EFFICIENCY AND FAIRNESS IN AMBULANCE PLANNING

Caroline Jagtenberg

This work was supported by the Dutch Technology Foundation STW, which is part of the Netherland Organisation for Scientific Research (NWO), and partly funded by the Ministry of Economic Affairs (project number 11986).

Printed by: Dekkers van Gerwen *sustainable printing*
Cover design: Remco Wetzels, www.remcowetzels.nl

An electronic version of this dissertation is available at
http://research.vu.nl

VRIJE UNIVERSITEIT

EFFICIENCY AND FAIRNESS IN AMBULANCE PLANNING

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. V. Subramaniam,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op dinsdag 28 februari 2017 om 13.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Caroline Jagtenberg

geboren te Grubbenvorst

Promotoren:
prof. dr. R.D. van der Mei
prof. dr. S. Bhulai

Forethought we may have, but not foresight.

- Napoleon Bonaparte

Acknowledgments

In de eerste plaats wil ik graag mijn promotor en copromotor bedanken. Rob, je enthousiasme en can-do attitude hebben ervoor gezorgd dat ik mijn PhD behoorlijk optimistisch en zorgeloos doorlopen heb. Bedankt voor het bieden van alle mogelijkheden, voor de vele buitenlandbezoeken, en bedankt dat je me vrij liet om mijn eigen onderzoek richting te geven. Vier jaar lang was jouw standaardantwoord op al mijn verzoeken: 'Alles best, als er in 2016 maar een proefschrift ligt'. Ik ken geen enkele andere promotor met zo'n relaxte houding, dankjewel! Sandjai, ik weet niet wat ik meer op prijs stelde: je inhoudelijke hulp of jouw kleurrijke aanwezigheid. Beide waren een aanwinst. Je observaties waren scherp en je interesse oprecht. Dank je voor de samenwerking.

I would like to thank my committee members for their time invested in this dissertation. Armann Ingolfsson, Sally Brailsford, Erwin Hans, Frank Phillipson, Rommert Dekker and Ger Koole, I am grateful for your valuable comments, your approval and your willingness to travel to Amsterdam for the defense.

Dit proefschrift is tot stand gekomen tijdens het Repro project. Ik wil alle leden van deze groep bedanken voor de samenwerking en de interessante discussies. In het bijzonder bedank ik de andere PhD studenten: Thije, naast een kamer deelden we kilo's paaseitjes, liters groene thee en malle yogalessen waarin je een kwartier mocht slapen onder werktijd. Met jouw relaxte kijk op de afronding van je PhD was je een voorbeeld voor mij, waarschijnlijk zonder het zelf te beseffen. Pieter, ik vond het altijd mooi hoe jij in discussies telkens weer de spijker op z'n kop wist te slaan. Talloze keren hoefde ik mijn mening niet eens meer te verwoorden omdat jij dat spontaan al gedaan had. Maar mijn leukste herinneringen zijn de conferentiereisjes waarop we de raarste dingen meemaakten. Abseilen met gekke obers of bergen beklimmen met een loze ballon waren eerder regel dan uitzondering - en hoe ik na drie uur off-road rijden in de woestijn een McDonald's binnenstapte en jou daar aantrof, is me nog steeds een raadsel. Martin, het is leuk om te zien hoe enthousiast jij wordt van je promotieonderwerp. Daarnaast was je inzet rondom de pilot in Flevoland ongekend. Als er iemand in de ambulancewereld moet blijven werken dan ben jij het wel. Alle drie wil ik jullie bedanken voor jullie co-auteurschap bij verschillende papers die al dan niet hoofdstukken in dit proefschrift zijn geworden.

Gedurende de afgelopen vier jaar heb ik de mogelijkheid gehad om mijn kennis in de praktijk te brengen bij verschillende ambulancediensten. In het bijzonder wil ik graag de RAVU en de GGD Flevoland bedanken voor de prettige samenwerking. De 'kijkjes in de keuken' waren zeer nuttig, zowel in de ambulance als op de meldkamer. Daarnaast wil ik jullie bedanken voor het in mij gestelde vertrouwen met betrekking tot de uitkomsten van mijn algoritmes en analyses. Verder wil ik Geert Jan Kommer en zijn collega's bij het RIVM bedanken voor de input die zij aan dit proefschrift leverden. Van rijtijdenmodellen tot kaartmateriaal, jullie waren altijd bereid alles te delen. Ook vond ik het leuk om plaats te nemen in jullie 'klankbordgroep' voor het referentiekader spreiding en beschikbaarheid ambulancezorg 2016. De discussies over de manier waarop DAM en B-vervoer in het capaciteitsmodel moeten worden meegenomen waren erg interessant. En misschien nog interessanter was de dynamiek die ik daarbij ervaren heb tussen zorgverzekeraars, ambulancediensten en onderzoekers: dit was zeer prettig en inspirerend. Bedankt dat ik daar deel van uit mocht maken.

Graag wil ik op deze plek ook Marc Uetz bedanken. Marc, mijn afstuderen bij jou zorgde ervoor dat ik al snel een heel rooskleurig beeld kreeg van promoveren. Onze samenwerking was ook zo leuk dat ik Enschede zelfs bijna als woon- en werkplaats heb overwogen. Het haalde het alleen nét niet bij Amsterdam, sorry... Ons werk heeft uiteindelijk een plekje gekregen in Hoofdstuk 7.

Tijdens het grootste deel van mijn PhD werkte ik twee dagen per week bij ORTEC. Alexander Poot, in eerste instantie stond je daar niet om te springen, en door de jaren heen liet je geen kans onbenut om te vragen wanneer ik nou eindelijk weer eens fulltime terug kwam. Maar toch heb je er voor gekozen om me parttime in dienst te houden. Ik weet dat dat een grote uitzondering is, en ik wil je er bij dezen voor bedanken. De afwisseling in werkgevers zorgde ervoor dat ik van beide kanten de voor- en nadelen zag, dat ik deze makkelijker kon relativeren, en dat ik me natuurlijk nooit hoefde te vervelen. Gedurende die jaren heb ik een aantal managers versleten. Jeroen Leeflang, Chantal van der Meer en Gerhard Post, hartelijk bedankt voor jullie flexibiliteit en begrip met betrekking tot mijn CWI werkzaamheden. Ook jullie tactische aanpak ('Ik zal Alexander wel vast voorbereiden') als ik weer eens met een verzoek kwam, heb ik zeer gewaardeerd.

I thank the board members of the LNMB, and especially Johann Hurink, for the interesting conversations that we had at board meetings, conferences and other events. You may not realize it, but you have at times changed the way I reflect on my own work, and on life in academia in general. I have really enjoyed my time as a PhD representative and it will forever be beyond me that I had no competition for that position.

During the last year of my PhD I visited Andrew Mason at the University of Auckland. Andrew, thanks for our fruitful collaboration, which resulted in Chapters 5 and 6. I was - and still am - very happy that you suggested that we work on fairness. Also, you were a wonderful host: your Herne Bay dinners and yacht races made me feel more than welcome. I greatly appreciated this Kiwi hospitality. Additionally, I would like to thank the PhD students at Engineering Science for making me feel at home in New Zealand. In particular: B-Meister, thanks for the work hard play hard attitude; Frodo, thanks for your wise words; McDane, tak for being the best office mate. Osc, thanks for being the Number 8 wire for Chapter 5.

Verder wil ik graag mijn familie bedanken. Jullie waren altijd geïnteresseerd in hoe mijn PhD er voor stond. Links en rechts hebben jullie zowaar zelfs de moeite genomen om papers van me te lezen. Dat heeft me eigenlijk wel verbaasd, want het is immers maar werk. Bedankt voor deze toewijding, maar eigenlijk wil ik jullie op deze plek vooral bedanken voor de niet-werkgerelateerde steun. Jullie wijze woorden en adviezen zijn heel waardevol voor me geweest, en daardoor hield ik genoeg ruimte in mijn hoofd over om dit proefschrift te kunnen volbrengen.

Ik bedank ook al mijn vrienden die de hoogte- en dieptepunten van de afgelopen vier jaar van dichtbij hebben meegemaakt. In het bijzonder: Arkmm en Awesommie, bedankt voor jullie al meer dan tien jaar durende vriendschap. Jullie zijn heel verschillend, en toch geven jullie me dezelfde redenen om jullie als paranimfen te kiezen: jullie luisteren naar me als ik wil praten, dansen met me als ik wil feesten, en drinken met me als ik wil vergeten. Al moet ik bekennen dat dat laatste wel pas met de jaren kwam, voor mij althans! Ik ben trots om jullie tijdens de verdediging aan mijn zijde te mogen hebben.

Last but not least: Scar. Dit PhD traject staat vrijwel gelijk aan de periode waarin wij onze levens deelden. Er is niemand die me meer heeft aangemoedigd dan jij, niemand die trotser was wanneer ik iets publiceerde en niemand die meer vertrouwen had in een goede afloop. Verder was jouw point of view - dat ik er misschien gelukkiger van werd als ik me op het proces zou focussen in plaats van op het resultaat - een openbaring voor me waar ik jaren profijt van heb gehad. Bedankt voor je eindeloze interesse in mijn onderzoek, maar meer nog bedankt voor de onuitputbare bron van afleiding die jij bent geweest.

Caroline Jagtenberg
Amsterdam, oktober 2016

Contents

1 Introduction 1

1.1 Background and motivation 1
1.2 Literature review 3
1.3 Ambulance care in the Netherlands 10
1.4 Thesis outline 12

2 Dynamic ambulance dispatching: is the closest-idle policy always optimal? 15

2.1 Introduction 15
2.2 Problem formulation 17
2.3 MDP-based solution 18
2.4 Heuristic solution 23
2.5 Simulation model 25
2.6 A motivating example 26
2.7 Computational results 29
2.8 Discussion 38

3 Benchmarking online dispatch algorithms 41

3.1 Introduction 41
3.2 Problem formulation 44
3.3 Solution methods 46
3.4 Worst case analysis 51
3.5 Computational results 52
3.6 Discussion 56

4 An efficient heuristic for real-time ambulance redeployment 63

4.1 Introduction 63
4.2 Problem formulation 64
4.3 Algorithm 66
4.4 Computational results 70
4.5 Discussion 78
4.6 Implementation in practice 80

5 Fairness in the ambulance location problem 83

5.1 Introduction 83
5.2 Problem formulation 86
5.3 Utilities 88

5.4 A small example 92
5.5 Methods 93
5.6 Computational results 97
5.7 Discussion 105

6 Improving fairness by time-sharing ambulances 107
6.1 Introduction 107
6.2 Preliminaries and related Work 108
6.3 Motivating examples 110
6.4 Optimization model 112
6.5 Computational results 114
6.6 Multi-objective optimization model 117
6.7 Discussion 120

7 Analysis of Smith's rule in stochastic machine scheduling 125
7.1 Introduction 125
7.2 Recap of the Kawaguchi & Kyan instance 127
7.3 The stochastic Kawaguchi & Kyan instance 128
7.4 Discussion 136

Bibliography 139

Summary 147

Samenvatting 150

About the author 154

List of publications 156

1
Introduction

In 1792, Napoleon Bonaparte decided to have his injured soldiers dragged off the battle field by horse-drawn carriages. At the time, these so-called *flying ambulances* were a novel idea, and they proved a complete success: not only did they increase the chances of survival of the wounded soldiers, they also lifted the morale and the confidence of the French troops [105]. Nowadays, ambulances are commonplace: in the Netherlands alone there are more than one million ambulance trips per year [89]. Emergency medical services (EMS) have evolved into a complex system of interacting ambulances, dispatch centers and hospitals, providing us with a challenge to model and optimize their dynamics. A timely response can literally be a matter of life or death, so naturally research is focused on reducing response times. One solution to improve response times is simply to drive faster. While Top Gear has looked into this option [10], this dissertation takes a different approach. We introduce mathematical models for various planning stages in the EMS process, aiming to reduce response times by a more efficient use of resources.

We continue this chapter by describing the events and processes that occur in an EMS system, and the typical planning questions an ambulance provider might face. Additionally, we give an overview of the literature in this field. Since most of the case studies in this thesis involve the Dutch EMS system, we also include a brief description of ambulance care in the Netherlands. We finish with an outline of the remaining chapters of this thesis.

1.1 Background and motivation

Emergency medical services deal with urgent requests for medical care and/or patient transport. A typical response process is as follows. The EMS provider learns about a requests when a call arrives at the dispatch centre. The call is answered by a dispatcher, who starts the *triage*: a process to determine the location of the patient and the urgency of the request. If an ambulance is needed, the dispatcher decides which vehicle to send to the scene of the incident. Almost always, this will be the closest idle ambulance - except if a special vehicle is needed due to the specifics of the incident. The ambulance drives to the scene of the incident, where the paramedics spend a certain amount of time with the

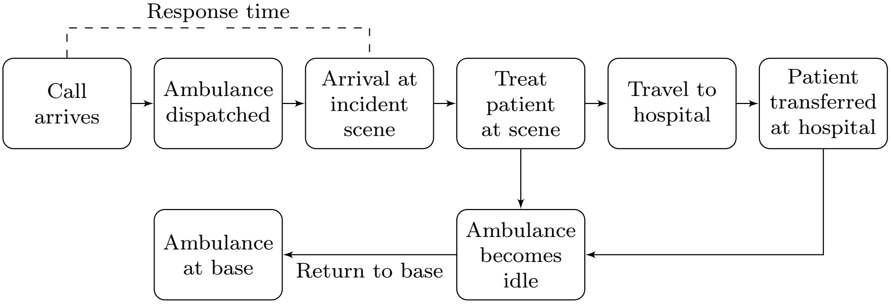


Figure 1.1 A typical response process for an emergency call.

patient. Then, it is decided whether or not the patient needs to be transported to a hospital. If not, the ambulance becomes idle at the scene of the incident. Otherwise, there is a travel time to the hospital, followed by a drop-off time during which the crew transfers the patient to the emergency department. If an ambulance becomes idle, it returns to one of the predefined waiting sites or *bases*. This response process is depicted in Figure 1.1.

In case a call arrives while all ambulances are busy, the dispatcher places the request in a queue, in order for it to be served as soon as a vehicle becomes idle. This situation, sometimes called *code red*, is quite rare in the Netherlands, but it appears to be more common in other places (e.g., Edmonton, Canada [58]).

Although there can be some differences between countries, the main goal of ambulance providers world wide is the same: provide good health care at a reasonable cost. Naturally, a tradeoff arises, and this warrants research for efficient operations. Medical decisions aside, in order to obtain efficiency there are many logistic aspects worth considering.

When ambulance providers face questions regarding their planning, geographical aspects and service level agreements are often involved. Typical questions are, for example: 'Can we improve performance by placing bases in different locations?' or 'If we were to purchase one extra vehicle, would we be able to serve 95% of all calls on time?' Other questions might be staff-related, such as whether to hire more paramedics, or reconsidering the roster. Perhaps an EMS provider is thinking about merging with a neighbouring provider, and wants to know how this will affect response times. Furthermore, EMS managers may anticipate to new scenarios, due to changing circumstances that they want to evaluate in theory before it occurs in practice. These *what if* scenarios could for example be: 'What if this hospital closes their emergency department?' or 'What if the demand for ambulances increases by 5%?'

The effects of EMS related changes or decisions can be difficult to oversee due to the stochastic nature of incidents. Furthermore, the decisions involved are often interrelated. This creates challenging mathematical problems, which - combined with the importance of high-quality EMS operations to society - have led Operations Research practitioners to pay much attention to EMS systems.

Over the years, numerous mathematical models have been developed that deal with planning and efficiency questions.

The decisions involved in planning EMS operations can be divided in three different planning stages: (1) at the *strategic* level, long-term decisions are made such as the opening and closing of base locations, the purchase of vehicles and the hiring and firing of staff; (2) the *tactical* level deals with the medium-long term, which may include decisions like how many vehicles to position at each base and how to design a staff roster, and (3) *operational* planning involves day to day or even real-time decisions. The latter includes decisions regarding the dispatch policy, which hospital to choose and where to send idle vehicles.

In practice, the same ambulance providers that serve emergency requests also handle non-urgent patient transport. These transports are often ordered and scheduled in advance, which make them intrinsically different from the urgent requests. The transports can be planned, and consequently have led to a separate set of models in literature. While some decisions may involve both the urgent *and* non-urgent ambulance operations - for example, when they are executed by the same fleet - this thesis focuses on the urgent requests only.

1.2 Literature review

This section discusses the literature on ambulance planning and gives a short overview of the various techniques used. We focus on models that can be solved analytically, which we divide in two types: (1) *static* models, which deal with problems at the strategic and tactical level, and (2) *dynamic* models, which are concerned with daily or even real-time planning.

1.2.1 Static planning

When it comes to ambulance planning, strategic and tactical problems are often solved simultaneously. At this planning stage the problems are often *emergency facility location problems.* They deal with two types of decisions: ‘Which bases should be opened?’ and ‘How many vehicles should be placed at each base?’

At this point, static models are often used to describe the problem. Here, ‘static’ means that each ambulance is assigned to a base location, and after serving an incident the ambulance is assumed to return its own home base. Typically there is a limited number of vehicles that need to be distributed over a set of possible base locations. These static models often use integer linear programming (ILP) to solve the problem.

Numerous objectives exist in literature, inspired by either the local EMS rules or a researcher’s belief of what is a relevant measure. This section gives a brief overview of the literature; for a more elaborate discussion, see [71] and [22].

A common way to measure the performance of an EMS provider is in terms of the *fraction of late arrivals*, i.e., the fraction of all calls for which the response time is larger than a certain response time threshold (RTT). This is probably the most widely used objective - and certainly the most relevant for the work in this thesis.

Early research in ambulance planning focused on deterministic location problems. These formulations ignore the stochastic aspects of an EMS system, for example by assuming that one vehicle is enough to cover a demand point. This is done, e.g., in the Location Set Covering Problem (LSCP) [110], which searches for the minimal number of bases to cover a region, and in the Maximal Covering Location Problem (MCLP) [34], which searches for the best possible locations for a given number of bases. Slightly more advanced models such as [51] recognize that one vehicle per base is most likely not enough to cover the demand; they include backup coverage by requiring a constant number of vehicles within reach of each demand point.

Later, research turned to probabilistic models: these explicitly model the probability that a vehicle is busy (due to serving other patients). A well-known example is the Maximum Expected Covering Location Problem formulation (MEXCLP) [36]. The MEXCLP model is particularly relevant for this thesis, as the underlying idea of MEXCLP is used throughout several chapters. Therefore, we next recap the full model as it was originally published.

THE MEXCLP MODEL. In this formulation there is a set of ambulances, denoted A, that needs to be distributed over a set of possible base locations W. Each ambulance is modeled to be unavailable with a pre-determined probability q, called the *busy fraction*. Consider a node $i \in V$ that is within range of k ambulances. The travel times τ_{ij}, $i, j \in V$ are assumed to be deterministic, which allow us to straightforwardly determine this number k. If we let d_i be the demand at node i, the expected covered demand of this vertex is

$$E_k = d_i(1 - q^k). \tag{1.1}$$

The marginal contribution of the kth ambulance to this expected value is $E_k - E_{k-1} = d_i(1-q)q^{k-1}$. We introduce a binary variable y_{ik} that is equal to 1 if and only if vertex $i \in V$ is within range of at least k ambulances. The variables x_j (for $j \in W$) represent the number of vehicles at each base. Let W_i denote the set of bases that are within range of demand point i, that is: $W_i = \{j \in W : \tau_{ij} \leq T\}$, then we can formulate the MEXCLP model as:

$$\text{Maximize} \sum_{i \in V} \sum_{k=1}^{p} d_i(1-q)q^{k-1} y_{ik}$$

subject to

$$\sum_{j \in W_i} x_j \geq \sum_{k=1}^{p} y_{ik}, \quad i \in V,$$

$$\sum_{j \in W} x_j \leq |A|,$$

$$x_j \in \mathbb{N}, \quad j \in W,$$

$$y_{ik} \in \{0, 1\}, \quad i \in V, k = 1, \ldots, p.$$

Note that there is no need to add the constraint $y_{ih} \leq y_{ik}$ for $h \leq k$. This will always hold for an optimal solution, since $E_k - E_{k-1}$ is decreasing in k.

Using a busy fraction makes the MEXCLP model elegant in its simplicity, but the underlying assumptions are quite strong. For example, in the definition of E_k in Equation (1.1) the underlying assumption is that vehicles operate independently. Furthermore, the busy fraction is the same for all vehicles, regardless of their position with respect to the demand and the other vehicles.

Despite these assumptions, the MEXCLP model has several upsides. First of all, the simplicity of the model ensures it is scalable. Second, it is a suitable base for many extensions. For example, there are extensions with stochastic travel times [17, 54], and a time-dependent version that divides the time horizon in a set of time periods [15].

A slightly different approach is taken in the Maximum Available Location Problem (MALP). MALP also uses a busy fraction q, but maximizes the population that will find a vehicle available within a time standard with a certain (fixed) reliability [97].

Some of the strong assumptions in MEXCLP - independent vehicles all having and the same busy probability - are relaxed in the Hypercube Queuing Models (HQM) [69], providing a more accurate representation of real systems. However, it should be noted that while restrictive assumptions limit a model's applicability, improving the modelling of the system performance makes the problem increasingly complicated and correspondingly more difficult to optimize.

There are other models that consider more than just response time thresholds. We leave the definition of other performance indicators open, but examples include an average response time, or a probability of survival. To compute such performance measures, it is useful to condition on the incident location and the base location of the responding ambulance. These models implicitly or explicitly assume that the closest idle vehicle is sent to an incident.

1.2.2 Dynamic planning

Dynamic models are used in the *operational* planning. They concern on-the-fly decisions, based on real-time information such as the current position and status of vehicles. Note that this stands in contrast to the static models described earlier. Dynamic solutions often outperform static solutions; however, optimality can usually not be guaranteed. This section briefly summarizes the literature on dynamic models; a more elaborate overview can be found in [11].

Most dynamic models concern *redeployment*: they aim to find good (re)distributions of vehicles when a number of ambulances is busy responding to incidents. This is sometimes referred to as *repositioning*, *dynamic ambulance management* or *move-up*. Over the last few years, redeployment has become increasingly popular in practice. Surveys of North American EMS operators showed that the percentage of operators who used a dynamic strategy increased from 23% in 2001 [28] to 37% in 2009 [117] (see also [2]). This indicates that the EMS community is becoming more aware that a dynamic policy can help to perform better without increasing capacity.

Dynamic models usually do not search for good base locations, but instead consider the bases as a given, fixed set. The point of issue is to make decisions based on real-time information on the state of all vehicles and incidents. This makes for a complex problem, and systems quickly become intractable when the number of vehicles grows.

Perhaps it is due to the difficulty of the problem, that dynamic models attract a wide range of solution methods. For example, there are approaches using dynamic programming [18], Integer Linear Programming (ILP) [46], stochastic programming [87], simulation-based optimization [20] and approximate dynamic programming [77]. The redeployment policies that have been published so far are roughly dividable in two subclasses, which we will refer to as *compliance tables* and *real-time optimization.*

Compliance tables are essentially lookup tables describing the desired configuration for each number of available ambulances. In order to obtain such a table, the system's state is defined as the *number of available ambulances*, and a model is formulated to find an optimal configuration for each state. Typically, such a model is an ILP that maximizes some objective for all possible system states. Constraints may be added to control the number of vehicles relocated between states. The model is solved once (a priori), and the result is stored in a compliance table, to be looked up and applied when needed. Examples of such models are [46, 86].

