an implementation of a more accurate representation of actual relocation decisions in the simulation would even further increase the extent to which it resembles reality.

References

Ambulancezorg Nederland. (2017, 11). Insight into the increase in the number of ambulance deployments. *Online publication*.

Ambulancezorg Nederland. (2018, 9). Tabellenboek ambulancezorg 2017. *Online publication*.

Ambulancezorg Nederland. (2019, 2). Arbeidsmarkt ambulancezorg. *Online publication*.

Andersson, T. & Värbrand, P. (2007). Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, *58*(2), 195–201.

Aringhieri, R., Bruni, M. E., Khodaparasti, S. & Van Essen, J. (2017). Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Computers & Operations Research*, *78*, 349–368.

Bélanger, V., Ruiz, A. & Soriano, P. (2018). Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*.

Boon, M., van Leeuwaarden, J., Mathijsen, B., van der Pol, J. & Resing, J. (2017, February). *Stochastic simulation (lecture notes).* Department of Mathematics and Computer Science, Eindhoven University of Technology.

Bowman, E. H. (1963). Consistency and optimality in managerial decision making. *Management Science*, *9*(2), 310–321.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and regression trees, wadsworth, california, usa, 1984* (Tech. Rep.). ISBN 0-534-98054-6.

Burkard, R. E., Dell'Amico, M. & Martello, S. (2009). *Assignment problems*. Springer.

Daskin, M. S. (1983). A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation science*, *17*(1), 48–70.

Donnot, B., Guyon, I., Schoenauer, M., Panciatici, P. & Marot, A. (2017). Introducing machine learning for power system operation support. *arXiv preprint arXiv:1709.09527*.

Galvao, R. D. & Morabito, R. (2008). Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. *International Transactions in Operational Research*, *15*(5), 525–549.

Ganzach, Y., Kluger, A. N. & Klayman, N. (2000). Making decisions from an interview: Expert measurement and mechanical combination. *Personnel Psychology*, *53*(1), 1–20.

Gendreau, M., Laporte, G. & Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing*, *27*(12), 1641–1653.

Han, J., Pei, J. & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Henderson, S. G. (2011). Operations research tools for addressing current challenges in emergency medical services. *Wiley Encyclopedia of Operations Research and Management Science*.

Isaac, A. & Sammut, C. (2003). Goal-directed learning to fly. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 258–265).

Jagtenberg, C., Bhulai, S. & van der Mei, R. (2017). Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health care management science*, *20*(4), 517–531.

Kim, M.-J. & Han, I. (2003). The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Systems with Applications*, *25*(4), 637–646.

Kommer, G. & Zwakhals, S. (2016). Referentiekader spreiding en beschikbaarheid ambulancezorg 2016.

Kotsiantis, S. B., Zaharakis, I. & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3–24.

Lafond, D., Roberge-Vallières, B., Vachon, F. & Tremblay, S. (2017). Judgment analysis in a dynamic multitask environment: Capturing nonlinear policies using decision trees. *Journal of Cognitive Engineering and Decision Making*, *11*(2), 122–135.

Lafond, D., Tremblay, S. & Banbury, S. (2013). Cognitive shadow: A policy capturing tool to support naturalistic decision making. In *Cognitive methods in situation awareness and decision support (cogsima), 2013 ieee international multi-disciplinary conference on* (pp. 139–142).

Lee, S. (2011). The role of preparedness in ambulance dispatching. *Journal of the Operational Research Society*, *62*(10), 1888–1897.

Lee, S. (2014). Role of parallelism in ambulance dispatching. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(8), 1113–1122.

Li, H. (2011). A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems*, *94*(10), 1854–1862.

Li, X. & Olafsson, S. (2005). Discovering dispatching rules using data mining. *Journal of Scheduling*, *8*(6), 515–527.

Lim, C. S., Mamat, R. & Braunl, T. (2011). Impact of ambulance dispatch policies on performance of emergency medical services. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 624–632.

Lin, F.-R., Hsieh, L.-S. & Chuang, F.-T. (2009). Discovering genres of online discussion threads via text mining. *Computers & Education*, *52*(2), 481–495.

Liu, H., Gegov, A. & Cocea, M. (2017). Rule based networks: an efficient and interpretable representation of computational models. *Journal of Artificial Intelligence and Soft Computing Research*, *7*(2), 111–123.

Liu, H. & Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective* (Vol. 453). Springer Science & Business Media.