In general, it is hard to give a reasonable estimate for the performance of a lookup table policy without simulating the system. However, [2] introduces a Markov chain model that provides a good approximation to several performance measures. This model can thus be used to identify near-optimal lookup tables.

A redeployment policy in the form of a lookup table has advantages and disadvantages. On the upside, a lookup table is easy to explain, and many EMS providers are familiar with this type of policy. Note that the job of steering the set of available vehicles towards the prescribed configuration is usually left to the dispatchers. This brings us to a downside: a poorly executed redeployment can devaluate even the most crafty lookup table. Furthermore, a lookup table is in general not able to suggest the most effective move-ups, because the amount of information used is limited. Another type of policy - that also uses the current locations of vehicles - may therefore perform better or require fewer move-ups, or both. Moreover, note that in busy regions, where the number of idle ambulances changes rapidly, the system will not be in compliance with the lookup table for most of the time.

Real-time optimization models calculate the 'best' ambulance movement in real time. The first of such models is known as the Dynamic Double Standard

Model (DDSM), published in 2001 [45]. This is an extension of the static Double Standard Model [44]. The model is an integer program (IP) that maximizes the demand covered twice, subject to two coverage constraints with different thresholds. The suggested moves are balanced with a certain cost that takes into account ambulance activity history: this reduces the number of moves that are undesirable from the crew's perspective (such as round trips).

Various papers model the randomness in the system explicitly, for example, by formulating the problem as a Markov decision process. When the model has only a few ambulances, one can solve it using exact dynamic programming (e.g., [121]). However, when the state space grows - for example due to the number of vehicles considered - the problem quickly becomes intractable. This is known as the *curse of dimensionality* [93].[1] Hence, in order to compute results for realistically-sized EMS regions, one needs to turn to alternative solution methods.

One way to deal with this curse of dimensionality is by looking only one time-step ahead. This is done in [6], which classifies possible redeployment actions by constructing several scenarios that may occur one time-step later and evaluates each feasible action under these scenarios. Another example of a myopic approach is [118], which determines redeployments of idle ambulances from a greedy algorithm that attempts to minimize a weighted sum of expected late and lost calls, as evaluated through simulations.

Other authors overcome the curse of dimensionality by applying Approximate Dynamic Programming (ADP). ADP is a powerful tool for solving stochastic and dynamic problems, and scales well to high-dimensional applications. There are multiple ways to apply ADP. In [102] the authors use a combination of *aggregation* and the *post-decision state.* The original problem is aggregated by placing a spatial grid over the geographic area, and dividing the time horizon in sub-intervals. The value function is then approximated by computing estimates for the aggregated states. The post-decision state describes the state of the system immediately after making the decision but before any new information arrives. Approximating the value function around the post-decision state removes the stochasticity at this point. For an elaborate discussion of the post-decision state, see [93]. In [77] ADP is applied in a different way. The value function is approximated by a linear combination of so-called *base functions*: well-chosen functions that each use limited state information, and are considered to hold explanatory power over the value of a state. The parameters that define the importance of each base function are tuned using simulated cost trajectories of the system. The mechanism to tune parameters to the use case is described in more detail in subsequent work [78]. This approach is a novel one, but it is time consuming to both implement and execute: for a large city the tuning process can take a year, which is reduced to twelve hours by using the post-decision state. It remains possible to calculate the repositioning decision in real time, because these heavy computations are done in a preparatory phase. The performance of the method

[1] In fact, [93] mentions three curses of dimensionality: the state space, the outcome space and the action space.

is highly dependent on the choice of base functions. In [78], base functions are essentially Erlang loss functions: the city is decoupled into smaller, independent regions each containing only a single ambulance base, and each region is modeled as an Erlang loss system. Note that this includes the implicit assumption that an incident is likely to be served late if there are no idle vehicles present at the nearest base.

In [120, Chapter 8] the authors introduce an IP model that they claim can be viewed as an extension of the ADP model in [77]. The model is extended in multiple ways, including the use of at-hospital ambulances and adding a cost for moving an ambulance to a base. Furthermore, the tuning process is updated, although the general idea that simulations are employed for function evaluations remains the same. The article mentions that future research may be directed towards rewards collected on the road during ambulance moves, which is relatively unexplored in current literature.

Although it appears that the majority of the dynamic models has not been implemented in practice, [75] is an exception. It describes an IP model that has been implemented in a commercial software package called *Optima Live* [75]. The method is a real-time optimization system that maximizes the total value from user-defined coverage reward functions minus redeployment costs.

In order to evaluate and validate move-up models, researchers typically use simulation. This makes it possible to get realistic estimates of the performance of an EMS system. Simulation is also useful stand-alone, to evaluate and compare scenarios. This is done for example in [53], which estimates the impact if all ambulances in Edmonton were to begin and end their shifts at the same location. Finally, simulation is used in so-called simulation optimization approaches (e.g., [122]). An overview of computer simulation models used for the analysis and improvement of EMS can be found in [1].

1.2.3 Model features

Several choices can be made regarding the modelling of EMS processes. In general, it is safe to say that George E. P. Box was right when he said: *“All models are wrong, but some are useful.”* [21, p.424]. This section discusses the most important model features, motivates the choices in this thesis and summarizes alternative approaches in literature.

Response times

A *response time* is defined as the time between the receipt of a call until an ambulance arrives on-scene (see also Figure 1.1). It consists of several components. First of all, the triage process takes place. Then, the dispatcher decides which ambulance to send. Subsequently, the crew makes their way to its ambulance. Together, these three events constitute the *pre-trip delay.* The total response time is given by the pre-trip delay plus the actual driving time.

In literature we see several ways to incorporate this pre-trip delay. It is not uncommon to simply add the average delay to the travel time. However, in [54] it is argued that the duration of the pre-trip delay is highly variable, and that hence such a deterministic approach is insufficient to accurately predict performance. This article includes a small case study showing that it may lead to either an under- or overestimation, but being only a couple of percent off, the magnitude of these mistakes seems small.

In this dissertation we take the same approach as Dutch ambulance providers: three minutes are 'reserved' for the pre-trip delay. This leaves at most twelve minutes of driving time in order to reach the incident within the prescribed fifteen minutes.

It is also debatable whether the *driving times* should be modeled as deterministic or stochastic. In literature many examples can be found taking either approach. Although the stochastic approach seems realistic, authors do not seem to agree on which distribution to use. For example, [54] suggests a lognormal distribution, whereas [17] proposes a normal distribution. Differences may depend on many things, including the country where the case study took place. Most chapters in this thesis assume driving times to be deterministic. This is perhaps the biggest simplification done in our models. Although they could be extended to stochastic driving times, this would make the notation more cumbersome - and solutions harder to compute.

Other literature includes approaches based on the distance between two points as the crow flies [3] and using Google Maps data [62] (perhaps multiplying the result with a factor to correct for the fact that EMS vehicles usually drive faster than regular cars).

Vehicle and patient types

Recall from Section 1.3 that an EMS provider may use several types of vehicles. Each vehicle type has its own characteristics, such as travel speed or the ability to reach a certain area. Vehicles and the corresponding crew may also differ in their capabilities regarding the handling of patients. For example, less equipped vehicles may not be able to help the most severely injured patients. Other vehicles may serve a patient at the incident scene, but do not provide patient transportation. Not only the vehicles, but also the patients may fall into different categories: the nature of the request may cause a need for a specifically equipped vehicle, for example with an incubator or a psychiatric nurse. Also, the patient's urgency may dictate the use of a different response time target.

The EMS system is rather complex, and to accurately capture it one also needs a complex model. Incorporating one or several of the features above would make the problem even less tractable. Furthermore, such a model would lead to a solution that is highly tailored to the specific situation. Instead, this dissertation focuses on a single type of patients, all equally urgent, and one type of vehicle which is capable of serving any patient.

Variations over time

Several aspects of the EMS process may vary throughout the day or week. Some models in literature anticipate time-dependent fluctuations, generally by considering the redeployment of ambulances to be pre-planned. These models are extensively surveyed in [71] and [22]. The rest of this section explores variations in demand and travel times in more detail.

It is commonly assumed that EMS call volumes follow a Poisson process. However, call volumes may vary by month, day of the week, and hour of the day (see for example [52]). If this is the case, the arrival process may be modelled as a time-varying Poisson process. To predict the arrival intensity in the future, there are several methods available. Successful approaches in an ambulance context include classical time series models [30] and Singular Spectrum Analysis (SSA) [114].

The travel times, and the corresponding coverage, may also vary over time. Some papers consider this explicitly (e.g., [103]). However, we point out that emergency services do not always experience the impact of the time of day on their response velocities. For example, empirical evidence shows only a minor impact for fire fighters in New York [64] and ambulances in Calgary [24]. Furthermore, even if one is certain that the time of day is relevant for the response velocities, the task remains to estimate the different velocities accurately. Care has to be taken of how to handle the data, for example, there is a risk of overfitting due to the data containing only a small number of trips from i to j in each time segment.

The methods and models introduced in this thesis assume fixed values for the demand and travel times. This assumption simplifies the notation and discussion, and allows the reader to focus on the core ideas of the proposed methods. When used in practice, the correct usage of models should be discussed with EMS managers. For example, one can simply use the peak demand (this is done, e.g., in [76]), or the week may be split up into different time blocks, using different parameters for each block.

1.3 Ambulance care in the Netherlands

The Netherlands is divided in 24 EMS regions, called Regionale Ambulancevoorzieningen (RAVs), depicted in Figure 1.2. The RAVs operate independently, although occasionally a neighboring RAV may be contacted for help.

In the Netherlands, ambulance providers distinguish four different call priorities. The most urgent calls are labeled A1, and require an ambulance to be at the scene within 15 minutes in 95% of the cases. A1 calls would include, for example, heart attacks or strokes. A response to A1 calls generally includes lights and sirens. Non-life-threatening yet urgent calls get an A2 label, which corresponds to a response time threshold of 30 minutes. Although lights and sirens are usually omitted for A2 calls, an ambulance is dispatched immediately. Both A1 and A2 calls have to be served by Advanced Life Support (ALS) ambulances. Additionally, there are B calls, which are non-urgent patient transports. These transports are often ordered in advance, and consequently there is no time

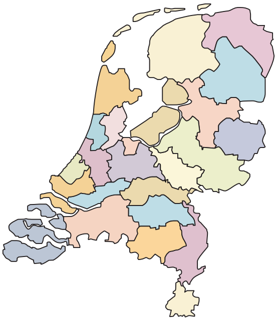


Figure 1.2 Currently, the Netherlands has 24 EMS regions.

standard for B calls. The B calls are further subdivided: B1 patients need to be transported in an ALS ambulance, while for B2 patients a cheaper Basic Life Support (BLS) ambulance would also be sufficient. All vehicles can transport at most one patient at a time. Unlike some other countries, the Netherlands uses the same standards for urban and rural areas[2].

The association *Ambulancezorg Nederland* (AZN) reports on the numbers of production and performance of RAVs on an annual basis. In 2014, 1,190,320 incidents were served in the Netherlands. Roughly 49% of these were A1 calls, 24% were A2 calls and 27% B calls. They were served using 231 bases and 755 vehicles nationwide. For 93.4% of the A1 requests an ambulance arrived within the prescribed fifteen minutes. The average A1 response time, however, was a lot smaller: 6 minutes and 41 seconds [89].

The Dutch National Institute for Public Health and the Environment (RIVM) [98] distributes the national budget for ambulance care among the different RAVs. This is done using several mathematical models, which are updated and published every few years (see, e.g., [65, 67]). The RAVs are free to spend their budget whichever way they choose, for example by investing in different vehicle types that they deem appropriate. Currently, there exist configurations ranging from a paramedic on a bike to a mobile intensive care unit (micu) - pretty much an operating room on wheels - with three staff members on board. Additionally, some RAVs use helicopters and boats to extend their service.

When it comes to modelling Dutch EMS systems, there are a few standard approaches. For example, demand is typically aggregated by using the first four digits of postal codes. This leads to regions of moderate sizes, with 40 to 456 demand points. The demand per point can then be estimated in a few different ways. One may use the observed demand in recent years, albeit at the risk of overfitting. Alternatively, one may assume that demand is roughly proportional to the number of inhabitants per demand point. This assumption may not be completely correct, but on the upside the number of inhabitants is known with

[2] although one might argue that the Netherlands does not have truly rural areas

great accuracy.

The RIVM estimated ambulance driving times between any two postal codes in the Netherlands [66, Chapter 3]. For this purpose, the RIVM used historical data from ambulances that drove from a base location to an incident. These measurements were further differentiated by time of day and type of region (urban or not). The travel times were then predicted by distinguishing twelve different road types, and estimating the travel speed at each road type.

1.4 Thesis outline

This thesis contains research that focuses both on theoretical results and practical applications. The content of this thesis is organized in six chapters.

Chapter 2 deals with the dispatch process. Most literature assumes that the closest idle ambulance is always sent, but this is not necessarily optimal. We provide two alternatives for the 'closest idle' method: one method is obtained by modelling the dispatch process as a Markov decision process, the other method is a heuristic. The *optimal* dispatch policy, however, remains unknown.

In Chapter 3 we bound the performance of an optimal dispatch policy. We do this by computing the optimum for the *offline* version of the dispatch problem. In the offline dispatch problem, the time and location of incidents are known in advance, which allows us to get better solutions than for the online problem. We analyze the problem from both a worst-case as well as the average-case point of view. By benchmarking the offline optimum against online policies, we give the first quantification of the 'performance gap' between online and offline dispatch policies.

Chapter 4 introduces an algorithm for proactive ambulance redeployment. Unlike many other redeployment algorithms in literature, our proposed solution is a polynomial-time heuristic that is easy to implement. We evaluate its performance in a simulation model of (EMS) operations and compare it to static solutions. The practical relevance of this chapter is demonstrated by the implementation of our heuristic in practice.

In Chapters 5 and 6 we focus on fairness in ambulance planning. Most models in ambulance planning maximize the number of people served, regardless of where they are living. This approach benefits people living in cities, at the expense of people living in remote areas. While most alternative models tend to aim for equity (providing the same service to people in every location), we seek for a compromise between these two options. This is done by viewing the ambulance location problem from a social welfare perspective: we show that maximizing the so-called *Bernoulli-Nash* social welfare results in a solution that we consider fair. Chapter 5 and 6 approach fairness in different ways: Chapter 5 introduces a facility location problem: we compute where to locate vehicles such that the Bernoulli-Nash social welfare is maximized. This requires the use of a non-linear model, which we approximate with piecewise linear functions and solve using a Mixed-Integer Linear Programming (MILP) solver. Chapter 6, on the other hand, proposes to improve fairness by time-sharing several static ambulance configurations. The individual configurations are evaluated by simulation, and the optimal mix between configurations is then computed using an Interior Point optimizer.

Finally, Chapter 7 deals with stochastic scheduling: the scheduling of jobs with a stochastic processing time, on parallel, identical machines. In particular, we focus on *Smith's rule* - scheduling jobs according to ratios weight over processing time - for minimizing the weighted sum of completion times. For jobs with deterministic processing times, Smith's rule is known to have a tight performance guarantee of $(1 + \sqrt{2})/2$. We recap the instance that proves this performance bound is tight, and analyze its stochastic counterpart with exponentially distributed processing times. Our analysis allows us to derive new qualitative insights, and sheds light on previously unknown phenomena in stochastic scheduling.

The work that resulted in this thesis was part of a larger project, called 'From REactive to PROactive planning of ambulance services', shortly REPRO. The REPRO project was focused on several areas, including (1) the development of relocation algorithms for dynamic ambulance management, using for example compliance tables [5] or taking into account different vehicle types [8], (2) the development of facility location models, incorporating aspects such as fractional coverage [17] and time dependency [15], and (3) the development of capacity models for EMS call centers [25]. A key aspect of REPRO is that the research not only led to a range of academic contributions, but that the tool implementations of the models were also successfully applied in real life, supporting the operational processes of several ambulance service providers in the Netherlands [27] and in Norway [100].

2
Dynamic ambulance dispatching: is the closest-idle policy always optimal?

This chapter addresses the problem of ambulance dispatching, in which one must decide which ambulance to send to an incident in real time. In practice, it is commonly believed that the ‘closest idle ambulance’ rule is near-optimal and it is used throughout most literature. In this paper, we present alternatives to the classical closest idle ambulance rule. The first alternative is based on a Markov decision problem (MDP) that remains computationally tractable for reasonably-sized ambulance fleets. The second alternative is a heuristic for ambulance dispatching that can handle regions with large numbers of ambulances. Our main focus is on minimizing the fraction of arrivals later than a certain threshold time, but we show that with a small adaptation our MDP can also be used to minimize the average response time. We evaluate our policies by simulating a large EMS region in the Netherlands. For this region, we show that our heuristic reduces the fraction of late arrivals by 18% compared to the ‘closest idle’ benchmark policy. A drawback is that this heuristic increases the average response time (for this problem instance with 37%). Therefore, we do not claim that our heuristic is practically preferable over the closest-idle method. However, our result sheds new light on the popular belief that the closest idle dispatch policy is near-optimal when minimizing the fraction of late arrivals.

This chapter is based on:
C.J. Jagtenberg, S. Bhulai and R.D. van der Mei. Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health Care Management Science*, To appear.

2.1 Introduction

The vast majority of the papers on dynamic ambulance management focus on how to redeploy idle vehicles (e.g., [2, 77, 118]). Perhaps in order not to overcomplicate things, they assume a basic dispatch rule: whenever an incident occurs, they send the ambulance that is closest to the incident (in time). Although this is a common dispatch policy, it was already shown to be suboptimal in 1972 [29]. Regardless, most authors make this assumption without much discussion or motivation.

The relatively few papers that have discussed alternatives to the 'closest idle' rule generally do so using simple, pragmatic rules. For example, [20] divides the region into separate subregions, and each subregion has a list of stations from which a vehicle should preferably depart. Another example is [109], which compares two different dispatch rules; the so-called 'closest-ambulance response' versus 'regionalized response'. Under regionalized response, each ambulance serves its own region first, even if it is temporarily outside its region. Only if it is unavailable, the closest idle ambulance is sent. However, note that both examples still ignore important information: the outcome does not depend on whether some regions remain uncovered after the dispatch is performed.

There exists a series of papers that considers a dispatch problem with prioritized patients [4, 80, 81]. Their main idea is to allow increased response times for the non-urgent patients, such that shorter response times can be realized for the urgent patients. Although this approach makes sense from a practical point of view, categorizing patients by their priority level is not the goal of this chapter. Instead, we assume that *all* patients have high priority, and investigate how to dispatch vehicles in order to maximize the fraction of arrivals within a response time threshold.

We seek for a policy that dispatches an ambulance such that remaining idle vehicles are in a good position with respect to expected incidents in the near future. This ensures that future incidents get a larger likelihood of being reached in time, thereby increasing the total expected fraction of incidents that can be reached within the time threshold. It should be clear that an EMS system can benefit from an improved dispatch policy, but since the topic has been underexposed in current literature, it is still unknown how much benefit can be expected. Furthermore, a dispatch policy can be combined with a relocation rule to realize even larger improvements in the objective value.

This chapter introduces two methods for ambulance dispatching, the first of which is a Markov Decision Problem (MDP). We mainly focus on minimizing the fraction of arrivals later than a target time - a typical objective in ambulance planning. However, we show that with a small change, our model can also minimize the average response time. A few authors have previously used MDPs to solve the dispatch problem. In [56] the authors define 'costs' in their MDP, but they do not discuss the meaning or interpretation of this. In their numerical work, they use randomly drawn instances. Moreover, they do not compare their solution with the closest-idle method. The authors of [4] maximize patient survivability, and furthermore use extremely small problem instances: two vehicles and two or three demand nodes. We conclude that neither [56] nor [4] analyzes the fraction of late arrivals. Second, we propose a heuristic for ambulance dispatching that behaves similarly to the policy obtained from our MDP. However, it is able to determine more accurately what the response time would be when dispatching a driving ambulance. Furthermore, the heuristic can be computed in polynomial time, which allows us to apply it to regions with a large number of vehicles.

We validate both our policies by a discrete-event simulation model of an urban EMS region. These simulations indicate that our proposed dispatch heuristic can decrease the fraction of late arrivals by as much as 18% relatively compared to the closest idle ambulance dispatch method. Our results provide a better understanding of dispatch policies and their potential to improve the objective. In the field of ambulance management, an improvement of 18% is considered large. However, it should be noted that there is a tradeoff: our policy significantly increases the average response time. Although we do not advise all EMS managers to immediately discard the closest idle dispatch method, we do show that the typical argument - that it would not lead to large improvements in the fraction of late arrivals - should be changed.

The rest of this chapter is structured as follows. In Section 2.2, we give a formal problem definition. In Section 2.3, we present our proposed solution using MDPs, followed by a solution based on a scalable heuristic in Section 2.4. We show our results for a small, intuitive region in Section 2.6 and in two more realistic case studies for the area of Utrecht in Section 2.7. We end with a discussion in Section 2.8.

2.2 Problem formulation

Define the set V as the set of locations at which incidents can occur. Note that these demand locations are modeled as a set of discrete points. Incidents at locations in V occur according to a Poisson process with rate λ.[1] Let d_i be the fraction of the demand rate λ that occurs at node i, $i \in V$. Then, on a smaller scale, incidents occur at node i with rate λd_i.

Let A be the set of ambulances, and $A_{idle} \subseteq A$ the set of currently idle ambulances. When an incident has occurred, we require an idle ambulance to immediately drive to the scene of the incident. The decision which ambulance to send has to be made at the moment we learn about the incident, and is the main question of interest in this chapter. When an incident occurs and there are no idle ambulances, the call goes to a first-come first-served queue. Note that when an incident occurs and an ambulance is available, it is not allowed to postpone the dispatch. Although in some practical situations dispatchers may queue a low priority call when the number of idle servers is small, in this chapter we focus on the most urgent incidents, which require service immediately.

Our objectives are formulated in terms of response times (see Figure 1.1). We want to minimize the fraction of incidents for which the response time is larger than T. Another observation is that we want response times to be short, regardless of whether they are smaller or larger than T. We translate this into a separate objective, which is to minimize the average response time. We assume that the travel time $\tau_{i,j}$ from node i to j $(i, j \in V)$ is deterministic, and known in advance.

[1] We will discretize the arrival process in the next section.

V	The set of demand locations.
H	The set of hospital locations, $H \subseteq V$.
A	The set of ambulances.
A_{idle}	The set of idle ambulances.
W_a	The base location for ambulance a, $a \in A$, $W_a \in V$.
T	The time threshold.
λ	incident rate per minute.
d_i	The fraction of demand in i, $i \in V$.
$\tau_{i,j}$	The driving time between i and j with siren turned on, $i, j \in V$.

Table 2.1 Notation.

Sending an ambulance to an incident is followed by a chain of events, most of which are random. When an ambulance arrives at the incident scene, it provides service for a certain random time $\tau_{onscene}$. Then it is decided whether the patient needs transport to a hospital. If not, the ambulance immediately becomes idle. Otherwise, the ambulance drives to the nearest hospital in the set $H \subseteq V$. Upon arrival, the patient is transferred to the emergency department, taking a random time $\tau_{hospital}$, after which the ambulance becomes idle.

An ambulance that becomes idle may be dispatched to another incident immediately. Alternatively, it may return to its base location. Throughout this chapter, we will assume that we are dealing with a *static* ambulance system, i.e., each ambulance has a fixed, given base and may not drive to a different base. However, it is possible that multiple ambulances have the same base location. We denote the base location of ambulance a by W_a, for $a \in A$. An overview of the notation can be found in Table 2.1.

2.3 MDP-based solution

In this section we model the ambulance dispatch problem as a *discrete-time* MDP. In each state s (further defined in Section 2.3.1), we must choose an action from the set of allowed actions, denoted as $\mathcal{A}_s \subseteq \mathcal{A}$, described in detail in Section 2.3.2. The process evolves in time according to transition probabilities that depend on the chosen actions (as described in Section 2.3.4 below). We are dealing with an infinite planning horizon, and our goal is to maximize the average 'reward'. We eventually find our solution by performing value iteration [94]. Our choice to use value iteration was motivated by it being simple in implementation, and sufficient to answer our central question on the closest idle policy.

In our model, we assume that at most one incident occurs within a time step. Therefore, the smaller the time steps, the more accurate the model will be. However, there is a tradeoff, as small time steps will increase the computation time. Throughout this chapter, we take time steps to be one minute, which balances the accuracy and the computation time.

2.3.1 State space

When designing a state space, it is important to store the most crucial information from the system in the states. However, when dealing with complex problems - such as real-time ambulance planning - it is tempting to store large amounts of information, resulting in an intractable state space. This would lead to the so-called curse of dimensionality [13], which makes it impossible to solve the problem with well-known MDP approaches.

As discussed before, there is little previous work on how to choose a good dispatch policy, but to some extent we can draw parallels with work on dynamic ambulance redeployment (which relocates idle vehicles): some researchers overcome the problem of an intractable state space by turning to Approximate Dynamic Programming (ADP), which allows for an elaborate state space to be solved approximately [77]. Alternatively, some researchers choose a rather limited state space, for example, by describing a state merely by the *number* of idle vehicles [2].

For the purpose of determining *which* ambulance to send, it is important to know whether the ambulance we might send will arrive within T time units. Therefore, it is crucial to know where the incident took place. Furthermore, we require some knowledge of where the idle ambulances are. Clearly, storing only the number of idle vehicles would be insufficient. However, storing the location of each idle ambulance would already lead to an intractable state space for practical purposes. Instead, we can benefit from the fact that we are trying to improve a *static* solution. In a static solution, the home base for any ambulance is known in advance. Note that an idle ambulance must be either residing at its base location, or travelling towards the base. Hence, if we allow for an inaccuracy in the location of idle ambulances, in the sense that we use their destination rather than their actual location, their location does not need to be part of the state. Merely keeping track of a simple status for each ambulance (idle or not), now suffices. Thereto, let $stat_i$ denote this status for ambulance i:

$$stat_i \in \{idle, busy\}, \qquad i \in A.$$

This leads us to a state s, defined as follows:

$$(Loc_{acc}, stat_1, stat_2, \dots, stat_{|A|}), \tag{2.1}$$

where Loc_{acc} denotes the location of the incident that has just occurred in the last time step. In case no incident occurred in the last time step, we denote this by a dummy location, hence

$$Loc_{acc} \in V \cup \{0\}.$$

This leads to a state space of size $(|V| + 1)2^{|A|}$. For future reference, let $Loc_{acc}(s)$ denote the location of the incident that has occurred in the previous time step when the system is in state s. For ease of notation, we introduce boolean variables $idle_i(s)$ and $busy_i(s)$ to denote whether $stat_i$ is idle or busy in state s, $i \in A$, $s \in S$.

2.3.2 Policy definition

In general, a policy Π can be defined as a mapping from the set of states to a set of actions: $S \to \mathcal{A}$. In our specific case, we define $\mathcal{A} = A \cup \{0\}$; that is if $\Pi(s) = a$, for $a \in A$, ambulance a should be sent to the incident that has just occurred at $Loc_{acc}(s)$. Action 0 may be interpreted as sending no ambulance at all (this is typically the choice when no incident occurred in the last time step, or when no ambulance is available). In a certain state, not all actions are necessarily allowed. Denote the set of feasible actions in state s as

$$\mathcal{A}_s \subseteq \mathcal{A}, \quad s \in S.$$

For example, it is not possible to send an ambulance that is already busy with another incident. This implies

$$busy_a(s) \to a \notin \mathcal{A}_s, \quad a \in A, \quad s \in S. \tag{2.2}$$

Furthermore, let us require that when an incident has taken place, we must always send an ambulance - if any are idle.

$$\exists a \in A : idle_a(s) \;\wedge\; Loc_{acc}(s) \neq 0 \to 0 \notin \mathcal{A}_s, \quad s \in S. \tag{2.3}$$

Moreover, if no incident has occurred, we may simplify our MDP by requiring that we do not send an ambulance:

$$Loc_{acc}(s) = 0 \to \mathcal{A}_s = \{0\}, \quad s \in S. \tag{2.4}$$

All other actions from $\mathcal{A}$ that are not restricted by (2.2)–(2.4) are feasible. This completely defines the allowed action space for each state.

2.3.3 Rewards

In ambulance planning practice, a typical goal is to minimize the fraction of late arrivals. Since our decisions have no influence on the number of incidents, this is equivalent to minimizing the *number* of late arrivals. An alternative goal might be to minimize average response times. Our MDP approach may serve either of these objectives, simply by changing the reward function. Define $R(s, a)$ as the reward received when choosing action a in state s, $s \in S$, $a \in \mathcal{A}_s$. Note that in this definition, the reward does not depend on the next state. Keep in mind that our goal is to maximize the average rewards.

Fraction of late arrivals
To minimize the fraction of late arrivals, i.e., the fraction of incidents for which the response time is larger than T, we define the following rewards:

$$R(s,a) = \begin{cases} 0 & \text{if } Loc_{acc}(s) = 0; \\ -N & \text{if } Loc_{acc}(s) \neq 0 \wedge a = 0, \text{ i.e., no idle ambulances;} \\ 0 & \text{if } Loc_{acc}(s) \neq 0 \wedge a \in A \wedge \tau_{W_a,\, Loc_{acc}(s)} \leq T; \\ -1 & \text{otherwise.} \end{cases}$$

Here N is a number that is typically larger than 1. We discuss the choice of this parameter further in Section 2.3.6.

Average response time
To minimize the average response time, one may use the same MDP model, except with a different reward function. Let M be a large enough number, typically such that $M > \tau_{i,j},\ i, j \in V$. Then we can define the rewards as follows.

$$R(s,a) = \begin{cases} 0 & \text{if } Loc_{acc}(s) = 0; \\ -M & \text{if } Loc_{acc}(s) \neq 0 \wedge a = 0, \text{ i.e., no idle ambulances;} \\ -\tau_{W_a,\ Loc_{acc}(s)} & \text{if } Loc_{acc}(s) \neq 0 \wedge a \in A. \end{cases}$$

2.3.4 Transition probabilities

Denote the probability of moving from state s to s', given that action a is chosen, as:

$$p^a(s,s'), \qquad a \in \mathcal{A}_s, \qquad s, s' \in S.$$

To compute the transition probabilities, note that the location of the next incident is independent of the set of idle ambulances. Thereto, $p^a(s,s')$ can be defined as a product of two probabilities. We write

$$p^a(s,s') = P_1(s') \cdot P_2^a(s,s'),$$

which stands for the product of the probability that an incident happened at a specific location (P_1), and the probability that specific ambulances became available (P_2), respectively.

First of all, let us define $P_1(s')$. Since incidents occur according to a Poisson process, we can use the arrival rate λ (the probability of an arrival anywhere in the region per discrete time step) to obtain

$$P_1(s') = \begin{cases} \lambda \cdot d_{Loc_{acc}(s')} & \text{if } Loc_{acc}(s') \in V; \\ 1 - \lambda & \text{else.} \end{cases}$$

Note that the occurrence of incidents does not depend on the previous state (s).

Secondly, we need to model the process of ambulances that become busy or idle. For tractability, we will define our transition probabilities as if ambulances become idle according to a geometric distribution. In reality - and in our verification of the model - this is not the case, but since our objective is the long term average cost, this modelling choice leads to the same performance. Let us define a parameter $r \in [0, 1]$, which represents the rate at which an ambulance becomes idle. We discuss the parameter choice for r in Section 2.3.6.

We include a special definition to cover the case where an ambulance was just dispatched. In such a case, the ambulance cannot be idle in the next time step. Furthermore, ambulances do not become busy, unless they have just been dispatched. We now define

$$P_2^a(s,s') = \Pi_{i=1}^{|A|} P^a_{change}\big(stat_i(s), stat_i(s')\big), \qquad s,s' \in S,$$

where

$$P^a_{change}\big(stat_i(s), stat_i(s')\big) = \begin{cases} 1 & \text{if } a = i \wedge busy_i(s'); \\ 0 & \text{if } a = i \wedge idle_i(s'); \\ r & \text{if } a \neq i \wedge busy_i(s) \\ & \wedge\ idle_i(s'); \\ 1-r & \text{if } a \neq i \wedge busy_i(s) \\ & \wedge\ busy_i(s'); \\ 0 & \text{if } a \neq i \wedge idle_i(s) \\ & \wedge\ busy_i(s'); \\ 1 & a \neq i \wedge idle_i(s) \\ & \wedge\ idle_i(s'). \end{cases} \tag{2.5}$$

2.3.5 Value iteration

Now that we have defined the states, actions, rewards and transition probabilities, we can perform value iteration to solve the MDP. Value iteration, also known as backward induction, calculates a value $V(s)$ for each state $s \in S$. The optimal policy, i.e., the best action to take in each state, is the action that maximizes the expected value of the resulting state s'. $V(s)$ is calculated iteratively, starting with an arbitrary value $V_0(s) \ \ s \in S$. (In our case, we start with $V_0(s) = 0 \ \ s \in S$.) In each iteration i, one computes the values $V_i(s)$ given $V_{i-1}(s) \ \ s \in S$ as follows:

$$V_i(s) := \max_{a \in \mathcal{A}_s} \{ \sum_{s'} p^a(s,s')(R(s,a) + V_{i-1}(s')) \}. \tag{2.6}$$

This is known as the *Bellman equation* [12].

When the span of V_i (i.e., $\max V_i(s) - \min V_i(s)$) converges, the left-hand side becomes equal to the right-hand side in Equation (2.6), except for an additive constant. After this convergence is reached, the value of $V(s)$ is equal to $V_i(s) \ \ s \in S$. Note that the MDP we defined is unichain. Hence, value iteration is guaranteed to converge.

Small regions, such as the region in Section 2.6, allow us to reach convergence and accurately determine the value function V. However, for larger regions (such as Utrecht in Section 2.7.1), value iteration simply takes too much time to reach convergence. Instead, we use the non-converged values V_i and analyze the performance of the corresponding policy.

2.3.6 Parameter choices

Recall that $-N$ is the reward given in the situation that there occurs an incident while all ambulances are busy, in the MDP that attempts to minimize the fraction of late arrivals. If $N > 1$, this implies that when all ambulances are busy, the rewards are smaller than when we send an ambulance that takes longer than T to arrive. This is in agreement with the general idea that having no ambulances available is a very bad situation. One might be tempted to make the reward for the only possible action ($a = 0$) in these states even smaller than we did, in order to influence the optimal actions in other states: the purpose would be to steer the process away from states with no ambulances available. However, note that this would not be useful, because our actions do not affect how often we end up in a state where all ambulances are busy. This is merely determined by the outcome of an external process, i.e., an unfortunate sequence of incidents. Therefore, an extremely small reward for action $a = 0$ in states where all ambulances are busy, would only blur the differences between rewards for actions in other states. In our numerical experiments, we use $N = 5$.

For the MDP that minimizes the average response times, the reward given in the situation that there occurs an incident while all ambulances are busy is given by $-M$. In our numerical experiments, we use $M = 15$ for the small region, and $M = 30$ for the region Utrecht. In our implementation, time steps are equal to minutes.

Recall that r is the rate at which an ambulance becomes idle. We should set it in such a way, that the expected duration is equal to the average in practice. So this includes an average travel time, and an average time spent on scene. We add an average driving time to a hospital to that, as well as a realistic hospital drop off time - both multiplied with the probability that a patient needs to go to the hospital. For Dutch ambulances, this results in an average of roughly 38 minutes to become available after departing to an incident. For the geometric distribution, we know that the maximum likelihood estimate $\hat{r}$ is given by one divided by the sample mean. In this case, $\hat{r} = \frac{1}{38} \approx 0.0263$, which we use as the value for r in our numerical experiments.

2.4 Heuristic solution

In this section we describe a dispatch heuristic that is easy to implement and scales well. It can be computed in real time, for any number of vehicles and ambulance bases that is likely to occur in practice. The general idea is that, at any time, we can calculate the *coverage* provided by the currently idle ambulances. This results in a number that indicates how well we can serve the incidents that might occur in the (near) future. More specifically, coverage is defined as in the MEXCLP model [36], that we will describe next.

2.4.1 Coverage according to the MEXCLP model

In this section we briefly recap the objective of the well-known MEXCLP model (see Section 1.2.1. MEXCLP was originally designed to optimize the distribution of a set of ambulances (denoted A) over a set of possible base locations. The objective is to maximize the total coverage of the region, which can be written as:

$$\text{Maximize } \sum_{i \in V} \sum_{k=1}^{|A|} d_i(1-q)q^{k-1}y_{ik}.$$

For the definitions of the variables and parameters we refer to Section 1.2.1. The original LP formulation in [36] requires several constraints to ensure that the variables y_{ik} are set in a feasible manner. For our purpose, we do not need these constraints, as we shall determine how many ambulances are within reach of our demand points - the equivalent of y_{ik} - in a different way.

2.4.2 Applying MEXCLP to the dispatch process

The dispatch problem requires us to decide which (idle) ambulance to send, at the moment an incident occurs. Thereto, we compute the *marginal* coverage that each ambulance provides for the region. The ambulance that provides the smallest marginal coverage is the best choice for dispatch in terms of remaining coverage for future incidents. However, this does not incorporate the desire to reach the current incident within target time T. We propose to combine the two objectives - reaching the incident in time and remaining a well-covered region - by always sending an ambulance that will reach the incident in time, if possible. This still leaves a certain amount of freedom in determining which *particular* ambulance to send.

The computations require information about the location of the (idle) ambulances. Denote this by $Loc(a)$ for all $a \in A_{idle}$. We evaluate two different options for $Loc(a)$, that we describe next.

Using actual positions of ambulances is the most accurate information one could use. In practice, $Loc(a)$ may be determined by GPS signals. For simulation purposes, the current position of the ambulance while driving may be determined using, e.g., interpolation between the origin and destination, taking into account the travel speed. In either case, the location should be rounded to the nearest point in V, because travel times $\tau_{i,j}$ are only known between any $i, j \in V$.

Using destinations of ambulances is a far simpler, albeit somewhat inaccurate alternative. The simplicity, however, does make it a practical and accessible option. When determining $Loc(a)$, simply take the destination of ambulance a. This is a good option, e.g., when no - or not enough - GPS information is available. Furthermore, this solution has a certain fairness in comparison to the MDP solution in Section 2.3, which is also required to make decisions based on the destinations of ambulances.

Let A^+_{idle} denote the set of idle ambulances that are able to reach the incident in time, i.e., the ambulances $a \in A_{idle}$ for which $\tau_{Loc(a),i} \leq T$ (where i denotes the incident location). Note that this definition depends on how $Loc(a)$ was chosen: when based on the true locations of ambulances, the set A^+_{idle} can be determined correctly. When one uses the destinations of ambulances, the decision of which ambulances are in A^+_{idle} may contain errors: some ambulances may in fact be closer to the incident than they appear (because they are driving towards a base that is further away from the incident), or the other way around they may in reality be further away from the incident than Loc_a suggests.

Similarly, let A^-_{idle} denote the set of idle ambulances that cannot reach the incident in time, which implies that $A^+_{idle} \cup A^-_{idle} = A_{idle}$. Then, if $A^+_{idle} \neq \emptyset$, we decide to dispatch a vehicle that will arrive within the threshold time, but chosen such that the coverage provided by the remaining idle vehicles is as large as possible:

$$\arg\min_{x \in A^+_{idle}} \sum_{i \in V} d_i (1-q) q^{k(i, A_{idle})-1} \cdot \mathbb{1}\{\tau_{Loc(x),i} \leq T\}. \tag{2.7}$$

Otherwise, simply dispatch a vehicle such that the coverage provided by the remaining idle vehicles is as large as possible (without requiring an arrival within the threshold time):

$$\arg\min_{x \in A^-_{idle}} \sum_{i \in V} d_i (1-q) q^{k(i, A_{idle})-1} \cdot \mathbb{1}\{\tau_{Loc(x),i} \leq T\}. \tag{2.8}$$

Note that in our notation, k is a function of i and A_{idle}. $k(i, A_{idle})$ represents the number of idle ambulances that are currently within reach of vertex i. After choosing the locations of ambulances that one wishes to use - the real locations or the destinations - $k(i, A_{idle})$ can be counted in a straightforward manner.

We have seen that the way one measures the location of ambulances - either the true location or just the destination - affects the definition of the set A^+_{idle} (resp. A^-_{idle}), and thereby also the number $k(i, A_{idle})$ in Equation 2.8. There is, however, one more aspect that is affected by the location of the ambulance: this is incorporated in $\mathbb{1}_{\tau_{Loc(x),i} \leq T}$ in Equation 2.8. Hence, using the destination of ambulances results in a small error in three different places. It is reasonable to assume that using the destinations of ambulances performs worse than using the real locations, but the magnitude of the performance difference is hard to oversee beforehand. Instead, we will show the performance difference in retrospect in our numerical examples in Sections 2.6, 2.7.1 and 2.7.2.

2.5 Simulation model

To compare the results of different policies, we measure their performance using simulation. All results mentioned, including the fraction of late arrivals and the average response times, are estimates based on the observed response times in our simulation model. This section explains the need for simulation to verify the results from the MDP.

The reason for using simulation is that the EMS process is rather complex. The aforementioned MDP does not capture all details and is therefore not able to estimate the performance accurately. We will next highlight the two main differences between the MDP and the simulation.

One reason why the MDP is not entirely accurate is that incidents that occur while no vehicles are available are 'lost'. This assumption is made to improve scalability: it avoids the need to expand the state with a queue of calls that are waiting for vehicles to become idle. However, counting these calls as lost is technically incorrect for two reasons. First of all, an ambulance might become available shortly after, and it is - although unlikely - still possible that it arrives within the time threshold. Second, a lost call in the MDP is not counted in the total workload, which leads to an overestimation in the number of idle vehicles in the time steps shortly after the lost call. In our simulation, we place the incidents that arrive while all vehicles are busy in a first come first serve queue. Ambulances that become idle are immediately dispatched to a waiting incident (if any), or else head back to their home base.

Our simulations are also able to handle the complex course of events that take place when an ambulance is dispatched while on the road. Such vehicles are typically returning to the home base, already idle and ready to serve an incoming call. Our simulation computes the current location of the vehicle based on an interpolation between the origin (typically a hospital where a patient was just dropped off) and the destination (the vehicle's home base) of the trip, taking into account the total time of that particular trip. The MDP is unable to distinguish between idle vehicles on the road and vehicles at the base. Adding on-the-road information to the MDP would require a state definition that includes (at least) the drop off location of the last patient. This alone would already lead to a state space explosion and therefore we do not recommend solving this for realistic instances.

In our simulation, $\tau_{onscene}$ is exponentially distributed with an expectation of 12 minutes. $\tau_{hospital}$ is drawn from a Weibull distribution with an expectation of 15 minutes. In our simulations, patients need hospital treatment with probability 0.8. This value was estimated from Dutch data [112]. (Similar numbers (78% nation-wide) can be deduced from [89].) Note that none of these parameters are explicitly part of our solution methods. Instead, they subtly affect the busy fraction q (for the heuristic) or the transition probabilities with rate r (for the MDP).

2.6 A motivating example

In this section, we consider a small region for which there is some intuition with respect to the best dispatch policy. We show that the intuitive dispatch policy that minimizes the fraction of late arrivals is in fact obtained by both our solution methods (based on MDP and MEXCLP). We will address the alternative objective - minimizing the average response times - as well.

Figure 2.1 shows a small example for demonstrative purposes. We let calls arrive according to a Poisson process with on average one incident per 45 minutes. Furthermore, incidents occur w.p. 0.1 in Town 1 and w.p. 0.9 in Town 2. 80 percent of all incidents require transport to the hospital, which is located in Town 2.

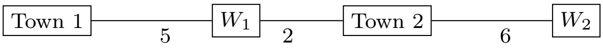


Figure 2.1 A graph representation of the region. The numbers on the edges represent the driving times in minutes with siren turned on. W_1 and W_2 represent the base locations of ambulance 1 and 2, respectively. Incidents occur only in Town 1 and Town 2. There is a single hospital, located in Town 2.

2.6.1 Fraction of late arrivals

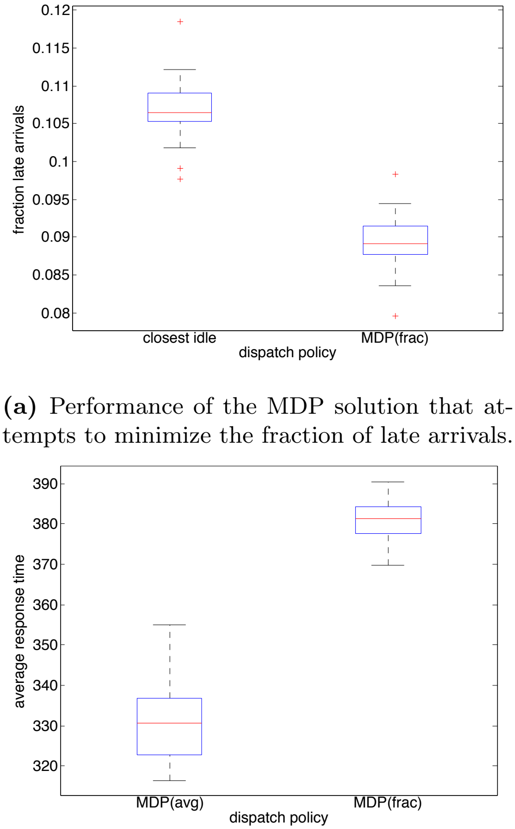
This section deals with minimizing the fraction of response times exceeding twelve minutes. A quick analysis of the region in Figure 2.1 leads to the observation that the 'closest idle' dispatch strategy must be suboptimal. To serve as many incidents as possible within twelve minutes, it is evident that the optimal dispatch strategy should be as follows: when an incident occurs in Town 2, send ambulance 2 (if available). When an incident occurs in Town 1, send ambulance 1 (if available). Both the MDP solution that attempts to minimize the fraction of late arrivals (with, e.g., $N = 5$), as well as the dispatch heuristic based on MEXCLP, lead to this policy.

The response times obtained by simulating the closest-idle policy and MDP (frac) solution are compared in Figure 2.2a. This clearly shows that the MDP solution outperforms the closest idle method, as was expected.

Note that in our model it is mandatory to send an ambulance if at least one is idle. Furthermore, our proposed solutions do not base their decision on the locations of idle ambulances (instead, we pretend they are at their destination, which is fixed for each ambulance). Therefore, in this example with two ambulances, one can describe a dispatch policy completely by defining which ambulance to send when both are idle, for each possible incident location. For an overview of the various policies, see Table 2.2.