Maghrebi, M., Sammut, C. & Waller, S. T. (2015). Feasibility study of automatically performing the concrete delivery dispatching through machine learning techniques. *Engineering, Construction and Architectural Management*, *22*(5), 573–590.

Maghrebi, M., Sammut, C. & Waller, T. (2013). Reconstruction of an expert's decision making expertise in concrete dispatching by machine learning. *Journal of Civil Engineering and Architecture*, *7*(12), 1540.

Majzoubi, F., Bai, L. & Heragu, S. S. (2012). An optimization approach for dispatching and relocating ems vehicles. *IIE Transactions on Healthcare Systems Engineering*, *2*(3), 211–223.

Menze, B. H., Kelm, B. M., Weber, M.-A., Bachert, P. & Hamprecht, F. A. (2008). Mimicking the human expert: pattern recognition for an automated assessment of data quality in mr spectroscopic images. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, *59*(6), 1457–1466.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, *42*(11).

Nasrollahzadeh, A. A., Khademi, A. & Mayorga, M. E. (2018). Real-time ambulance dispatching and relocation. *Manufacturing & Service Operations Management*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

RIVM. (2010, 9). Meetplannen subset logistiek (22nd ed.) [Computer software manual]. (Received by Geert Jan Kommer)

Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European journal of operational research*, *219*(3), 611–621.

Shaw, M. J. & Gentry, J. A. (1988). Using an expert system with inductive learning to evaluate business loans. *Financial Management*, 45–56.

Sudtachat, K., Mayorga, M. E. & Mclay, L. A. (2016). A nested-compliance table policy for emergency medical service systems under relocation. *Omega*, *58*, 154–168.

Theeuwes, N. B. J. M. (2018). *A review of literature on operational ambulance management models with a focus on applicability in practice.* (Unpublished)

Van Barneveld, T. (2016). The minimum expected penalty relocation problem for the computation of compliance tables for ambulance vehicles. *INFORMS Journal on Computing*, *28*(2), 370–384.

Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

Yue, Y., Marla, L. & Krishnan, R. (2012). An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *Twenty-sixth aaai conference on artificial intelligence.*

Appendices

A Simulation input

This Appendix outlines how the input for the simulation was obtained from available data. Unless otherwise indicated, data from the period between January 1 2017 and December 31 2018 was used.

A.1 Input samples

Input samples of ambulance requests were obtained from dispatch center data, by applying filters consistent with the instructions of the RIVM. This results in excluding requests outside the BZO region and those that are not indicated to be of 'SEH', 'poliklinisch', 'opname' or 'EHGV' nature.

The remaining number of requests in the year 2017 were checked with published numbers. Of those requests remaining, some postal codes of the request location were missing. However, the coordinates of the request location were always given. For those requests missing a postal code, the coordinates were mapped onto the closest postal code instead. Subsequently, the arrival time of each request was converted into the corresponding quarter of a week (0 - 671). Outliers, or logging errors, in both the treatment time and the transfer time (if applicable) were identified by computing the lower and upper outer bounds as follows:

$$Lowerouterbound = Q_1 - 3 * (Q_3 - Q_1) \tag{1}$$

$$Upperouterbound = Q_3 + 3 * (Q_3 - Q_1) \tag{2}$$

Those values outside of those outer bounds were replaced by its median value.

A.2 Answering times

The main performance measure in ambulance management revolves around the fraction of requests with a response time less than a threshold, which depends on the request's urgency. The response time lasts from the moment the call for an ambulance is answered in the dispatch center to the moment the (first) ambulance arrives at the request's location. Therefore, not only the ambulance's driving time is relevant, but also the time it took until the ambulance was dispatched. The *answering time*, i.e. the time it takes to answer a call for an ambulance, perform triage and dispatch the ambulance, is obtained from the data. Any unlogged answering times and errors (exceeding ten minutes) were excluded. The answering time depends on a request's urgency as follows:

- Answering time A1: 1:35 minutes
- Answering time A2: 2:21 minutes
- Answering time B: not applicable because requested in advance

A.3 Chute times

Between the moment an ambulance is dispatched and it starts driving, the ambulance crew needs to get into the ambulance (if it is not already driving), read the request details, load the driving route etc. This time is called the *chute time*, and is part of the delay in the simulation. Similarly to the answering time, its expected duration is extracted from the data, after excluding unlogged chute times and errors (exceeding ten minutes), and depends on a request's urgency as follows:

- Chute time A1: 0:46 minutes
- Chute time A2: 0.47 minutes
- Chute time B1/B2: 1:19 minutes

A.4 Shift roster

Table A.1: Shift roster (weekly) of ALS ambulances used as input to simulation (June - December 2018)

Station	Time (h)	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
Helmond	7:00	2	2	2	2	2	2	2
	8:00	1	1	1	1	1		
	10:00	1	1	1	1	1	1	1
	15:00	3	3	3	3	3	3	3
	23:00	2	2	2	2	2	2	2
Eindhoven N.	7:00	2	2	2	2	2	1	1
	8:00				1	1	1	1
	9:00	1	1	1	1	1		
	14:00	1	1	1	1	1	1	1
	15:00	3	3	3	2	2	2	2
	23:00	2	1	1	1	2	3	3
Eindhoven C.	7:00	1	1	1	1	1	1	1
	8:00	1	1	1				
	9:00	1	1	1	1	1	1	1
	14:00	1	1	1	1	1		
	15:00	1	1	1	2	2	1	1
	23:00	1	1	1	1	1	1	1
Eersel	7:00	1	1	1	1	1	1	1
	8:00	1	1	1	1	1		
	9:00						1	1
	15:00	2	2	2	2	2	1	1
	23:00	1	1	1	1	1	1	1
Valkenswaard	7:00	2	2	2	2	2	2	2
	10:00	1	1	1	1	1		
	15:00	2	2	2	2	2	2	2
	23:00	1	2	2	2	1	1	1
Total		35	35	35	35	35	29	29

Each week ambulance shifts are scheduled according to a base shift roster. This base shift roster dictates the number of shifts to start at each moment in the week, including the ambulance station where the shift should start (and hence end). Due to illness or vacations, however, the base shift roster is rarely adhered to. Therefore, a representative shift roster was deducted from realized shifts to be used as input to the simulation. Since the ambulance station in Eindhoven Noord was only taken into service in June 2018, after the closure of the ambulance station in Best, only data of realized shifts from the period June 2018 to December 2018 was used to deduct the shift roster.

B Confidence intervals of simulation results

Table A.2: Resulting performance for potential dispatch enhancements, with confidence intervals

Redispatch	Reevaluation	Coverage red.	Postpone A2	A1 requests		A2 requests		Nr. redisp./yr	Nr. reeval. P1/yr	Nr. reeval. P2/yr	Nr. reeval. P3/yr	Nr. postponed/yr
				On-time (%)	Mean RT (min:sec)	On-time (%)	Mean RT (min:sec)					
	Base			[93.58;93.69]	[9.02;9.03]	[97.01;97.11]	[13.61;13.65]	[1411;1438]				
x				[94.01;94.11]	[8.91;8.93]	[96.1;96.2]	[14.05;14.09]	[3272;3314]				
	x			[94.01;94.08]	[8.93;8.94]	[97.46;97.55]	[13.54;13.58]	[1398;1429]	[109;116]	[73;79]	[629;642]	
		x		[93.73;93.83]	[9;9.02]	[97.03;97.11]	[14.58;14.62]	[1396;1418]				
			x	[93.63;93.72]	[9.01;9.02]	[96.33;96.45]	[14.45;14.51]	[1542;1578]				[1322;1355]
x	x			[94.35;94.45]	[8.82;8.84]	[96.67;96.78]	[13.94;13.99]	[3243;3295]	[96;102]	[92;98]	[571;584]	
x		x		[94.12;94.22]	[8.89;8.91]	[96.02;96.1]	[15.04;15.08]	[3338;3383]				
x			x	[94.04;94.12]	[8.9;8.91]	[95.55;95.65]	[14.85;14.9]	[3403;3448]				[1273;1299]
	x	x		[94.07;94.16]	[8.91;8.93]	[97.45;97.53]	[14.51;14.55]	[1376;1404]	[106;111]	[71;76]	[614;628]	
	x		x	[94.00;94.09]	[8.92;8.93]	[96.77;96.88]	[14.38;14.45]	[1534;1575]	[106;111]	[77;82]	[621;633]	[1336;1369]
		x	x	[93.73;93.83]	[9;9.01]	[96.32;96.45]	[15.34;15.4]	[1522;1559]				[1290;1321]
x	x	x		[94.47;94.57]	[8.81;8.82]	[96.48;96.56]	[14.96;15.01]	[3324;3368]	[93;99]	[87;93]	[569;585]	
x	x		x	[94.37;94.46]	[8.82;8.83]	[96.07;96.16]	[14.78;14.83]	[3389;3422]	[93;99]	[91;97]	[565;579]	[1294;1321]
x		x	x	[94.15;94.25]	[8.88;8.9]	[95.46;95.59]	[15.73;15.79]	[3435;3484]				[1239;1269]
	x	x	x	[94.07;94.18]	[8.91;8.93]	[96.76;96.86]	[15.27;15.33]	[1515;1546]	[102;108]	[76;81]	[613;627]	[1298;1337]
x	x	x	x	[94.47;94.58]	[8.81;8.83]	[95.89;96.02]	[15.67;15.73]	[3418;3468]	[92;97]	[88;94]	[562;580]	[1243;1272]