Solution method	Loc_{acc} = Town 1	Loc_{acc} = Town 2
MEXCLP(dest)	W_1	W_2
MDP(frac)	W_1	W_2
MDP(avg)	W_1	W_1

Table 2.2 An overview of the various dispatch policies when both ambulances are idle. The value in the table represents the base from which an ambulance should be dispatched.



(a) Performance of the MDP solution that attempts to minimize the fraction of late arrivals.

(b) Average response time (in seconds) for two MDP solutions with different objectives.

Figure 2.2 Box plots showing the performance as observed in a simulation of the small region. Each policy was evaluated with twenty runs of 5,000 simulated hours. The red plus-signs indicate outliers.

As shown in this table, the MDP solution minimizing the fraction of late arrivals - in this particular instance - comes down to exactly the same policy as the MEXCLP dispatch heuristic using destinations of vehicles. Therefore, the results mentioned for either of those two policies, also hold for the other. For this problem instance the closest-idle dispatch method turns out to be roughly equivalent with the MDP solution minimizing the average response time (except for the fact that the MDP can only use destinations of vehicles, whereas closest-idle uses their true positions).

2.6.2 Average response time

We used the MDP method described in Section 2.3.3 to obtain a policy that should minimize the average response time. Let us denote this policy by MDP(avg). We evaluate the performance of the obtained policy, again by simulating the EMS activities in the region. These simulations show that the MDP solution indeed reduces the average response time significantly, compared to the policy that minimizes the fraction of late arrivals, denoted MDP(frac), see Figure 2.2b.

2.7 Computational results

In this section, we validate our redeployment method on two realistic problem instances. Both instances are based on the county of Utrecht, which is hosted by one of the largest ambulance providers of the Netherlands. Utrecht is a densely populated area, with approximately 1.2 million inhabitants and an area of approximately 1,400 square kilometers. The ambulance provider for this region handles more than 100,000 incidents per year - a number equal to roughly 10% of all ambulance demand in the Netherlands.

parameter	**magnitude**	**choice**
V	217	4 digit postal codes.
H	10	The hospitals within the region in 2013, excluding private clinics.
$\tau_{i,j}$		Driving times as estimated by the RIVM.
d_i		Fraction of inhabitants as known in 2009.

Table 2.3 Parameter choices for our implementation of the region of Utrecht.

The area contains several cities, including Amersfoort and Utrecht city. However, the whole region may - by international standards - be considered an urban area. The two problem instances differ in two ways: the number of vehicles and the incident arrival rate. We consider one problem instance with eight vehicles, for which we can compare the MDP and the heuristic. The second problem instance has nineteen vehicles, which only allows us to compute the results for the heuristic. Apart from this, the problem instances use the same model for the region, which we summarize in Table 2.3. Utrecht is a region with multiple hospitals, and for simplicity we assume that the patient is always transported to the nearest hospital, if necessary.

Note that we used the fraction of inhabitants as our choice for d_i. In reality, the fraction of demand could differ from the fraction of inhabitants. However, the number of inhabitants is known with great accuracy, and this is a straightforward way to obtain a realistic setting. Furthermore, the analysis of robust optimization for uncertain ambulance demand in [61] indicates that we are likely to find good solutions, even if we use inaccurate estimates for d_i.

In the Netherlands, the time target for the highest priority emergency calls is 15 minutes. Usually, three minutes are reserved for answering the call, therefore we choose to run our simulations with $T = 12$ minutes. The driving times for EMS vehicles between any two nodes in V were estimated by the Dutch National Institute for Public Health and the Environment (RIVM) in 2009 [66, Chapter 3]. The RIVM uses measurements of a full year of ambulance movements for this, and differentiates between road type, region and time of day. The driving times we use are estimates for ambulance movements with the siren turned on, at the time of day with most traffic congestion. Therefore, they could be considered a pessimistic or safe approximation. Note that these travel times are deterministic. For ambulance movements without siren (e.g., when repositioning) we used 0.9 times the speed with siren.

2.7.1 Region Utrecht with eight vehicles

parameter	**magnitude**	**choice**
A	8	Small enough for a tractable MDP.
λ	1/15	A reasonable workload for 8 ambulances.
W_a $(a \in A)$		Postal codes 3582, 3645, 3958, 3582, 3991, 3447, 3811, 3417.

Table 2.4 Parameter choices for our implementation of the region of Utrecht.

In this section, we do a case study for the region Utrecht with eight vehicles. This number is small enough such that the MDP is still tractable. For the parameters used in the implementation, see Table 2.4. The locations we used as home bases are depicted in Figure 2.3, and correspond to actual base locations in the EMS region.

For this problem instance, one value iteration takes approximately 70 minutes to calculate on a 2.4 GHz Intel Core i5. Although this seems to be rather long, we emphasize that these calculations take place in a preparatory phase. We perform 21 iterations after which the current solution is used as policy. After these calculations, the final policy constitutes a lookup table for which online decision making can be done without additional computation time.

Analysis of the MDP solution

In this section, we highlight some features of the MDP solution that attempts to minimize the fraction of late arrivals for the region Utrecht. In particular, we focus on the states for which the MDP solution differs from the closest idle policy.

The output of the MDP is a table with the incident location, the status of the different ambulances (idle or not), and the optimal action. This output is a rather large table that does not easily reveal insight into the optimal policy. Therefore,

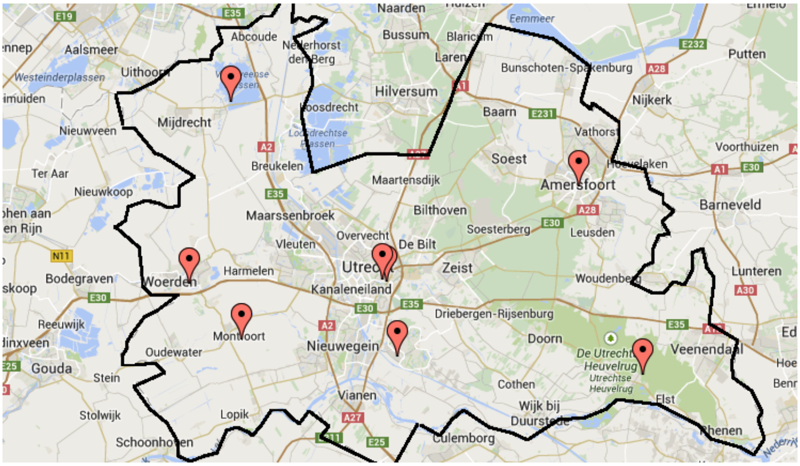


Figure 2.3 The home bases for each of the eight ambulances in region Utrecht. The chosen locations currently exist as base locations operated by the ambulance provider for this region. Note that in this figure, two vehicles are stationed at the base in the center of Utrecht.

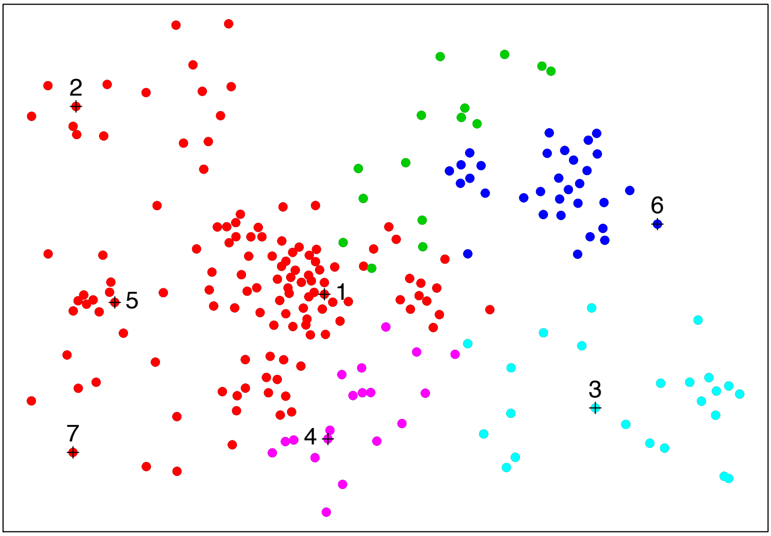


Figure 2.4 Each node represents a postal code in Utrecht. Nodes with the same colour have similar MDP solutions. The numbers indicate the ambulance bases. (Two vehicles are stationed at base number 1.)

we used classification and regression trees (referred to as CART trees) [68] on the table to find structure in the form of a decision tree. We used random forests to create the decision tree, since it is known that a basic CART has poor predictive performance (see [68, Chapter 14]). Another option is to use bagging (i.e., bootstrap aggregation) trees. This effectively generates several bootstrap samples of the original table, trains CART trees on the sample, and finally averages the results. While bagging trees reduces the variance in the prediction, random forests also cancel any correlation structure in the generation of the trees that may present while bagging.

The outcome that describes the best policy after 21 value iterations is a decision tree that divides the state space into five regions, see Figure 2.4. If an incident occurs in the red region, then in most cases the closest idle ambulance is dispatched. If the ambulance at base 4 is idle, this is even more often the case than when it is busy. The location of the base stations plays an essential role in the final decision tree.

For some nodes, whether or not the closest idle ambulance should be dispatched depends even more heavily on which ambulances are idle. For example, if an incident occurs in the dark blue region while the ambulance at base 6 is idle, the MDP tells us to almost always (in more than 98% of the states) send the closest idle ambulance. Conversely, if the ambulance at base 6 is busy, it is better to strategically choose a different ambulance instead of simply applying the closest idle policy.

This may be intuitively understood as follows. Generally speaking, the dark blue nodes can be reached within the time threshold from base 6, and only base 6. Therefore, if the ambulance at base 6 is busy, incidents on the dark blue nodes will not be reached in time. For those dark blue nodes, the next closest base is base 3. But dispatching this vehicle (if it is idle) will leave the entire East side of the region without idle ambulances. Therefore, it is in this case better to use an ambulance from the west side of the region. The enlarged response time is - using our objective of the fraction of late arrivals - not a downside, since the incident could not be reached in time anyway.

For incidents on the purple and cyan nodes, the best decision depends mostly on the state of the ambulance at base 3 and 6. If both ambulances are simultaneously busy, then the best ambulance to send to incidents in the purple region is usually the closest idle one. In the same scenario, incidents in the cyan region are typically *not* helped by the closest idle one. Note that this is the scenario when the entire East side of the region is not covered. This behaviour can be interpreted in a way similar to the case above (regarding dark blue nodes). When an incident cannot be reached in time, we might as well choose a vehicle other than the closest idle one. This can be beneficial, because the choice can be made such that the remaining ambulances are in a favourable position with respect to possible future incidents. Note that this is also the general idea that forms the basis of our MEXCLP dispatch heuristic.

Results

In this section, we show the results from our simulations of the EMS region of Utrecht. We ran simulations using four different dispatch policies: the closest idle method, the MEXCLP-based heuristic (using both destinations and real locations of vehicles) and the MDP solution after 21 value iterations. Figure 2.5 compares their performance in terms of the observed fraction of response times larger than the threshold time.

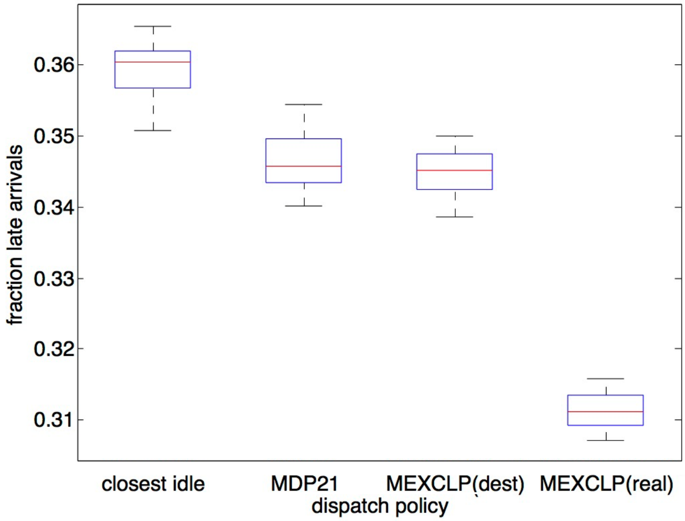


Figure 2.5 Comparing the performance of the MDP solution after 21 value iterations, with two variants of the Dynamic MEXCLP dispatch method (where $q = 0.3$). The benchmark is the 'closest idle' policy. Each policy was evaluated with twenty runs of 5,000 simulated hours.

The results show that the MDP solution that was designed to minimize the fraction of late arrivals has approximately the same performance as the MEXCLP-based dispatch heuristic that uses the destinations of vehicles. Both policies perform better (on average) than the 'closest idle' policy. In addition, the MEXCLP-based dispatch heuristic that uses the real locations of vehicles performs even better.

For the region Utrecht with eight ambulances, value iteration took a long time to converge. Instead of waiting for convergence, we applied the policy we get after a fixed number of value iterations. Figure 2.6 indicates that the performance increases when we increase the number of value iterations.

Up until now we have focused on the fraction of late arrivals, a key performance measure in ambulance operations. However, other aspects of the response times can also be important. For example, it is considered a drawback if patients have to wait an extremely long time for their ambulance to arrive (i.e., the response time distribution is heavy tailed). In this example - as well as in others - there exist trade offs between performance indicators.

Next, we visualize the cumulative distribution of response times, as obtained from our simulation. Figure 2.7 shows - just like Figure 2.5 - that the MEXCLP

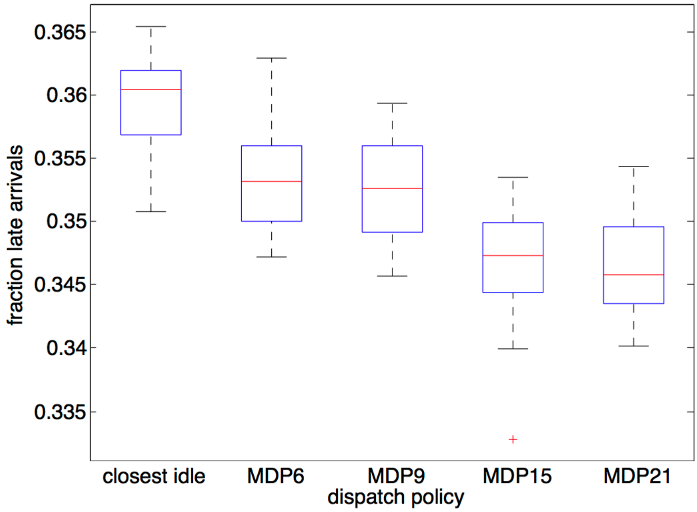


Figure 2.6 The performance of the MDP solution for region Utrecht after 6, 9, 15 and 21 value iterations. Each policy was evaluated with twenty runs of 5,000 simulated hours. The 'closest idle' dispatch policy is the benchmark.

heuristic outperforms the other policies for response times within the time threshold (720 seconds). However, it also shows significant differences in response times for response times greater than T, for which the MEXCLP heuristic performs *worse* than the benchmark.

How much one is willing to sacrifice on one performance indicator in order to realize an improvement in the other, is typically the source of a lively discussion. Although such choices depend on how the different aspects of the response times are weighted, we expect that in realistic cases decision makers will prefer the closest idle policy over the MEXCLP heuristic. The reason for this is in the tail of the response times (see Figure 2.7).

When taking a closer look at Figure 2.7, we make two other observations. First of all, the line of the MDP(frac)21 solution is very close to the MEXCLP(dest) line. Remember that these two policies base their decisions on the same information (that is, the destinations of idle vehicles and the location of a possible incident). This observation confirms our belief that these two policies have a similar underlying idea: they attempt to balance the response time for a *current* incident with the coverage for possible *future* incidents. Secondly, one may note that the line for the MDP(avg) solution is remarkably similar to the line for the closest idle method.

Sensitivity to the parameter q

The dispatch heuristic based on MEXCLP has an input parameter q, which represents the busy fraction. In this section we analyse the sensitivity of the

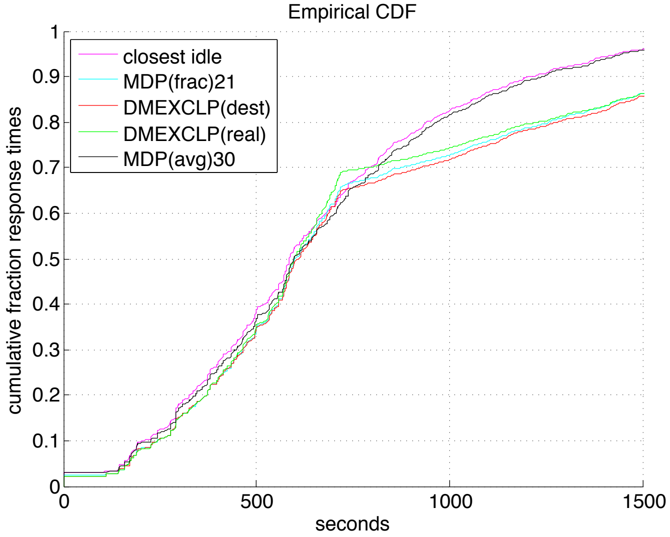


Figure 2.7 The cumulative distribution of response times observed in a simulation of 5,000 hours per dispatch policy, for the region Utrecht with eight ambulances. For the MEXCLP algorithms, a value of $q = 0.2$ was used. For the Markov Decision Problems, the notation MDP(objective)#iterations was used.

performance to the value of q that is used. Thereto, we simulated the EMS system several times for several values of q.

In theory, q should be equal to the true busy fraction throughout the system. However, one may observe different behaviour for different values of q, and the true busy fraction need not necessarily be the one with optimal performance. This may seem counter-intuitive at first, but fact is that dynamic ambulance management is such a difficult problem, that we cannot hope to find a model that captures everything perfectly. Generally speaking, using MEXCLP with $q \approx 0$ puts emphasis on covering the *next* incident. Using a higher busy fraction is equivalent with creating preparedness for incidents further into the future - at the cost of performing worse with respect to incidents in the near future. The true busy fraction could be a good starting point, but in practice one may choose a different value based on performance in simulations.

We simulated the EMS system of Utrecht, again with eight ambulances and (on average) four incidents per hour. We executed the MEXCLP dispatch heuristic for values of q between 0.1 and 0.8. The performance is shown in Figure 2.8. We observed that the true busy fraction throughout the simulations was between 37.5% and 38.1% (as measured when using $q = 0.2$ and $q = 0.8$, respectively).

First, analysis of Figure 2.8 suggests that $q = 0.2$ would be a good choice for this particular scenario: it seems to result in the lowest fraction of late arrivals. Second, note that q varies between 0.1 and 0.8 in this analysis, which are quite extreme values. In practice, discussions will typically be about smaller perturbations, e.g., should we use $q = 0.2$ or $q = 0.3$? Furthermore, it is also important to recognize the scale on the vertical axis, as well as the overlap in the boxes of the box plot. Recall that the performance of the benchmark (the closest idle policy) is approximately 36%, which is significantly worse than our heuristic for any value of q. We conclude that the MEXCLP dispatch method is fairly insensitive to the value of the parameter q.

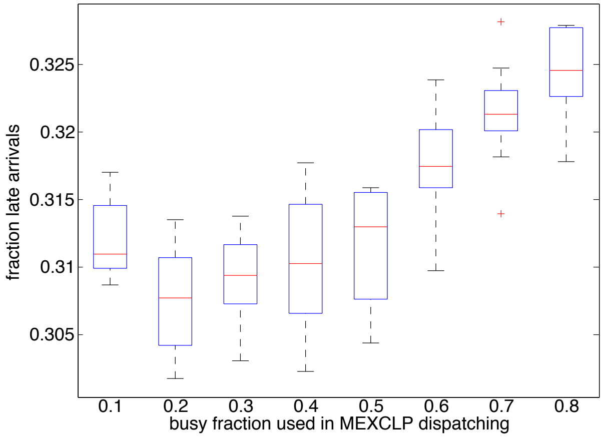


Figure 2.8 The performance observed for the MEXCLP dispatch heuristic, for different values of parameter q. Each box consists of 10 simulations of 5,000 hours each, for the region Utrecht with eight ambulances.

2.7.2 Region Utrecht with nineteen vehicles

In the previous section, we used eight vehicles in the region of Utrecht, due to the scaling limitations of our MDP solution. In this section, we analyze a more realistic representation of Utrecht: we increase the incident frequency to one incident per 10 minutes (on average). This is quite a reasonable estimate for this region during the summer period[2]. Simultaneously, we increase the total number of ambulances to nineteen. For the other simulation parameters, we use the same values as in Section 2.7.1.

We allow ambulances to be stationed only at locations that match the EMS base locations that exist in reality (using data from 2013). Throughout this section, we assign ambulances to the available bases according to the solution of the *static* MEXCLP model, which is generally assumed to give reasonable solutions (for a comparison of static methods, see [16]).

Figure 2.9 compares the performance of the MEXCLP dispatch heuristic with the benchmark (the closest idle policy). Note that the obtained fraction of late

[2] Our dataset for this region in the month of August 2008 shows 4775 urgent ambulance requests, which is on average 9.4 minutes between incidents.

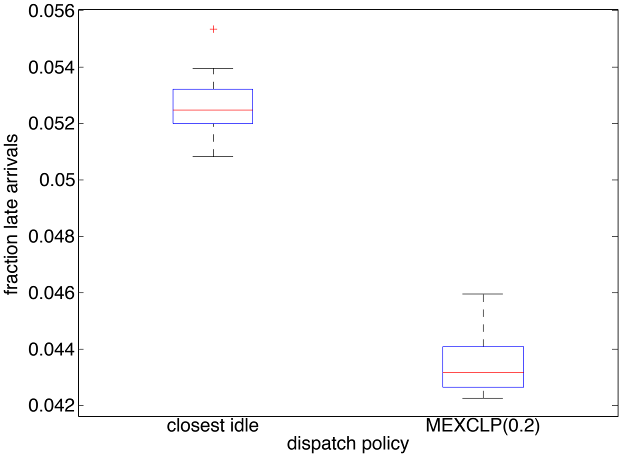


Figure 2.9 The objective (fraction of late arrivals), as observed in a simulation of 5,000 hours per dispatch policy, for the region Utrecht with 19 ambulances.

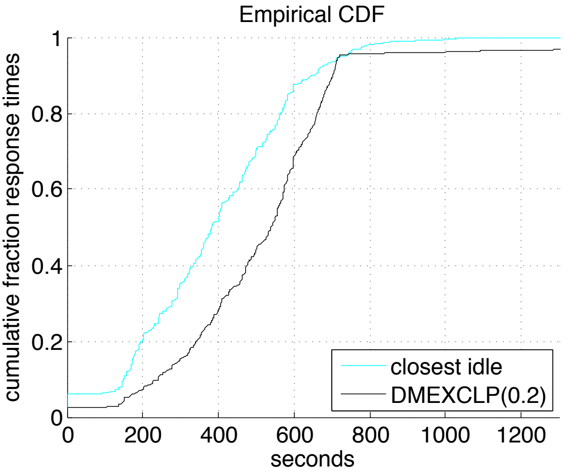


Figure 2.10 The cumulative distribution of response times, as observed in a simulation of 5,000 hours per dispatch policy, for the region Utrecht with 19 ambulances.

arrivals - roughly 5% - is realistic for this region in practice. The MEXCLP dispatch heuristic reduces the fraction of late arrivals from 0.053 to 0.043 (on average), a relative improvement of approximately 18%. To the best of our knowledge, no previous literature on ambulance dispatching has described a performance improvement of this magnitude - except perhaps for artificial problem instances that were designed for this purpose. Moreover, it was often assumed that changing the dispatch policy - as opposed to changing the position of idle vehicles - would not lead to major improvements (see, e.g., [118]). Our results shed new light on this belief. Note that an improvement of 18% is considered large, even with respect to algorithms that *are* allowed to reposition idle vehicles.

It should be clear that - when solely focusing on the fraction of late arrivals - the MEXCLP dispatch heuristic can offer great improvements compared to the closest idle policy. However, decision makers are often interested in more than just the fraction of late arrivals. They should be warned that changing from the closest idle dispatch policy to the MEXCLP heuristic comes at a price: it considerably diminishes the performance of other quality indicators, as can be seen in Figure 2.10. We highlight the difference in the average response time: when switching from the closest idle method to our heuristic, the average response time increased from 390 seconds to 535 seconds (an increase of 37%).

2.8 Discussion

This chapter provides new insight into the popular belief that deviating from the closest idle dispatch policy cannot greatly improve the objective (the expected fraction of late arrivals). We found an improvement of 18%, which was unexpectedly large. We consider this the main contribution of our work. Practitioners and researchers who define the fraction of late arrivals as their sole objective, should no longer claim that the closest idle method is near-optimal. Our methods yield in a great improvement in this KPI, however: one should be careful if one is also interested in other aspects of the response time. It is important to remember that our policies were designed with emphasis on the fraction of late arrivals only. Therefore, we do not claim that our dispatch policies are practically preferable over the closest idle policy, but we have shown that the argumentation for not using alternatives should be different. One should argue that we do not deviate from the closest idle policy, because we do not know how to do this while improving response times overall - and not because the alternatives fail to improve the fraction of late arrivals.

Next, we discuss the limitations of our work. Although it is possible to apply our MDP in practice for reasonably-sized ambulance fleets, we do not recommend it: computation times are rather long and the performance improvement is small. The MDP is in our opinion mostly of theoretical interest. On the other hand, the heuristic could very well be applied in practice, but decision makers should be aware of its side effects: the heuristic aims to minimize the fraction of late arrivals, which does not reduce - and can in fact increase - response times overall.[3] We recognize this as an important downside and emphasize that practitioners should carefully consider whether the response time threshold really is the way they want to evaluate their performance.

Finally, we adress some topics for further research. One might consider making small changes to the MDP that could benefit the performance. For example,

[3] Note that the same effect holds for the MDP that aims to minimize the fraction of late arrivals.

one idea is to artificially increase the rate with which busy ambulances become idle. This extra time would allow for ambulances to drive back to their home base, before the MDP considers them to be idle again. That way, we avoid the error where the MDP decides that an ambulance will reach an incident within the time threshold, but in fact the ambulance is still returning to base and happens to be further away from the incident. We suspect that this approach might give a small improvement; however, it should be noted that there is also a downside to making this change: ambulances are considered to be busy even though they are free, and hence suboptimal decisions will be made from time to time. In fact, sometimes an ambulance is *closer* than the MDP knows, because its previous patient was in the same area as the next patient. Other changes could be, to add more information in the state about the ambulance's actual location while driving back to the home base. This, however, would lead to a state space explosion and the resulting model will - for realistically-sized regions - most certainly not be solvable by value iteration.

3
Benchmarking online dispatch algorithms

Providers of EMS face the online ambulance dispatch problem, in which they decide which ambulance to send to an incoming incident. Their objective is to minimize the fraction of arrivals later than a target time. Today, the performance gap between existing solutions and the optimum is unknown, and we provide a bound for this gap.

Thereto, we propose a benchmark model (referred to as the *offline* dispatch model) to calculate the optimal dispatch decisions assuming that all incidents are known in advance. For this model, we introduce and implement three different methods to compute the optimal offline dispatch policy for problems with a finite number of incidents. The performance of the offline optimal solution serves as a bound for the performance of an - unknown - optimal online dispatching policy.

We show that the competitive ratio (i.e., the worst case performance ratio between the optimal online and the optimal offline solution) of the dispatch problem is unbounded; that is, even an optimal online dispatch algorithm can perform arbitrarily bad compared to the offline solution. Then, we performed benchmark experiments for a large ambulance provider in the Netherlands. The results show that for this realistic EMS system, when dispatching the closest idle vehicle to each incident, one obtains a fraction late arrivals that is approximately 2.7 times that of the optimal offline policy. We also analyze another online dispatch heuristic, that manages to reduce this gap to approximately 1.9. This constitutes the first quantification of the performance gap between online and offline dispatch policies.

This chapter is based on:
C.J. Jagtenberg, P.L. van den Berg and R.D. van der Mei. Benchmarking online dispatch algorithms for Emergency Medical Services. To appear in *European Journal of Operational Research.*

3.1 Introduction

In the previous chapter, we obtained dispatch policies that attempt to minimize the fraction late arrivals. This showed that we can in fact do better than the classical ‘closest idle’ policy; however, the optimal dispatch policy remains unknown. The question addressed in the present chapter is how to benchmark dispatch policies against optimal policies with full information: suppose we were to know all incident arrivals and locations *in advance*, then how much better would the performance of the optimal dispatch policy be? What is the potential improvement if we were able to perfectly predict future incidents? The answer to such questions addresses the *value of information* about future incidents, and give insight into how far we are from the optimum under full information and what is the potential for developing accurate forecasting models for emergency incidents.

By analyzing the dispatch process from a new angle, this chapter provides a contribution that will be of interest to researchers who develop mathematical models for EMS planning. This new perspective helps to develop a deeper understanding of EMS planning models. Furthermore, we link ambulance dispatching to the literature on online/offline optimization. For a general introduction to the concept of online versus offline algorithms, see [59].

The concept of bounds on the performance of EMS systems is relatively new. There is one recent paper that provides a bound for the performance of an optimal ambulance redeployment policy [79]. However, for a bound on the performance of *dispatch* policies, we are not aware of any result.

There is previous work on ambulance planning that uses ideas similar to offline dispatching, although authors typically do not recognize the idea as such. For example, [118] aims to analyze and evaluate repositioning algorithms, and to that end uses optimal offline dispatch policies as an upper bound on the possible performance. Instead of calling it an offline version of an online problem, the authors refer to the offline approach as ‘the omniscient observer’. Most importantly, this paper differs from our work because it does not include a comparison with online dispatch methods. Other researchers use offline dispatching to compute the number of vehicles needed to serve all incidents, without noting that this is perhaps a rather optimistic approach [39, Chapter 3].

A related problem is the dial-a-ride problem, which deals with online arriving requests for transports between an origin and destination. For an overview of literature on this problem, see [35]. The dial-a-ride problem is similar in the sense that routes are created; however, it typically allows for flexibility in the execution time of each request, whereas the (urgent) ambulance requests require a vehicle to be sent immediately. Furthermore, in dial-a-ride problems the objective is typically either related to efficiency (such as transportation cost or travel time) or based on customers’ inconvenience (such as lateness or excess drive time). There is literature that considers dial-a-ride problems specifically in the ambulance context. However, this usually concerns the non-urgent patient transports, see e.g. [82, 91, 99]. Due to the fact that their objectives are *not* related to a response time threshold, we cannot directly use their results or formulations.

Another related problem is the k-server problem [74], which is one of the classical problems in competitive analysis. In this problem, each time step corresponds to a *request* arriving somewhere in a metric space. There is a set of k servers available, and an algorithm prescribes for each request which server should respond. The objective is to minimize the total distance moved by all servers. The competitive ratio of the k-server problem is currently unknown, although it can be shown that it is at least k, and there exists a conjecture stating that the competitive ratio is exactly k [74]. This problem differs from our ambulance problem in three crucial ways. First of all, in the k-server problem requests do not overlap in time. Second, servers await their next move at the location of their last request, whereas ambulances return to their home base. Third, there is no response time threshold in the k-server problem.

The contribution of this chapter is twofold: (1) to give a bound for the fraction of late arrivals that can be achieved by *any* ambulance dispatch policy, even if all future incident times and locations would be known in advance, and (2) to benchmark and assess the potential for improvement of existing dispatch algorithms. To this end, we introduce three different methods to compute the optimal offline dispatch decisions in case future incident arrivals are known in advance. The first method is Constraint Programming (CP); to the best of our knowledge, this chapter is the first to apply CP to ambulance planning. Next, as an alternative, we also formulate the offline dispatch problem as a Dynamic Programming (DP) problem, and we discuss how this DP provides insight into the problem. We introduce a third method, that is the fastest among the three, using Binary Linear Programming (BLP). We emphasize that all three methods result in the same solution, that is, the optimal solution for the offline problem. Subsequently, we determine the performance of two key online algorithms: the classical 'closest idle ambulance' rule, and the heuristic method described in Chapter 2. These performances are obtained by a discrete event simulation model of an urban EMS region.

Our interest in quantifying the performance gap between online and offline algorithms is twofold. From a theoretical point of view, we are interested in the competitive ratio of the dispatch problem (i.e., a worst case measure for an optimal online algorithm). Conversely, from a practical point of view, we are interested in the performance ratio between online and offline algorithms for *realistic* incident chains. This gives an indication of how much performance improvement can be obtained by developing better dispatch methods, and at the same time shows how much one can benefit from developing accurate incident prediction models.

We do a worst case analysis by constructing a toy example that shows that the so-called *competitive ratio* (i.e., the worst case performance ratio of the fraction of late arrivals between the optimal online and the optimal offline solution) of the dispatch problem is infinitely large; in other words, the optimal online dispatch algorithms can perform arbitrarily bad compared to the offline solution. We also analyze *realistic* problem instances by performing benchmark experiments for a large ambulance provider in the Netherlands. The results show that for this realistic EMS system, the fraction late arrivals of the classical 'closest idle' dispatch heuristic is approximately 3.5%, whereas the offline optimum is 1.5%. What is perhaps most surprising, is that our results show there exists an online dispatch heuristic that closes roughly half of this gap between ‘closest idle’ and the offline optimum. This is the so-called DMEXCLP dispatch heuristic, that results in 2.6% late arrivals (and thereby performs only 1.9 times worse than the optimal offline policy). The remainder of this chapter is structured as follows. In Section 3.2, we give a formal problem definition. In Section 3.3, we describe the two online policies, and introduce - and analyze - three methods to find optimal offline solutions. In Section 3.4, we perform a worst case analysis of the problem, and show that the competitive ratio is infinitely large. We end with computational results for the average case in Section 3.5 and a discussion in Section 3.6.

3.2 Problem formulation

We consider the problem of ambulance dispatching. In this problem, incidents occur randomly in time and space, and the task is to determine which ambulance to send to each incident. Throughout this chapter, we assume the following.

Assumption 1. *The occurrence of incidents is independent of previous incidents and the chosen dispatch policy.*

We consider this assumption to be very realistic. From Assumption 1 follows that we can generate incidents in a preparatory phase, prior to determining the decisions made by each dispatch policy.

3.2.1 Model and notation

We generate incidents in time and space as follows. Define V as the set of locations at which incidents can occur. These demand locations are modeled as a set of discrete points. Incidents at locations in V occur according to a Poisson process with rate λ. Let d_i be the fraction of the demand rate λ that occurs at node i, $i \in V$. Then, on a smaller scale, incidents occur at node i with rate λd_i. According to these Poisson processes, we can simulate sequences of incidents.

Let A be the set of ambulances, and $A_{idle} \subseteq A$ the set of currently idle ambulances. When an incident has occurred, we require an idle ambulance to immediately drive to the scene of the incident. The decision which ambulance to send is the main question of interest in this chapter. Throughout this chapter, we assume the following.

Assumption 2. *There are sufficiently many ambulances, such that at least one ambulance is idle whenever an incident occurs.*

We consider two types of problems: (1) *online* problems, and (2) *offline* problems. In the online problem, the decision which ambulance to send has to be made at the moment the incident occurs; future incidents are unknown and can at best be predicted. In the offline version of the problem, all incidents (i.e., their time stamps and locations) are known *in advance*.

Our objective is formulated in terms of response times, defined as the time between an incident and the arrival of an ambulance at the emergency scene. In practice, incidents have the requirement that an ambulance must be present within T time units. Therefore, we want to *minimize* the *fraction late arrivals*, defined as the fraction of incidents for which the response time is larger than T.

Assumption 3. *We assume that the travel time $\tau_{i,j}$ between two nodes $i, j \in V$ is deterministic, and known in advance.*

Our objective can be formalized as follows. Recall that incidents are generated according to the Poisson process described above. Let C denote a finite set of generated incidents (also known as *calls*), and let n be the number of incidents, i.e., $n = |C|$. Straightforwardly, $t(c)$ denotes the time that incident c occurs ($c \in C$). Let furthermore, $h^{\pi}(c)$ represent the time a vehicle arrives at the scene of incident c, under policy π. Now we can express our objective as:

$$\arg\min_{\pi\in\Pi} \lim_{n\to\infty} \frac{\sum_{c=1}^{n} \mathbb{1}[h^{\pi}(c) - t(c) > T]}{n}. \tag{3.1}$$

Sending an ambulance to an incident is followed by a chain of events, such as spending time on scene with the patient, deciding whether the patient needs transport to a hospital (and if so: additional travel time and a drop-off time at the emergency department). In practice, these events will take a random amount of time. However, this creates a very complex problem, to which both the online *and* offline optimal solution is not known. Thereto, we use a simplified model of the EMS process, which ensures that the optimal offline solution can be computed.

Assumption 4. *The busy time, excluding travel time, is known and deterministic and the same for all calls.*

We define an ambulance to be busy for x minutes after arriving at the scene of an incident. Note that this parameter x is assumed to be independent of the incident location and the base location the ambulance departed from. After these x minutes, the ambulance becomes idle at its (predefined) base location.

We denote the base location of ambulance a by W_a, for $a \in A$. Note that it is possible for multiple ambulances to have the same base location. As soon as an ambulance has reached its base location, it is ready to be dispatched again[1]. An overview of the notation can be found in Table 3.1.

[1] This problem description is similar to the one defined in Chapter 2, with the following two main differences. In Chapter 2 vehicles are allowed to be dispatched while returning to their home base (i.e., when they are on the road) and the ambulance service times are modeled as a stochastic process, rather than a constant time x.

V	The set of demand locations.
A	The set of ambulances.
A_{idle}	The set of idle ambulances.
W_a	The base location for ambulance a, $a \in A$, $W_a \in V$.
T	The time threshold.
x	The time an ambulance is busy with one incident, from the moment of arrival at the scene.
λ	Incident rate.
d_i	The fraction of demand in i, $i \in V$.
$\tau_{i,j}$	The driving time between i and j with siren turned on, $i, j \in V$.
C	A finite chain of incidents.
n	The number of incidents, $\|C\|$.
$t(c)$	The time that incident c occurs, $c \in C$.
$loc(c)$	The location of incident c, $c \in C$, where $loc(c) \in V$.

Table 3.1 Notation.

3.2.2 Goal

In this chapter, we focus on bounding the performance of any online solution to the ambulance dispatch problem. Since the optimal solution to the online problem (in which future incidents are unknown) is not known, we use the optimal solution to the *offline* version of the problem (in which all incidents are known in advance) as a bound.

Our first goal is to formulate a model that allows us to compute the optimal (offline) dispatch policy. Our second goal is to compare this offline optimum to the performance of existing online (heuristic) methods.

3.3 Solution methods

We introduce and implement three different methods to find the optimal offline solution for a general instance of the dispatch problem (with a finite number of calls). The first method, constraint programming (CP), has the advantage that it is easy to implement. The second, dynamic programming (DP), is able to find the same solution with somewhat shorter running times, and on top of that allows us to investigate which properties make an instance hard to solve. The downside of this method is that it is the most time consuming to implement. The third method, Binary Linear Programming (BLP) solves the problem the fastest. In this section we describe the DP and the BLP; the CP model can be found in Appendix 3.A.

We also define the online dispatch policies that we use in our analysis. These solution methods are eventually used to compare the performance on several problem instances.

3.3.1 The optimal offline solution using dynamic programming

In this section, we describe how we built the dynamic program, and what extra features could be added to it in order to speed up the computation. Note that we only need to make decisions right after an incident has occurred. Therefore, we define states at time steps that coincide with the incidents - and just like the incidents, we denote them c from 1 to n. That way, time step c corresponds to the actual time $t(c)$. Additionally, we add a dummy time step $n+1$, with $t(n+1)$ large enough such that all vehicles are idle again, regardless of the dispatch decisions made in the past. The only allowed action at this time step is a dummy action with reward 0.

States, actions and rewards
We define our states to be vectors containing the time in minutes until each ambulance becomes idle. This implies that a state s is a vector of length $|A|$, the number of ambulances in the system. Let $s[a]$ denote the number of minutes until ambulance a becomes idle, for $a \in A$. If this is 0 minutes, that means the vehicle is already idle. At time 0, nothing has happened yet, and all ambulances are idle. Therefore, we start with the zero vector, having a value of 0. In any state s, the allowed actions, i.e., the ambulances that are eligible for dispatch, are given by: $a \in A$ for which $s_c[a] = 0$. At time step c, the penalty corresponding to action a_c ($a_c \in A$) is given by

$$R(s_c, a_c) = \begin{cases} 1 & \text{if } \tau_{W_{a_c}, loc(c)} > T; \\ 0 & \text{otherwise.} \end{cases}$$

Note that the reward for $c = n+1$ is defined as 0.

To know how to update the states, we can precompute the time differences between the incidents. Thereto, we define:

$$\mathit{diff}_c = t(c+1) - t(c) \quad \text{for } c \in C,$$

and define $\mathit{diff}_n = 0$.

Next, we describe how to update any state s_c to state s_{c+1}, where a_c denotes the chosen action at time step c. Let Γ be the transition function, that depends on s_c and a_c. Define $s_{c+1} = \Gamma(s_c, a_c)$ such that

$$s_{c+1}[a] = \begin{cases} \max(\tau_{W_a, loc(c)} + x - \mathit{diff}_c, 0) & \text{if } a = a_c; \\ \max(s_c[a] - \mathit{diff}_c, 0) & \text{otherwise.} \end{cases}$$

The value of being in state $s_{c'}$ at time step c' can then be defined as:

$$V_{c'}(s_{c'}) = \min_{\{a_c\}_{c=0}^{c'}} \sum_{c=0}^{c'} R(s_c, a_c)$$

subject to

$$\mathrm{a}_c \in A \text{ and } s_c[a_c] = 0$$

and

$$s_{c+1} = \Gamma(s_c, a_c), \quad \forall c = 0, 1, 2, \ldots n - 1.$$

The objective is to minimize the fraction of late arrivals, which - for any fixed number of incidents - is equal to minimizing the *number* of late arrivals. So we are interested in the value $V_n(\vec{0})$.

Note that decisions made in the past have a large effect on the set of states that we need to analyze in the future. In fact, only a small subset of all states we can think of will ever be reached. That is, one can obtain s_{c+1} from s_c, but not the other way around. Therefore, a backward recursion does not make sense for this problem; instead, we used a *forward recursion* to obtain the set of states that we need to analyze. Hence, for each state s, we computed the value at time step c based on the value in the previous time step, as follows:

$$V_{c+1}(s_{c+1}) = \min_{a_{c+1}} \{V_c(s_c) + R(s_{c+1}, a_{c+1})\}.$$

Although this method in theory computes the optimal solution to any instance of the offline ambulance dispatch problem, practical difficulties can occur. The difficulty is that many situations need to be considered (in a DP context, that means many states need to be stored).

Note that it is hard to give an exact formula that describes the number of states that need to be computed in order to find the solution. There are, however, two formulas that both give an upper bound on the number states. The first one follows straightforwardly when one realizes that the time until each ambulance becomes available completely defines a state (and that we should consider this n times). This means there are at most $nM^{|A|}$ relevant states, where M is the maximum driving time between any base location and demand point. Furthermore, there is a bound on the number of decisions that can be made. Assuming all possible combinations of ambulance assignments are allowed, this leads to a maximum $|A|^n$ decisions, and hence states, to be considered.

There are some ways to reduce the total number of states required to store, which directly lead to shorter computation times. Appendix 3.B describes three ways to accomplish this.

3.3.2 The optimal offline solution using binary linear programming

To formulate the problem as a BLP, we first introduce parameters $p_{cj} \in \{0, 1\}$, for $c \in C$, $j \in A$. This parameter is the penalty of assigning ambulance j to incident c: it will be set to 0 if ambulance j arrives within threshold time T. Note that the values of p_{cj} can be deduced from the problem specification, using the base locations, driving times between those bases and the incident locations, and the (fixed) parameter T. Our decision variables will be $x_{cj} \in \{0,1\}$, for $c \in C$, $j \in A$, which will be 1 if and only if ambulance j is assigned to incident c.

The most important constraint of our problem is that two incidents handled by the same ambulance may not overlap in time. At first sight, it seems hard to model this in a linear way: recall that the travel time depends on the ambulance that is chosen. However, we can precompute for each combination of incidents c and c', whether or not they overlap in time if they were to be served by ambulance j. Denote this with parameter $o_{cc'j}$, for $c, c' \in C, j \in A$, which is equal to 1 if the incidents overlap in time, and 0 otherwise. If $o_{cc'j}$ equals 1, we add a constraint that at most one incident in $\{c, c'\}$ may be served by ambulance j. Then, the offline ambulance dispatch problem can be modeled as a BLP as follows:

$$\text{Minimize } \sum_{c \in C} \sum_{j \in A} p_{cj} x_{cj}$$

subject to

$$\sum_{j \in A} x_{cj} = 1, \quad c \in C,$$

$$o_{cc'j} \cdot (x_{cj} + x_{c'j}) \leq 1, \quad j \in A, \quad c, c' \in C, c \neq c',$$

$$x_{cj} \in \{0,1\}, \quad c \in C, j \in A.$$

3.3.3 Online solutions

In this section, we describe two online dispatch methods. The first is often used in practice, and the second was shown to give good performance for our objective (the fraction of late arrivals).

The 'closest idle' dispatch method
When an incident occurs, all idle ambulances are considered. The idle ambulance that is closest to the incident location (in time, not necessarily in space), is then dispatched. This notion can be formally expressed as follows:

$$\arg\min_{a \in A_{idle}} (\tau_{W_a, loc(i)}),$$

i.e., the ambulance a for which the travel time τ is the smallest amongst all idle ambulances.

The DMEXCLP dispatch heuristic
This heuristic was introduced in Chapter 2, and applied to test data similar to the data we will use in this chapter[2]. In Chapter 2 we showed that the

[2] In Chapter 2, the region considered is also Utrecht. However, the incident rate as well as the number of vehicles used is slightly lower than in the current chapter.

heuristic reduces the fraction of late arrivals by 18% compared to the 'closest idle' benchmark policy. A mentioned drawback is that this heuristic increases the average response time. Therefore, we do not claim that this heuristic is practically preferable over the closest-idle method. However, the mentioned improvement of 18% is considerable, and hence it would be interesting to see how the heuristic performs compared to the offline optimum.