*P1, P2, and P3 refer to priority one, two, and three of reevaluation, i.e. A1 request from late to on-time, A2 request from late to one time, and RT improvement of A1 request respectively.

C Coverage by ambulance stations

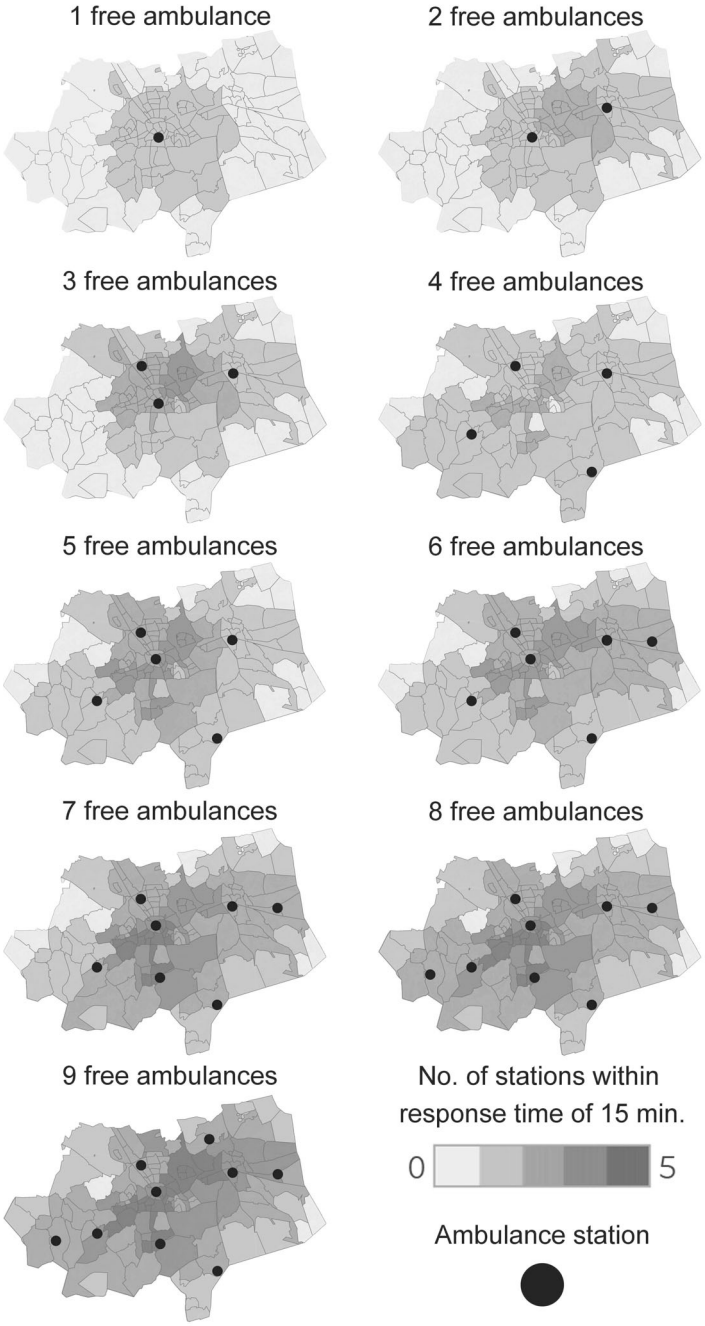

Figure A.1: Overview of postal codes that can be reached within a response time of fifteen minutes from the stations that should be occupied given the number of available ambulances