The general idea of the DMEXCLP dispatch heuristic is that we choose an ambulance such that the remaining idle ambulances provide good *coverage* of the region. Coverage can be interpreted as a number that indicates how well we can serve the incidents that might occur in the (near) future. The definition of coverage for the DMEXCLP dispatch heuristic was borrowed from the well-known MEXCLP model [55], that is described in Section 1.2.1. At the moment an incident occurs, the DMEXCLP dispatch heuristic computes the *marginal* coverage that each ambulance provides for the region, at this point in time. The ambulance that provides the smallest marginal coverage, is the best choice for dispatch, in terms of remaining coverage for future incidents. The heuristic limits the set of ambulances to choose from, by requiring that we always send an ambulance that will reach the incident in time, if possible. This still leaves a certain amount of freedom in determining which *particular* ambulance to send.

3.3.4 Benchmarking solutions

In this section, we describe how we calculated the performance ratio between an online and an offline dispatch policy. By definition, the performance of an online policy must be equal to or worse than the offline optimum. Recall that our objectives are defined as the fraction of late arrivals. Since we are minimizing our objective, we can immediately conclude that the online/offline performance ratio will be ≥ 1.

Given a specific EMS region, we drew a finite sequence of incidents according to the Poisson process defined in Section 3.2.1. Denote the fraction of late arrivals for a certain policy P and incident sequence s by $FracLate_P(s)$. We repeated this process multiple times, using a large set of incident sequences (S), in order to determine the objective more accurately. Our final estimate for the performance ratio is then computed as the ratio of the average performances:

$$\text{Performance Ratio} := \frac{\frac{1}{|S|}\sum_{s\in S} FracLate_{\mathit{Online}}(s)}{\frac{1}{|S|}\sum_{s\in S} FracLate_{\mathit{Offline}}(s)}. \tag{3.2}$$

Note that we do *not* compute the performance ratio of each individual incident sequence. The reason for this, is that when the offline optimum results in 0 late arrivals, the performance ratio becomes infinitely large, and this does not lead to a meaningful average performance ratio.

3.4 Worst case analysis

In this section, we describe a worst case realization of incidents. This example is meant to illustrate to what extent an 'unfortunate' chain of incidents can affect the performance of online dispatch algorithms. The example directly leads to the so-called *competitive ratio* of the dispatch problem.

Consider a region where the time threshold $T = 12$ minutes, and the busy time for an ambulance is $x = 37$ minutes. There are two nodes in which incidents can occur, and the driving time between these nodes is 13 minutes. Each node is the base location for one ambulance. For simplicity, let us say ambulance 1 has base location 1, and ambulance 2 is stationed at location 2. For a graphic representation, see Figure 3.1. It is easy to see that an ambulance will reach an incident in time, if and only if the ambulance at the location of the incident is available.

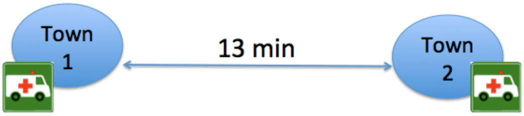


Figure 3.1 Region with two towns, each being the home base for one ambulance.

Incidents			Optimal offline		Closest idle (online)	
Number	Time	Location	Send ambu	In time?	Send ambu	In time?
1	0	1	2	No	1	Yes
2	5	1	1	Yes	2	No
3	51	2	2	Yes	1	No
4	56	1	1	Yes	2	No
5	102	2	2	Yes	1	No
6	107	1	1	Yes	2	No
⋮						
$n = 2m + 1$	$m{\cdot}51$	1	1	yes	2	no

Table 3.2 A worst case example of incidents for the region described in Section 3.4. The corresponding solution of two policies is denoted, as well as whether or not they can serve each incident within the threshold time. Note that the incidents in each location are exactly 51 minutes apart.

Table 3.2 shows a chain of incident realizations for which the closest idle dispatch policy performs particularly poorly. Typical about this example is that a dispatch algorithm only has a choice for the first incident (at time 0). After that, the sequence of incidents is timed such that there is only one ambulance idle at any decision moment[3]. By our problem definition, that ambulance must then be dispatched immediately. So, if an algorithm makes the wrong decision in the

[3] Note that incidents in each location are 51 minutes apart, while the busy time of an ambulance is at most 13+37=50 minutes.

first time step - like the closest idle policy does - all following incidents except the first one the ambulance will arrive later than the threshold time. Alternatively, if the correct decision is made in the first time step, only the first incident's ambulance will arrive late.

Note that it is impossible for any online algorithm to know what is the best decision in the first time step. To see this, imagine an (online) algorithm that upon seeing the first incident at location 1, sends ambulance 2 (hence it does the opposite of the closest idle method.) The worst case instance for this algorithm, would have the same incident times as in Table 3.2, but have the locations of incidents $2 \ldots n$ *swapped* (i.e., location $1 \leftrightarrow 2$). Then, again, only the first incident would be reached in time. Thereto, we conclude that the performance ratio of any online algorithm can be a factor

$$\frac{(n-1)/n}{1/n} = \frac{n-1}{1} \to_{n \to \infty} \infty.$$

larger than the optimal offline policy.

One might argue that the ambulance dispatcher should be allowed to change his mind and send a different ambulance whenever new information becomes available - like a new vehicle becoming idle - as long as the originally dispatched ambulance still has not arrived. That is, the dispatcher might be able to perform better if he is allowed to schedule with preemption during the travel time. However, with a small adaptation to the problem instance in Table 3.2, we obtain a problem instance that again leads to an unbounded competitive ratio, even when preemption is allowed. To see this, change the problem instance in Table 3.2 by increasing the time of incident 2 from 5 to 13, and update the consecutive (odd) incidents accordingly. Then, the new information arrives too late, i.e., the originally dispatched ambulance has already arrived, and hence the competitive ratio remains infinitely large.

Although this worst case is interesting from a theoretical perspective, we want to clarify that this is not a case that is likely to occur in practice. Since ambulance planning is a topic of practical importance, the rest of this chapter focuses on the performance ratio between online and offline algorithms for *realistic* incident chains. More specifically, we are interested in the *expected* performance ratio for incident chains that originate from an incident distribution as described in Section 3.2.1.

3.5 Computational results

In this section, we analyze the ambulance dispatch problem based on an EMS system that represents Utrecht. Figure 3.2 shows a map of the region and the base locations that we used. For more information on the region, we refer to Section 2.7.

We chose realistic parameters to model the EMS region Utrecht. For example, the base locations that we used are equal to the ones used in practice (for at

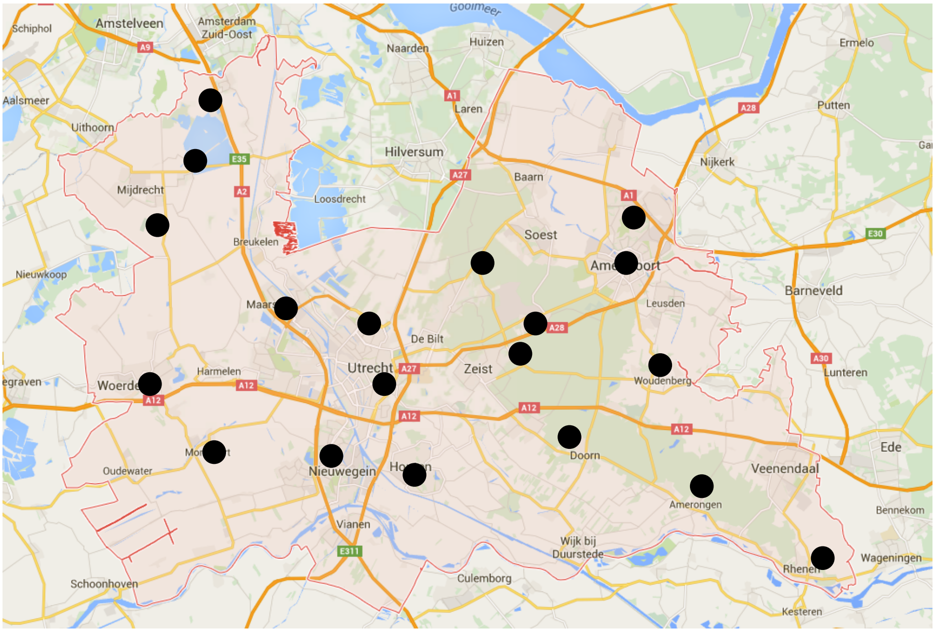


Figure 3.2 The 19 existing ambulance base locations in the region of Utrecht, The Netherlands. We distribute the 25 available ambulances over the bases according to the MEXCLP solution with busy fraction $q = 0.3$.

least the period between 2013 and 2015). Furthermore, we divided the region by postal codes, and model the incident arrivals in each postal code as a Poisson process with a rate proportional to the population. Ambulance travel times were provided by the Dutch National Institute for Public Health and the Environment (RIVM). For the exact parameters used in the implementation, see Table 3.3.

It is clear that we can only analyze *finite* incident chains; however, it is not immediately clear what the length of such chains should be. One might argue that longer chains will lead to a larger performance difference between online and offline solutions - simply because the offline solution is able to look further into the future. On the other hand, it seems reasonable to assume that incidents that are *very* far in the future do not greatly affect current decisions. Thereto, we analyzed incident chains of four different lengths: 6, 12, 18 and 24 hours. One might also argue that the result depends on the value of λ, thereto we analyzed three different values (λ_1, λ_2 and λ_3, as described in Table 3.3).

The parameters described above lead to the analysis of 12 different cases. For each of those cases, we drew $|S|$=1000 incident chains according to the Poisson process described in Section 3.2.1, for the region Utrecht defined in Table 3.3.

In order to compute the optimal offline performance, we implemented all three methods from Section 3.3. First, we tried the CP, which we implemented in the MiniZinc modeling language, using its standard G12 finite domain solver. This

Parameter	**Magnitude**	**Choice**
λ_1	$0.9 \cdot \lambda_2$	A 10% lower rate than normal for this region.
λ_2	$1/6.4$	Rate per minute, realistic for urgent calls on a weekday in this region.
λ_3	$1.1 \cdot \lambda_2$	A 10% higher rate than normal for this region.
A	25	A number chosen such that performance is realistic (near 5% late arrivals).
W	19	Base locations as existing in 2013-2015. We divide the ambulances over the bases according to the static MEXCLP solution.
V	217	4 digit postal codes.
τ_{ij}		Driving times as estimated by the RIVM, rounded to minutes.
d_i		Fraction of inhabitants as known in 2009.
T	12 min	Typical time standard for high priority incidents in the Netherlands.
x	37 min	Realistic average busy time for ambulances.

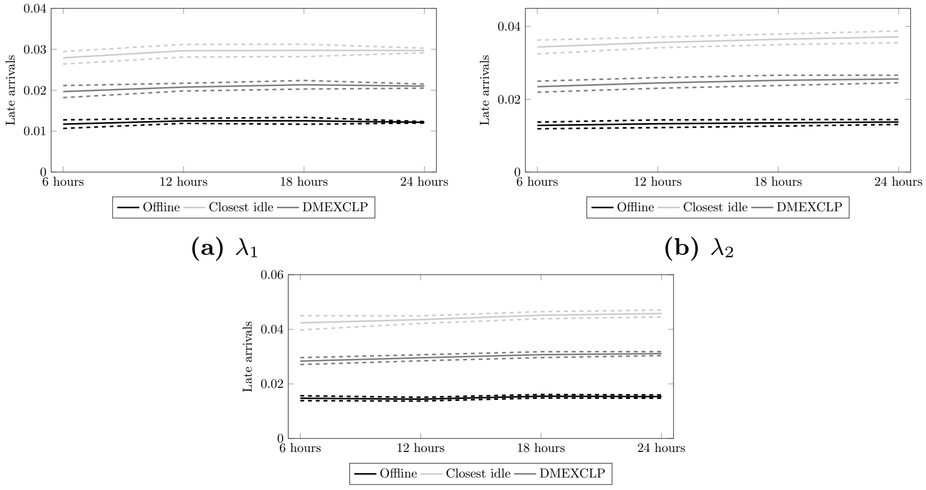
Table 3.3 Parameter choices for our implementation of the region of Utrecht.

could only handle very small problem instances. The largest instances we tried to solve with CP had a simulation time of six hours. The computation time varied widely among the different instances, the longest ones taking more than a day. Next, we implemented the DP in C++, which reduced computation times - again for instances of six hours simulation time - to a range of 20 minutes to a few hours. Finally, we implemented the BLP in Java using solver CPLEX 12.6, which solves all instances, including ones for 24 hours simulation time, in a fraction of a second. As stated in Section 3.3, the performance of the two online dispatch policies is calculated by simulating the EMS system.

Ambulance optimization is a complex topic, and it is often hard to oversee whether stated theoretical results will hold up in practice - even for experts. It is our opinion that in order for results to be meaningful, at least the performance should be close to the performance in practice. In the Netherlands, urgent incidents should be served within the time standard in at least 95% of all cases. The ambulance provider for Utrecht performs slightly better than this 95% on average. Thereto, we decided to use a number of vehicles such that the average fraction of late arrivals for the online dispatch methods is roughly between 3 and 5%. We believe that this choice leads to the most realistic and insightful results.

In practice as well as in our experiments, it is rather unlikely that EMS region Utrecht faces a situation in which all vehicles are busy. This means that Assumption 2 is quite realistic. We validated this assumption for all incident chains in our numerical work.

The obtained fraction of late arrivals for each of the twelve cases is depicted in Figure 3.3. Recall that we compute the performance ratio as described in Equation 3.2. The results from Figure 3.3 then lead to the performance ratios



(a) λ_1

(b) λ_2

(c) λ_3

Figure 3.3 Average fraction late arrivals for 1000 chains of incidents, with different incident intensities and chain lengths. A 95% confidence interval is displayed.

	λ_1	λ_2	λ_3
DMEXCLP			
6 hours	1.72 ± 0.07	1.86 ± 0.07	1.96 ± 0.12
12 hours	1.67 ± 0.05	1.87 ± 0.07	2.06 ± 0.06
18 hours	1.72 ± 0.07	1.88 ± 0.07	2.00 ± 0.05
24 hours	1.74 ± 0.05	1.87 ± 0.05	2.04 ± 0.06
Closest idle			
6 hours	2.48 ± 0.19	2.73 ± 0.11	2.91 ± 0.17
12 hours	2.39 ± 0.09	2.72 ± 0.14	3.05 ± 0.09
18 hours	2.40 ± 0.11	2.73 ± 0.11	2.94 ± 0.09
24 hours	2.46 ± 0.06	2.72 ± 0.10	3.00 ± 0.09

Table 3.4 The observed Performance Ratio and 95% confidence interval of online dispatch policies.

found in Table 3.4.

A bound on optimal online algorithms

Our offline optimum constitutes the first known bound on the performance of an optimal online ambulance dispatch policy. Let us focus on incident arrival rate λ_2, since it is realistic for this particular EMS region. Then Table 3.4 shows that the DMEXCLP dispatch policy performs approximately 1.9 times worse than the offline optimum.[4] This means that there cannot exist an online dispatch method that improves the performance of the DMEXCLP dispatch method by more than a factor 1.9 on average. We emphasize that this bound is an optimistic one, since it is obtained using information - on future incidents - that is inaccessible to online policies, but it is a bound nonetheless.

The value of information
Generally speaking, the competitive ratio of a problem shows the importance of knowing the future for this problem. In terms of the ambulance dispatch problem, it gives an indication of 'unfortunate decisions' made by online policies - even an optimal one - that could not have been avoided unless one knew about future incidents. Our results are perhaps surprising: we had previously expected that knowing incidents in advance would have a greater impact on performance. However, our results show that even an omniscient dispatcher will still be left with $\frac{1}{1.9} \approx 53\%$ of the late arrivals, compared to a dispatcher that executes DMEXCLP.

3.6 Discussion

We have introduced three methods to compute the offline optimal solution to the ambulance dispatch problem. Note that, due to scalability issues, the CP and DP method are not advisable for most numerical work; we recommend the BLP to solve the problem practically.

One may perform the analysis as described in this chapter for multiple EMS regions. Different regions typically have different characteristics, such as the average busy fraction of ambulances, or the distance between bases and demand. These differences will most likely result in a different online/offline performance ratio, and it would be interesting to see how these ratios vary over different regions. However, regions are always hard to compare, and therefore instead of simulating different regions we chose to analyze the effect of different arrival intensities.

Table 3.4 shows that the Performance Ratio between the online policies and the offline optimum increases with λ. This may be explained as follows. A larger λ leads to more incidents within a short time frame. As we have seen in Section 3.3.1, this makes for a more complex problem, because many decisions are now dependent on one another. In particular, an unfortunate choice at some point can have an effect on many incidents after that. It is therefore not surprising that the performance gap between the online heuristics and the offline optimum increases with λ.

[4] Furthermore, Table 3.4 indicates that the Performance Ratio does not vary greatly between cases of 12, 18 and 24 hours. Therefore we conjecture that the 24 hour case gives a reasonable estimate of the true Performance Ratio.

Although it was previously known that the closest idle method is not optimal, it is often assumed to be quite a good policy. In fact, the insight that the ‘closest idle’ performance is still a factor 2.7 away from the offline optimum, is something that many researchers in the field of ambulance planning may be tempted to attribute to the *value of information*: to the fact that the offline policy has much more knowledge. However, as Figure 3.3 depicts, the DMEXCLP dispatch heuristic is able to close about half the gap between the ‘closest idle’ and the offline optimum. We find this observation rather surprising, as it implies that the value of information for the dispatch problem is smaller than we had previously anticipated.

In order to compute the Performance Ratio, we drew random chains of incidents. However, we always started at time 0 with all ambulances idle. This may perhaps be interpreted as the start of the day, for EMS providers that serve few calls at night. However, one might also argue that we should focus more on the system in *steady state.* We conjecture that our result - a Performance Ratio of 1.9 - will roughly hold for steady state as well, since the value did not change much between incident chains of 12, 18 and 24 hours (see Table 3.4).

Finally, one might suggest to make the model more realistic, e.g., by defining the busy time of an ambulance after arrival at an incident to be a random variable. Then, however, determining the optimal offline policy becomes a very difficult task. We see only one way to overcome this difficulty, and that is to let the offline solution have knowledge of the *realizations* of these random times. Since online policies can only use the busy times in *distribution*, this would increase the gap between information given to the offline and online policies. This deviates further from our main research question, which was how much it helps to have information on when and where incidents will occur. Increasing the gap between what is known in the online and offline case will not help us to gain more insight in this matter. Therefore, we decided not to proceed in this direction.

APPENDICES

3.A Constraint Programming formulation

In this appendix, we describe how we found the optimal offline solution with CP. For this purpose, we used the MiniZinc constraint modeling language. We modelled a set of n incidents by the following variables:

- $t(c)$, the cth element of vector $\vec{t}$. This is the time that incident c occurs, $c \in \{1, \dots, n\}$.
- $loc(c)$, the cth element of vector $\vec{loc}$. This is the location of each incident, where $loc(c) \in V$, $c \in \{1, \dots, n\}$.

The input further consists of the base location W_a of each ambulance a, as well as the driving times $\tau_{i,j} \quad \forall i, j \in V$ (in minutes). For each incident c we introduced a variable $\mathscr{A}(c)$, which can take a value between 1 and $|A|$. These variables indicate which ambulance is assigned to each incident.

We aimed to minimize the fraction of arrivals later than threshold time T. Note that since the number of incidents - and therefore the number of arrivals - is known in advance, this is equivalent to minimizing the *number* of late arrivals. In our implementation, we focused on the number of late arrivals, denoted by $\mathscr{N}$.

Finally, we needed to ensure feasibility of the solution. Thereto, we added two constraints[5]. Equation (3.3) makes sure variable $\mathscr{N}$ is set correctly, i.e., it is the number of incidents for which the dispatched ambulance was further than T minutes away. Equation (3.4) ensures that two incidents (c_1 and c_2) assigned to the same ambulance do not overlap in time.

Note that this is not the only CP model one could formulate. In fact, a model similar to the BLP model that we have seen in Section 3.3.2 is also possible for CP. However, we chose to keep the models diverse.

$$\text{minimize } \mathscr{N}$$

s.t.

$$\mathscr{N} = \sum_{c \in C} \mathbb{1}(\tau_{W_{\mathscr{A}(c)}, loc(c)} > T) \tag{3.3}$$

and

$$\begin{aligned} &\nexists\, c_1, c_2 \in C \quad \text{such that} \\ &\quad c_1 < c_2 \quad \wedge \quad \mathscr{A}(c_1) = \mathscr{A}(c_2) \quad \wedge \quad t(c_1) + \tau_{W_{\mathscr{A}(c_1)}, loc(c_1)} + x > t(c_2). \end{aligned} \tag{3.4}$$

[5] We also added other - redundant - constraints in order to find solutions faster. However, they do not change the result and therefore we do not mention them here.

Note that in the formulation of Equation (3.4), we used the assumption that incidents are ordered chronologically.

One can immediately see the benefit of the flexibility that CP has to offer: we were able to write the problem using just two constraints, which look very similar to the way one might naturally think about the dispatch problem.

3.B DP speed-up

In this appendix, we describe three ways to speed up the computation time of the DP. We illustrate the usefulness of each technique, by the effect it has on the following two problem instances[6]. In both examples, $T = 12$.

Instance 1.

$$\vec{t} = [9, 13, 35, 47, 70, 95, 104, 105, 115, 127, 152, 169].$$

$$\vec{loc} = [34, 54, 23, 159, 81, 81, 39, 10, 142, 146, 140, 156].$$

For this instance, the closest idle dispatch policy results in one late arrival. The offline optimum is also one late arrival.

Instance 2.

$$\vec{t} = [1, 1, 30, 33, 34, 43, 43, 62, 63, 81, 103, 114, 124, 135, 138, 139, 168, 174].$$

$$\vec{loc} = [200, 182, 135, 217, 67, 131, 74, 179, 95, 15, 74, 37, 206, 206, 142, 54, 145, 44].$$

For this instance, the closest idle dispatch policy results in four late arrivals. The offline optimum is equal to three late arrivals.

Eliminating dominated states

One well-known way to reduce the number of states, is to eliminate so-called dominated states. We define a state s to be dominated at time c, if there exists another state s', such that:

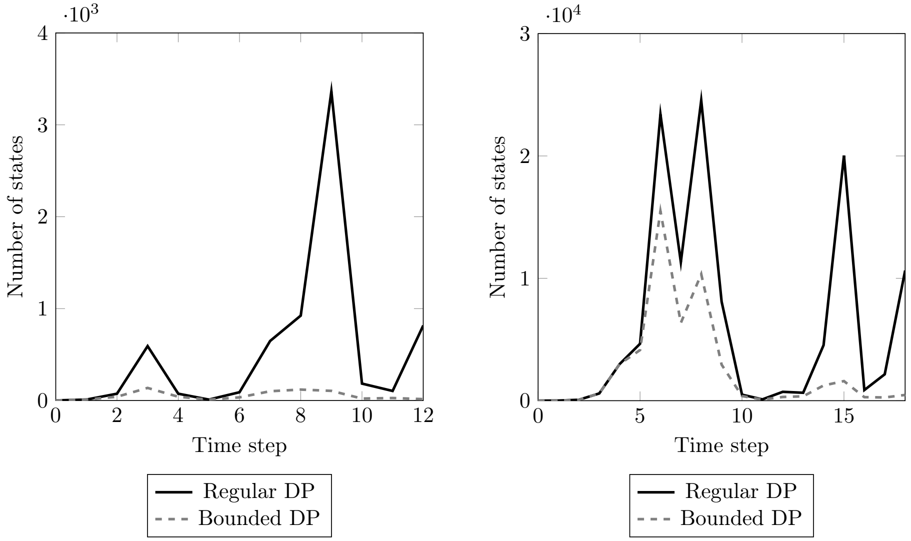
$$s'[a] \leq s[a] \quad \forall a \in A \quad \text{and} \quad V_c(s') \leq V_c(s).$$

That is, there exists another state for which all vehicles will be idle at earlier (or equal) times, while resulting in fewer (or equal) late arrivals. We iteratively removed dominated states until none are left in our state space.

Bounding the objective

Another way to reduce the time and memory spent on the dynamic program, is to bound the solution by any feasible objective value. For example, we can quickly pre-compute the objective from the 'closest idle' dispatch heuristic and eliminate any state that has a larger value. We show the benefit of this approach by example: Figure 3.4 depicts the number of states that we need to analyze at each time step. Note that Figure 3.4b has more time steps than Figure 3.4a, simply because more incidents occur.

[6] The cases considered here are for the region Utrecht. Due to the scalability issues of the DP, we used only 10 ambulances in this example.



(a) Instance 1

(b) Instance 2

Figure 3.4 Comparison of the number of states stored for each time step in the dynamic program, with and without bounding the solution by the value of the 'closest idle' policy.

Figure 3.4 shows that in the first few time steps the number of states for the bounded and unbounded DP are more or less equal. Let us explain why this makes sense, by the example of Instance 2. Here, the closest idle method results in four late arrivals. Therefore, bounding the states by the ones with values ≤ 4 does not have any effect before time $c = 5$ (since the value can increase by at most 1 per incident).[7]

Also note that the number of states does not always increase over time. So what is it exactly, that causes the need to store many states? A key insight is that an incident that occurs at time t, only has an *indirect* effect on the system[8] after time $t + \tau_{i,j} + 37$, for some travel time $\tau_{i,j}, i, j \in V$. That is, the ambulance will be idle by time $t + \tau_{i,j} + 37$, and can be used for any incident after that time, regardless of whether it is dispatched to the incident at time t. However, whether or not this particular ambulance is dispatched at time t, *does* have an effect on which ambulances are eligible for dispatch to incidents between time t and $t + \tau_{i,j} + 37$. Hence, we regard the effect as an indirect one. Note that this indirect effect occurs *only* when incidents arise within this time frame. This leads to the following observation.

[7] An alternative, more elaborate way to bound the DP would be to store all states that the (online) heuristic passes through over time. That is, after each incident, store the remaining busy time for each ambulance, as well as the number of late arrivals observed in the past. Then, when computing the offline optimum in the DP, remove all states that are dominated by the states observed in the online solution. Note that we did not implement this idea.

[8] assuming that we never run out of ambulances

Observation 1. *Longer inter-arrival times lead to a reduction in the number of states.*

In particular, if the inter-arrival time between two consecutive incidents is larger than 37 minutes plus the response time from any base to the first incident (of the two), then the state space reduces to a single state (all ambulances are idle). This can be viewed as a 'reset' of the system, i.e., all information on past decisions are irrelevant for future decisions. When this occurs, it avoids an explosion of the state space that we would otherwise see for instances with a long horizon.

Roughly speaking, the computation time should scale linearly with the simulation horizon (given a fixed λ). However, what makes an instance harder to solve is the number of incidents that occur quickly after one another. Many incidents in a short time window imply many *dependent* decisions - and precisely this increases the number of states. This effect can be seen in the 'spikes' in Figures 3.4a and 3.4b (one can manually confirm this using the incidents times given in 2 and 1. Conversely, the time steps with a very small number of states correspond to large inter-arrival times between incidents.

Rounding the numbers in the input

Another way to speed up the computations is by rounding the driving times and incident times. For example, instead of rounding to minutes, we could round the times to multiples of five minutes. The reason why this would result in fewer states, is that more states will be dominated.

Unfortunately, rounding means some accuracy will be lost: we make use of a trade-off between running time and accuracy here. At the very least, one should make sure to also round the times in the input for the heuristic solutions, or else the computed ratio is meaningless. Then, one might argue that the computed ratio will be similar to the unrounded case. Note that we did not implement this method, but instead suggest to use a Binary Linear Programming approach as described in Section 3.3.2.

4
An efficient heuristic for real-time ambulance redeployment

This chapter addresses the problem of dynamic ambulance repositioning, in which the goal is to minimize the expected fraction of late arrivals. The decisions on how to redeploy the vehicles have to be made in real time, and may take into account the status of all other vehicles and incidents. This is generally considered a complex problem, especially in urban areas, and exact solution methods quickly become intractable when the number of vehicles grows. Therefore, there is a need for a scalable algorithm that performs well in practice.

We propose a polynomial-time heuristic that distinguishes itself by being scalable, easy to program and easy to deploy, while giving good performance for busy regions. The performance of our repositioning method is evaluated in a simulation model of EMS operations, and compared to static solutions. The results show that the heuristic performs better than the optimal static solution for a tractable problem instance. Moreover, we perform a realistic urban case study in which we show that the performance of our heuristic is a 16.8% relative improvement on a benchmark static solution. The studied problem instances show that our algorithm fulfils the need for real-time, simple redeployment policies that significantly outperform static policies.

This chapter is based on:
C. J. Jagtenberg, S. Bhulai and R. D. van der Mei. An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care* 4:27–35, 2015.

4.1 Introduction

This chapter considers the problem of dynamic ambulance repositioning, also known as *redeployment* or *move-up*: proactively relocating idle vehicles in order to reduce response times. The general idea is that the idle vehicles should be relocated to compensate for other ambulances that are busy and hence temporarily unavailable to respond to incidents. Decisions on how to redeploy the vehicles are to be made in real time, and may take into account the status of all other vehicles and incidents.

A variety of techniques has been used to tackle this problem, a summary of which can be found in Section 1.2.2 and in [11]. The randomness in the EMS system combined with a large state space make this problem difficult: while exact models can be solved for small problem instances, realistically-sized EMS regions require an approximation. Such approximations typically include simplifying assumptions. For example, some of the models in literature assume that an incident is served late if there are no idle vehicles present at the nearest base (e.g., [78]). This particular assumption would make a model unsuitable for the EMS region that we have in mind: it includes demand points that can be reached within the time threshold from as many as eight different bases. Despite using simplifying assumptions, some of the existing approximations are in fact not all that simple: they are still computationally heavy and require an expert to implement them.

We conclude that there is a need for a clear, scalable algorithm that performs well in practice. Motivated by this, the current chapter proposes a method that is easy to implement and allows computations to be done in real time, even for large problem instances. We believe that this properly balances the trade-off between simplicity, effectiveness and scalability. Furthermore, our method only uses limited information about the system, which allows even EMS providers with few tools available to track real-time information to implement this solution.

Throughout this chapter the key performance indicator (KPI) is the expected fraction of late arrivals. We validate our method through simulation: our results show that we can obtain an average of 7.8% late arrivals, compared to 9.5% for a benchmark static policy under the same circumstances. In fact, our simulations show that our policy not only performs better for the time threshold, but shifts the entire distribution of response times to the left. These results demonstrate that our algorithm has the potential to be used in real systems, which eventually lead to the implementation in practice in Flevoland, the Netherlands.

The rest of this chapter is structured as follows. In Section 4.2 we formulate the problem. In Section 4.3 we give our ambulance redeployment algorithm and analyze its computation time. In Section 4.4 we describe our case studies and measure the performance of our algorithm on these cases. We do a small case study, allowing us to compute the *optimal* static policy as a benchmark. We also include a realistic case study on one of the largest EMS regions in the Netherlands. Section 4.5 contains a discussion of our approach. We finish by briefly covering the implementation of our method in practice in Section 4.6.

4.2 Problem formulation

In this section we introduce the real-time ambulance redeployment problem. To formulate the problem, define the set V as the set of locations at which demand for ambulances can occur. Note that the demand locations are modeled as a set of discrete points. Incidents at locations in V occur according to a Poisson process with a rate λ. Let d_i be the fraction of the demand rate λ that occurs at

A	The set of ambulances.
V	The set of demand locations.
H	The set of hospital locations, $H \subseteq V$.
W	The set of base locations, $W \subseteq V$.
T	The time threshold.
λ	Incident rate.
d_i	The fraction of demand in i, $i \in V$.
τ_{ij}	The driving time between i and j with siren turned on, $i, j \in V$.
n_i	The number of idle ambulances that have destination i, $i \in W$.

Table 4.1 Notation.

node i, $i \in V$. Then, on a smaller scale, incidents occur at node i with rate $d_i\lambda$.

Let A be the set of ambulances. When an incident has occurred, we require the nearest (in time) available ambulance to immediately drive to the scene of the incident. We assume that the travel times τ_{ij} between two nodes $i, j \in V$ are deterministic.[1] Idle ambulances can only be on the road while driving to a base location in the set $W \subseteq V$, or be at a base location itself waiting for an incident to respond to. Note that idle ambulances on the road may be dispatched immediately, and need not arrive at the base location they were headed to. When an incidents occurs and there are no ambulances idle, the call goes into a first-come first-serve queue. Incidents have the requirement that an ambulance must be present within T time units. When an ambulance arrives at the incident scene, it provides service for a certain random time $\tau_{onscene}$. Then it is decided whether the patient needs transport to a hospital. If not, the ambulance immediately becomes idle. Otherwise, the ambulance drives to the nearest hospital in a set $H \subseteq V$. Upon arrival, the patient is transferred to the emergency department, taking a random time $\tau_{hospital}$, after which the ambulance becomes idle. For an overview of notation, see Table 4.1.

We allow an ambulance only to relocate whenever it becomes idle, which could be at the incident scene or at a hospital. Although this choice may seem restrictive, it is a reasonable choice in practice, and is both crew and fuel friendly. In particular, in complicated busy regions, an ambulance becomes idle quite often. Our restriction on relocation moments provides the system enough freedom to keep updating and avoids getting stuck in a local optimum. In our model, any ambulance is capable of serving any incident. An ambulance is able to respond to an incident (queued or newly arriving), immediately when it becomes idle. Note that this implies that the vehicle does not need to return to a base location before being dispatched again.

[1] Our model uses two different travel speeds. If the ambulance is traveling without siren, its travel speed is 0.9 times the travel speed when it is traveling towards an incident scene.

4.2.1 State space and policy definition

When defining the state space, one should consider all information of the EMS system that the best relocation might depend on. In a way, the state should represent a ‘snap shot’ of the system at a decision moment. Most dynamic models (see Section 4.1) use a rather elaborate description of the system, which results in a large state space. In contrast, we will define a relatively small state space, which will help us obtain an intuitive policy that can be understood and explained to EMS employees in practice.

A state describes the *destinations* of all *idle* ambulances. (If an ambulance is waiting to be dispatched, we say its destination is simply its current location.) It should be clear that this definition of the state space ignores many details of the system, such as information about the busy vehicles and the exact location of ambulances that are driving. Note that ignoring this information (which might affect the best relocation decision) implies that we cannot possibly hope for our method to find an optimal solution. Nevertheless, we show that we can obtain a policy with good performance using only this small state space.

Remember that idle ambulances can only be sent to the predefined base locations in W. Furthermore, the vehicles are exchangeable or identical. It is then sufficient to model the state as the *number* of idle ambulances that are headed to each base location. Hence, define the state space $\mathcal{S}$ to be the set of states $s = \{n_1, \dots, n_{|W|}\}$ such that $n_i \in \mathbb{N}$ for $i = 1, \dots, |W|$ and $\sum_{i=1}^{|W|} n_i \leq |A|$, where n_i represents the number of idle ambulances that have destination i. We also define the action space $\mathcal{A} = W$, where the action represents the new destination for the newly available ambulance. Now we can define a *policy* π, as a mapping $\mathcal{S} \to \mathcal{A}$. Let Π denote the set of all such policies.

4.2.2 Objective

We look for a relocation policy that minimizes the expected fraction of incidents that are reached later than T. Recall that incidents are generated according to the Poisson process described above. Therefore, we can give our incidents an index $i = 1, 2, \dots, I$, sorted by their arrival time. Now we can express our objective as:

$$\arg\min_{\pi \in \Pi} \lim_{I \to \infty} \frac{\sum_{i=1}^{I} \mathbb{1}[h^\pi(i) - t(i) > T]}{I}, \tag{4.1}$$

where $t(i)$ represents the time that incident i occurs, and $h^\pi(i)$ represents the time a vehicle arrives at the scene of incident i, under policy π.

4.3 Algorithm

In this section, we develop an algorithm to solve the dynamic ambulance relocation problem. In some sense, this problem can be considered the counterpart of the dispatching problem in Chapter 2: instead of deciding from which location to remove (dispatch) a vehicle, we now decide which location to add an ambulance to. Therefore, our solution will also show similarities to the dispatch heuristic presented in Chapter 2.

Our goal is to minimize the expected fraction of late arrivals. In order to reach this goal, we will use the notion of *coverage*. It is intuitive that a well-covered region will result in a small expected fraction of late arrivals. Coverage is often used in models for the ambulance location problem, i.e., problems where one searches for a *static* solution. We notice that we can benefit from these models by adapting them in such a way, that they can be used in a dynamic context.

Our tactic is to use as little information as possible, such that it can be applied in general settings, and such that it is implicitly insensitive to changes or estimation errors of the parameters. Hence, we search for a redeployment policy π, using the state space as described in Section 4.2. This means that whenever an ambulance becomes idle, we can only use the destinations of all other idle ambulances to base our decision on. This corresponds to taking a decision in the state in which all idle ambulances have arrived at their destination. Note, however, that this situation may not even occur, because incidents may occur or other vehicles may become idle in the mean time. However, it will turn out to be a useful state description nonetheless.

Recall that we are looking for a policy that minimizes the expected fraction of late arrivals over a set of random incidents (see Equation (4.1)). At any decision moment, the idle ambulances at that epoch already provide a certain coverage of the region. We then decide where to send the vehicle that is about to become idle, by calculating the coverage improvement when it is sent to base w, for all $w \in W$. Note that there are several definitions of 'coverage', which all lead to different redeployment strategies. We find it instructive to first address the most basic notion of coverage. This results in a myopic redeployment policy. We discuss its behavior and shortcomings, which builds up to our proposed solution that uses the same definition of coverage as the MEXCLP model.

Myopic solution

At decision moments, we can straightforwardly calculate which regions are not covered at all. That is, the demand nodes that are further than T away from any idle ambulance destination. We can then make a greedy choice by sending the newly idle ambulance to a base that covers most of the yet uncovered demand. Note that this is a myopic solution, it is in fact a dynamic version of the Maximum Coverage Location Problem (MCLP) [34]. We have implemented this policy, and found that its performance hardly improved the static MEXCLP solution (as elaborated in Section 1.2.1). For some choices for the parameters of the system, the performance was even worse than the static solution. The intuition behind this poor performance is that this MCLP-based policy steers towards a configuration that is optimal with respect to covering the *next* emergency call. This might be sufficient for problem instances with a very low incident arrival rate, but in busy regions such behavior is too shortsighted. In other words: it lacks the insight of how much coverage is left after responding to the first call. This is typical for myopic policies, and in order to overcome this, we require some quantification of where there will be a shortage of ambulances in the future.

Dynamic MEXCLP solution
To obtain a good policy for busy regions, we need to include some measure of how much coverage we can provide in the future. In other words, we need to take into account that some of the currently idle vehicles may be dispatched, and ensure the remaining coverage in the future is still good. Therefore, we propose a policy that sends the idle ambulance to the base that results in the largest marginal coverage according to the MEXCLP model (see Section 1.2.1).

Recall that the MEXCLP model defines the expected covered demand of a node i to be $E_k = d_i(1-q^k)$, where parameter q is the busy fraction, and k are the number of vehicles within reach of demand node i. The corresponding *marginal* coverage, i.e., the benefit of adding a k^{th} ambulance within reach of demand node i, is then given by $E_k - E_{k-1} = d_i(1-q)q^{k-1}$. We next apply this notion of coverage - that was originally defined to find good *static* solutions - in a dynamic context.

We send the ambulance that recently became idle to the base that gives the largest marginal coverage over all demand, which implies that also the largest coverage overall is obtained. This can be expressed as follows:

$$\pi(\{n_1,\dots,n_{|W|}\}) = \arg\max_{w\in W} \sum_{i\in V} d_i(1-q)q^{k(i,w,n_1,\dots,n_{|W|})-1}\cdot \mathbb{1}(\tau_{wi}\le T),$$

$$\text{where } k(i,w,n_1,\dots,n_{|W|}) = \sum_{j=1}^{|W|} n_j \cdot \mathbb{1}(\tau_{ji}\le T) + \mathbb{1}(\tau_{wi}\le T).$$

The travel times τ_{ji} are taken as estimates for movements with siren turned on. We perform the search for the best relocation brute force, as described in Algorithm 1.

4.3.1 Limitations

As described in Section 4.2.1, our state space definition prohibits the ambulance relocation problem from being solved to optimality. But even within our state space, the Dynamic MEXLP model need not lead to optimal decisions. The definition of (marginal) coverage as given by the MEXCLP model has some well-known imperfections. For example, vehicles are assumed to operate independently, and the busy fraction is assumed to be the same for all vehicles. These limitations also transfer to the dynamic usage of (MEXCLP) coverage. Therefore, our proposed solution must be a heuristic one, and we do not claim to have solved the problem in an exact manner. However, heuristic policies are common

```
Data: The demand $d_i$ per node $i \in V$,
base locations $W \subseteq V$,
busy fraction $q \in [0, 1]$,
current destinations $dest(a)$ for all $a \in IdleAmbulances \subseteq A$
travel times $\tau_{ij}$ between any $i, j \in V$,
time threshold $T$ to reach an emergency call.
Result: A new destination for the ambulance that is about to become
        idle
BestImprovement = 0
BestLocation = NULL
foreach $j$ in $W$ do
    CoverageImprovement = 0
    foreach $i$ in $V$ do
        $k = 0$
        if $\tau_{ji} \leq T$ then
            $k$++
            foreach $a$ in IdleAmbulances do
                if $\tau_{dest(a)i} \leq T$ then
                    $k$++
                end
            end
            CoverageImprovement $+= d_i(1 - q)q^{k-1}$
        end
    end
    if CoverageImprovement > BestImprovement then
        BestLocation = $j$
        BestImprovement = CoverageImprovement
    end
end
return BestLocation
```

Algorithm 1: Dynamic MEXCLP

in dynamic ambulance planning, due to the difficulty of the problem. Furthermore, we consider the MEXCLP definition of coverage an elegant one, and it allows for fast computations (as we will see in Section 4.3.2).

4.3.2 Computation time

We analyse the computation time of dynamic MEXCLP, in order to determine the scalability of our method. In Algorithm 1 it is easy to see that we loop over all bases, demand nodes and idle ambulances. Therefore, the dynamic MEXCLP algorithm runs in $\mathcal{O}(|W||V||A|)$ iterations.

In practice the number of base locations is typically small, e.g., 20 or 30. Also the number of ambulances that an EMS provider uses, is limited (e.g., [2, 77, 89]). The size of V is mostly dependent on the way the data is aggregated, and it is the only quantity that is likely to be large. The fact that the computation time is linear in $|V|$, ensures that Algorithm 1 will remain tractable even for large regions or regions with a high level of detail.

4.4 Computational results

In this section we verify our dynamic MEXCLP repositioning policy by simulating several EMS regions. To this end, we built a discrete event simulation model that keeps track of all incidents and vehicles. There are events for an incident occurring, an ambulance arriving at the scene of the incident, an ambulance leaving for a hospital, an ambulance arriving at a hospital, and an ambulance becoming idle.

We draw incident arrival times and locations according to a spatial Poisson process as described in Section 4.2. When an incident occurs, the closest idle ambulance is dispatched. For every vehicle we keep track of the origin and destination, including the start time of its movement. This allows us to determine where moving ambulances are while we look for the closest available vehicle. We do this by a linear interpolation between the origin and destination, given the time since the ambulance started moving and the known total driving time from origin to destination. We then round our result down to the nearest point in V, since our estimates for driving times are only given between points in V. Our experiments show that for the majority of the incidents, approximately 77%, the corresponding ambulance departs from a base location.

In our simulation, $\tau_{onscene}$ is exponentially distributed with an expectation of 12 minutes. $\tau_{hospital}$ is drawn from a Weibull distribution with an expectation of 15 minutes. More specifically, it has shape parameter 1.5 and scale parameter 18 (in minutes). We state these distributions for completeness, however, numerical experiments (done by the authors in ongoing work) indicate that the performance does not depend much on the chosen distribution for $\tau_{onscene}$ or $\tau_{hospital}$. In our simulations, patients need hospital treatment with probability 0.8. This value

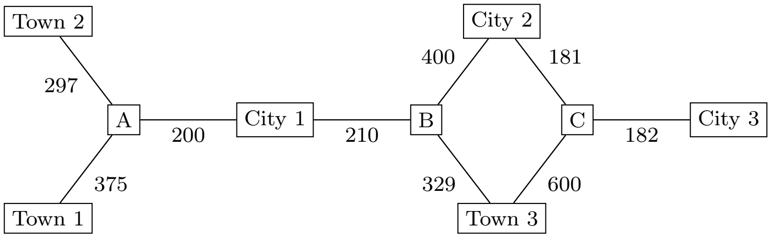


Figure 4.1 A graph representation of the region. The numbers on the edges represent the driving times in seconds with siren turned on.

was estimated from Dutch data [112]. Similar numbers (78% nation-wide) can be deduced from [89].

When an ambulance completes an incident, we check if there are any unattended incidents left in the queue. If not, the ambulance becomes idle, and is sent to a base location.[2] In our proposed solution, this base location is determined by Algorithm 1. As benchmarks, we use static solutions, in which the idle ambulance returns to its own pre-defined home base. This is a typical benchmark in ambulance redeployment literature (used, e.g., in [77] and [118]). Recall that we measure the fraction of ambulances arriving at the scene of an incident with a response time larger than T.

4.4.1 A small region

We first introduce a tractable region, which consists of a small number of demand nodes and vehicles. This is insightful as it allows for a brute force search among all static policies, and thereby allows us to use the *optimal* static policy as a benchmark. Note that this is not possible for a large region: although there exist many models for the ambulance location problem in literature, their solution can only ever be considered optimal with respect to the selected model. None of these models are able to fully capture the complex dynamics of the EMS process: from the way ambulance unavailability is modelled to the fact that ambulances are allowed to be dispatched while on the road.

The region we use is inspired by a small part of the Netherlands. We aggregate the demand at the level of municipalities, which in this case boils down to cities and towns. Furthermore, we add three nodes, A, B and C, that are located at important road intersections. These last nodes have no demand, but it is possible to strategically station an ambulance there. For the geographical characteristics of the region, see Figure 4.1. In this region there is only one hospital, which is located in City 2.

For illustration, we set the time threshold to $T = 10$ minutes, and use demand as described in Table 4.2. Furthermore, we allow exactly five ambulances to

[2] Recall that the ambulance might not arrive at this base location, because it may be dispatched before reaching its destination.

i	d_i
City 1	0.2
City 2	0.4
City 3	0.2
Town 1	0.07
Town 2	0.07
Town 3	0.06
A	0
B	0
C	0

Table 4.2 Distribution of demand in a small region.

serve the incidents in this region.

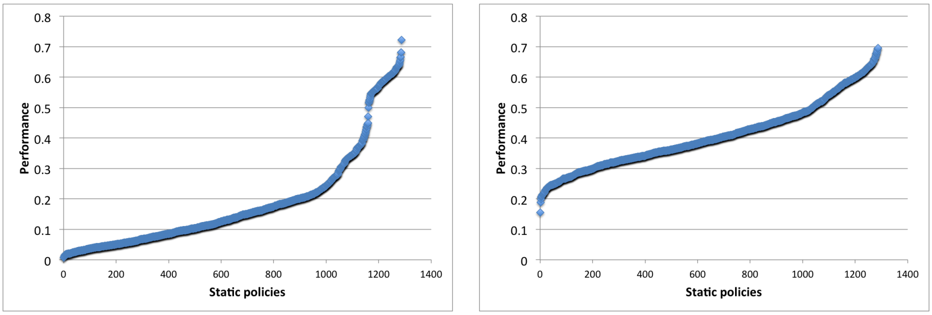
Static policies
Let us consider static policies first. We have nine nodes and give vehicles available. If vehicles were distinguishable, this would mean there are $9^5 = 59,049$ different static policies. Instead, we assume vehicles are indistinguishable, which makes the set of truly different policies smaller. If we number the nodes 1 up until 9, we can describe a policy by a five tuple of non-decreasing integers, representing the home locations of the five vehicles. For example., (2,2,5,8,9) denotes a policy, but (5,6,3,1,9) does not. Using this definition, we can iterate over all static policies. This allows us to take a closer look at the static solution space. Finding the optimal solution for a discrete event dynamic system (DEDS) is in general difficult due to the large search space and the simulation-based performance evaluation. Inspired by Ordinal Optimization (see, for example, [70] or [104]), which has become an important tool for optimizing DEDSs, we create an Ordered Performance Curve (OPC) as follows. For each policy, we simulate the EMS region for an amount of time, and use the measured fraction of late arrivals as an estimate for the true performance of the policy.[3] Then, we sort the policies by their estimated performance, giving us the desired OPC. At first, we look into the case where there are relatively few incidents, i.e., $\lambda = 1/45$ (per minute). In this case, we evaluate each policy with 10 simulated days. For the corresponding OPC, see Figure 4.2a. According to the theory of Ordinal Optimization, the shape of this OPC indicates that there are many good solutions (policies) for this problem [70].

However, it would be incorrect to conclude that this is true for all static ambulance positioning problems. In fact, our experiments show that changing the incident rate λ, while keeping all other parameters the same, already affects

[3] We start with an empty system, i.e., no incidents have occurred. Therefore, we need to allow the system some time to evolve towards a more natural and representative state. We disregard the first five simulated hours in each run, and only consider the performance of the remaining time.

the shape of the OPC. For $\lambda = 1/13$, the OPC is shown in Figure 4.2b. For this case, we evaluate each policy with 2.9 simulated days, which boils down to the same expected number of incidents per evaluation as in the $\lambda = 1/45$ case. First of all, note that the best static solution for this problem seems to have a performance of 17% (compared to 1% in Figure 4.2a). An increase was to be expected, because the same number of vehicles needs to serve a higher number of incidents. Perhaps more surprising is that also the shape of the OPC has changed. For Figure 4.2b, the OPC indicates that there exist only a few good static policies for this problem.

In order to determine the best static policy, we perform longer simulations to explore the region of the good solutions with more accuracy. Note that when λ changes, the optimal static policy may change as well. In fact, we find that for $\lambda = 1/45$ the best static policy is (City 1, City 1, City 2, C, C), while for $\lambda = 1/13$ the best static policy is (City 1, City 1, City 2, City 2, C).



(a) $\lambda = 1/45$

(b) $\lambda = 1/13$

Figure 4.2 OPC curves for static policies in the same region, for two different incident intensities.

DMEXCLP versus the best static policy

We now compare the performance of dynamic MEXCLP (DMEXCLP) with the best static policy. We will test our method on multiple scenarios, to show that the method gives good results for more than just one specific problem instance. We create different problem instances by changing the value of λ. Since we keep the number of vehicles equal to five, by varying λ we also vary the load of the system. In Figure 4.3, it shows that the DMEXCLP policy outperforms the best static policy for every choice of λ. When we let λ take even more extreme values, we see that DMEXCLP has approximately the same performance as the best static solution. This occurs when $\lambda = 1/9$, in which case the expected fraction of late arrivals for both the best static and the DMEXCLP solution is around 67%. A fraction this high will never be acceptable in real life, and would indicate that more vehicles are needed. Therefore, we should not draw conclusions on the applicability based on this parameter choice. Note that, even if the performance of DMEXCLP is equal to the performance of the best static policy, DMEXCLP

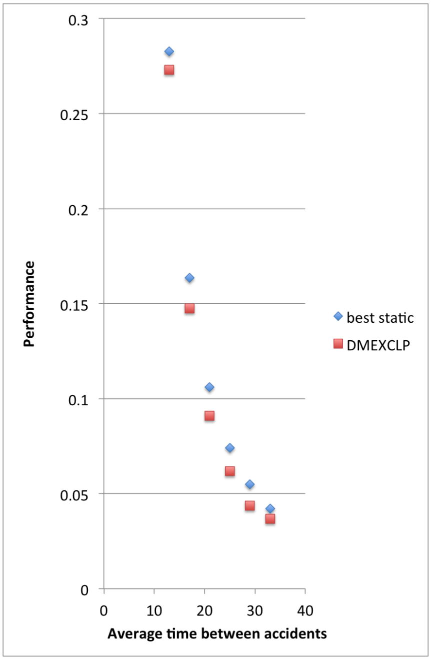


Figure 4.3 The absolute performance (expected fraction of late arrivals) of Dynamic MEXCLP compared to the best static policy. The horizontal axis displays the average time between incidents in minutes. Each policy was evaluated long enough such that the tolerance interval (1.96 times the sample standard deviation) is within 2.5% of our estimated value.

is still useful in the sense that its calculations are faster than the search for the best static policy.

In Figure 4.4 we see that the relative performance improvement for this region can be as high as 20%. In the following section we will investigate whether this number is representative for a more realistic region with demand aggregated on a smaller scale.

4.4.2 A realistic case study

In this section, we validate our redeployment method on a realistic problem instance. We chose to model the region of Utrecht, which was described in Section 2.7. For the parameters used in the implementation, see Table 4.3. This is a region with multiple hospitals, and for simplicity we assume that the patient is always transported to the nearest hospital, if necessary.

In the Netherlands, the time target for the highest priority emergency calls is fifteen minutes. Usually, three minutes are reserved for answering the call, therefore we choose to run our simulations with $T = 12$ minutes. The driving times for EMS vehicles between any two nodes in V were estimated by the RIVM

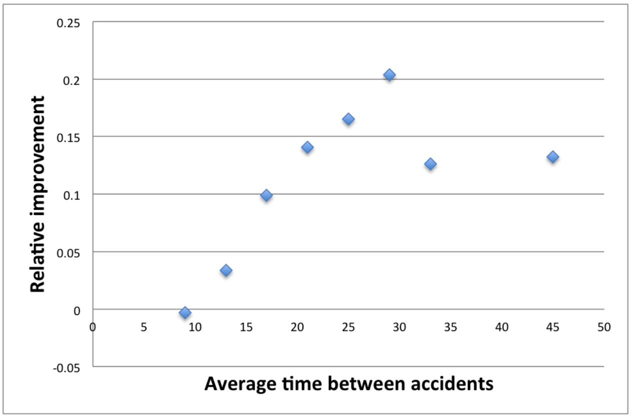


Figure 4.4 The relative improvement in performance of Dynamic MEXCLP compared to the best static policy. The horizontal axis displays the average time between incidents in minutes. Each policy was evaluated long enough such that the tolerance interval (1.96 times the sample standard deviation) is within 2.5% of our estimated value.

parameter	magnitude	choice
λ	1/6.4 minutes	Realistic for urgent calls on a weekday in this region.
A	19	Realistic number to cover demand.
W	19	Base locations as existing in 2013.
V	217	4 digit postal codes.
H	10	The hospitals within the region in 2013, excluding private clinics.
τ_{ij}		Driving times as estimated by the RIVM.
d_i		Fraction of inhabitants as known in 2009.

Table 4.3 Parameter choices for our implementation of the region of Utrecht.

in 2009 [66, Chapter 3]. These are driving times with the siren turned on. For ambulance movements without siren (e.g., when repositioning) we use 0.9 times the speed with siren. The number of vehicles used in our implementation is such that a good policy gives a performance (expected fraction of late arrivals) of a magnitude that is realistic for practical purposes.

Results

We compare the performance of the DMEXCLP solution with a benchmark. We let the benchmark be the static MEXCLP solution, which is generally assumed to give a good static policy (for a comparison of static methods, see [16]). Note that the verification of the value of one single policy is not feasible within polynomial time. Therefore, it is not tractable to perform a brute force search over all static policies using nineteen base locations and nineteen vehicles. Since there is no alternative known to compute the optimal static solution, this means we cannot use the optimal static solution as a benchmark.

In both the static (benchmark) and the dynamic (proposed solution) case, we initialize the locations of the ambulances according to the static MEXCLP solution. We simulate the EMS system ten times per policy and compare the results in Figure 4.5. We measure the fraction of late arrivals, which decreased from on average 9.5% to 7.9%. This is a difference of 1.6 percentage point, and a decrease of 16.8%. This is a significant improvement that can be made without purchasing extra vehicles or increasing the number of crew shifts. Furthermore, this improvement is large in comparison to other results in literature (e.g., an improvement from 26.7% to 25.8% in [78], which boils down to a 3.4% gain).

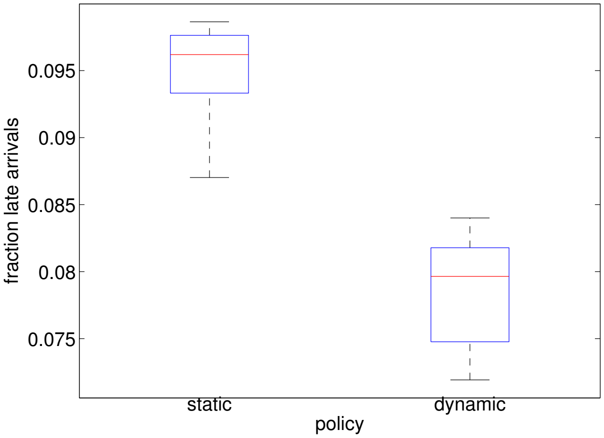


Figure 4.5 Comparing the performance of Dynamic MEXCLP with the static MEXCLP solution. For both policies a value of $q = 0.3$ is used. Each policy was evaluated with 10 runs of 500 simulated hours.

We emphasize that the dynamic MEXCLP policy does not only reduce the expected fraction of late arrivals, but also reduces the average response times overall. This can be concluded from Figure 4.6.

4.4.3 Sensitivity to the busy fraction

We investigate the sensitivity of Algorithm 1 to the parameter q, the busy fraction. To this end, we keep the number of vehicles equal to nineteen, and we also keep the average time between incidents equal to 6.4 minutes. We run the DMEXCLP algorithm for several values of q, and compare the performance in Figure 4.7. We conclude that, at least for this particular problem instance, the

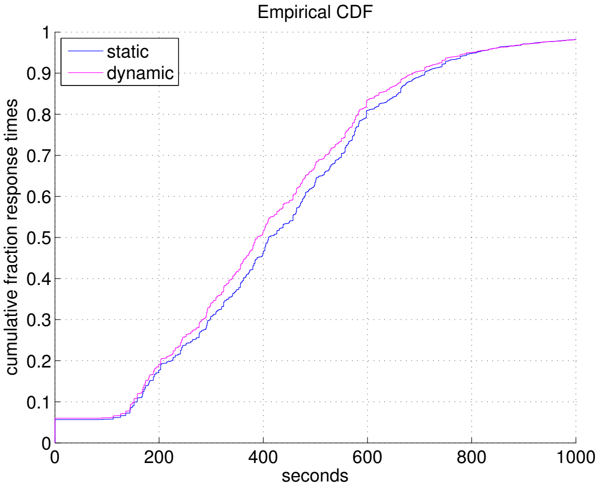


Figure 4.6 Response times for dynamic MEXCLP and the static MEXCLP solution. For both policies a value of $q = 0.3$ is used. Each policy was evaluated with 2,500 simulated hours.

quality of the solution is highly insensitive to the value of q.

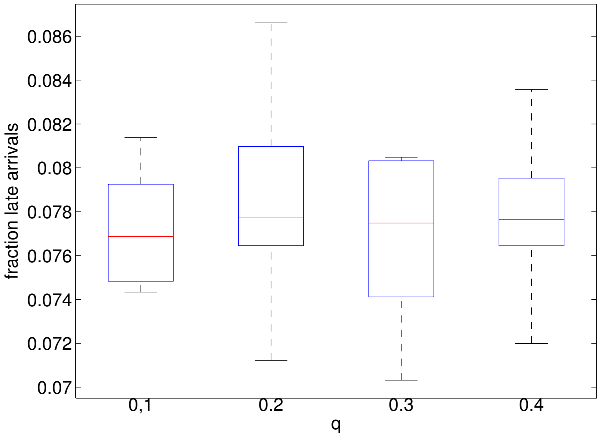


Figure 4.7 Comparing the performance of DMEXCLP for several values of q. The boxes consist of ten runs, in which we simulate 1000 hours, each.

4.5 Discussion

In this chapter we have developed real-time scalable algorithms for dynamic ambulance redeployment with a focus on minimizing the expected fraction of late arrivals. We have introduced a DMEXCLP heuristic (see Algorithm 1) that reduces the expected fraction of late arrivals by relatively 16.8% compared to a good static policy. Additionally, the DMEXCLP heuristic also reduces the average response times overall. The heuristic depends on the busy fraction, i.e., the fraction of time that an ambulance is unavailable, that needs to be estimated. Our experiments indicate that good performance is still obtained, even if there is an error in the estimate of the busy fraction.

We believe that the simplicity of our algorithm is in fact its strength: it makes it easy for researchers and practitioners to implement, and also makes it a suitable base for extensions. This belief is confirmed by the fact that several other studies each implemented an extension to the DMEXCLP algorithm [33, 50, 119].

Note that we use the fraction of inhabitants as our choice for d_i. In reality, the fraction of demand could differ from the fraction of inhabitants. However, the number of inhabitants are known with great accuracy, and this is a straightforward way to obtain a realistic setting. Furthermore, the analysis of robust optimization for uncertain ambulance demand in [61] indicates that we are likely to find good solutions, even if we make mistakes in our estimates for d_i.

In terms of applicability, we find it useful to consider whether the DMEXCLP heuristic is still feasible when we relax some of our assumptions. We address the following cases.

Changes during the day
In practice, EMS systems may deal with characteristics that change over the course of a day. This is reflected in time-dependent parameters in our model. We mention a few examples.

- Incident probabilities may shift, for example, an incident is more likely to occur in an industrial area during office hours.
- Travel times may be longer in rush hour, or may depend on the weather.

Changing parameters over time, such as the examples above, are often difficult to incorporate in a solution. However, in our case, there is no need to complicate the algorithm. At any decision epoch, a new set of parameters could be used. The question remains how to choose relevant parameters. One should keep in mind that there is only a limited number of decision epochs. Hence, a redeployment decision should not necessarily use the parameters of the system *at the exact decision moment*, but parameters that are relevant for the upcoming period. The choice of the period size may depend on the EMS region, but for example 30 minutes would be a good starting point.

Stochastic travel times
One straightforward way of dealing with stochastic travel times is to use the expectations $E[\tau_{ij}]$ in Algorithm 1. Alternatively, one could use for example the 0.8 quantile of the driving time distribution, i.e., the number X_{ij} such that $P[\tau_{ij} \leq X_{ij}] = 0.8$. This showed to give a good performance in some additional numerical work that we performed. The performance will generally depend on the exact distribution function chosen, and we suggest some preliminary experiments to obtain a good strategy.

Acceptance by crew
Staff members that come from a 'static' work environment may be used to having their own, fixed home base. Giving up this concept can be difficult. Although our proposed method already limits the relocation moments, extra adjustments can be made to accommodate the staff. For example, a good compromise would be the following. Each vehicle (and the corresponding crew) still has its own, fixed home base. Preferably, we send the vehicle to this home base, but we may choose another base if the expected gain is large enough. One can measure this by calculating the marginal coverage that would be obtained if we were to send the vehicle to its own home base, and compare this with the marginal coverage that could be obtained by a relocation. The vehicle could be relocated if and only if the difference in coverage is greater than a certain threshold.

Rural regions
Our algorithm was designed with a busy (urban) area in mind. For rural regions, however, the same technique may still be applicable, albeit with some adaptations. A key observation is that rural regions have a lower incident frequency - which is directly related to the frequency at which ambulances become idle. This implies that there will be fewer relocation moments, and therefore we expect performance improvements to be smaller. In order to overcome this, we suggest adding some additional relocations.[4] For example, one could allow a relocation when a new incident arrives. In addition, it is possible to allow *two* vehicles to relocate upon completion of an incident. The decision on where to send the vehicles, can still be made using the DMEXCLP method.

Multiple targets
In some countries there exist multiple time targets, depending on the urgency of the situation. For example, in the Netherlands, the highest priority incidents have to be reached within 15 minutes, and the less severe (but still urgent) incidents have to be reached within 30 minutes. We advise to apply the DMEXCLP algorithm using the most stringent time target. Our preliminary numerical experiments regarding realistic use cases indicate that this results in a policy that also has a good performance for a target of 30 minutes.

[4] This will obviously increase the workload for the crew, but we think this is acceptable since a rural region is typically not very busy.

Several of the above-mentioned adaptations were studied in [7]. This paper uses trace-driven simulations based on real-life datasets of two ambulance providers in the Netherlands. It showed that (1) adding more relocation decision moments is indeed highly beneficial, particularly for rural areas, (2) replacing the 0-1 coverage performance criterion by a smoothed version has a very small impact on response times, and (3) the inclusion of busy ambulances in the state description of the system leads to a small reduction in workload, but did not really improve response times. In addition, [7] considers (4) chain relocations and (5) time bounds on the execution of an ambulance relocation.

4.6 Implementation in practice

The DMEXCLP repositioning algorithm was implemented in practice [27]. During several periods in 2015, the relocation moves were displayed on a screen in the EMS callcenter in Flevoland (see Figure 4.8).

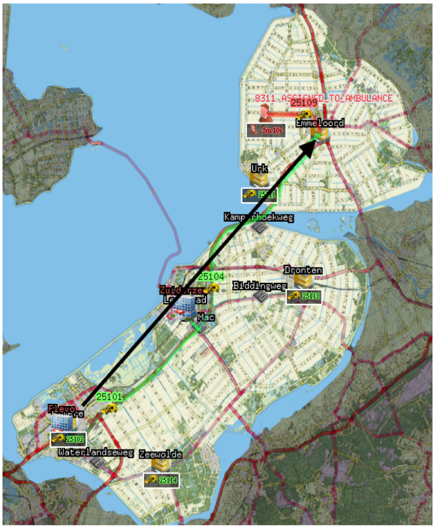


Figure 4.8 A screenshot of the pilot in Flevoland.

In collaboration with the EMS managers of GGD Flevoland [43], some practical adaptations were made to the DMEXCLP algorithm. Due to the low incident frequency in the region, we created extra decision moments: a relocation was allowed whenever a vehicle became idle *or* busy. Furthermore, some moves were rather long trips, taking half an hour or more. We decided to split those travels in two, if possible, by repositioning two vehicles that resulted in the same net move. Note that this is beneficial for the system because the desired configuration is reached quicker.

The dispatchers generally followed the relocation advice, unless they had good arguments not to. In the initial phase, we discussed all situations for which the dispatchers disagreed. This lead to interesting conversations and new insights on both ends. In some cases, the situation was simply too complex for a single person to oversee, and the system made better decisions than the dispatchers. In other cases, the dispatchers were right, and often this was because they were able to use more information than our software tool.

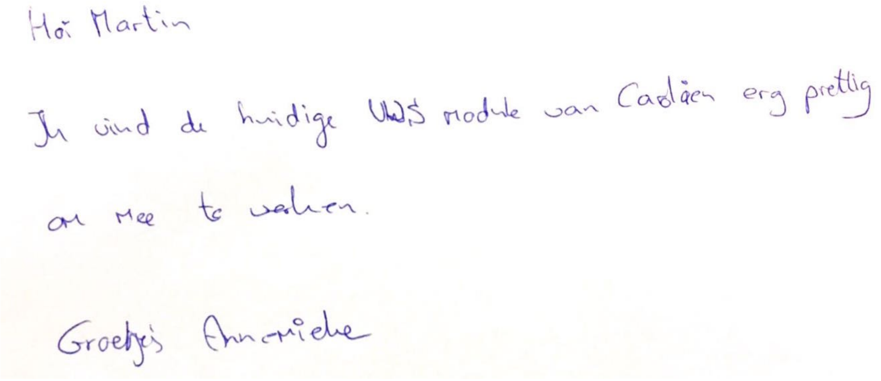
For example, EMS region Flevoland works with very long shifts in one particular part of the region. There are special labor rules that prescribe how many hours of such a shift the staff is allowed to be away from their home base. If the crew has already used their allowed hours, they may reject to be dispatched even for a severe incident. Therefore, if dispatchers realize that such a crew is already close to their maximum number of hours away from base, they choose not to relocate this vehicle even though our algorithm may suggest it. The introduction of our system has inspired a discussion on how to handle such shifts, alongside the question of whether such shifts are really desirable for the region.

Sometimes ambulances travel quite far outside of the EMS region to drop a patient off. When such an ambulance is returning, our modelling choice for moving vehicles - pretending they are at their destination - leads to a large overestimation of the coverage provided. These situations occurred more often than we had previously anticipated. Thereto, we decided to only include ambulances in the coverage calculations if they are reasonably close to the EMS region.

The pilot period was benchmarked against the same period a year earlier. The DMEXCLP algorithm seemed to perform better than the benchmark, but we find it hard to determine the significance of this result due to the limited number of observations, the large amount of randomness in the EMS process and the fact that demand increased compared to a year earlier. For a discussion of the numerical results of the pilot, we refer to [27].

Dispatchers quickly got used to working with the new screen. Generally they shared the opinion that it was a pleasant way of working, for several reasons. First of all, the introduction of the system lead to the exchange of views and new insights on what makes for a good relocation decision. Second, the performance of the EMS region became more consistent because it was no longer strongly affected by the individual dispatchers at work. Third, once the dispatchers got used to the system, it made their job less stressful and allowed them to shift focus from dispatch decisions to the communication with the patients and the ambulance crew. See also Figure 4.9.

Currently, the software is developed further in order to add more practical features. For example, the software will include information about the shift start- and end times, such that it can steer ambulances towards their finish location when the end of their shift is approaching. Furthermore, ambulances should not be sent towards a base where a new shift will start shortly. These developments are being done under the name of Stokhos B.V. [108]: a spin-off company founded after the success of the Flevoland pilot.



Hoi Martin

Ik vind de huidige UWS module van Carlien erg prettig om mee te werken.

Groetjes Annemieke

Figure 4.9 Feedback during the pilot: EMS dispatcher Annemieke thinks the DMEXCLP algorithm is very pleasant to work with.

5
Fairness in the ambulance location problem

We discuss how to position ambulances across an EMS region in a 'fair' way. Ambulance literature often focuses on maximizing the number of people served, regardless of where they live. This is equivalent with optimizing a utilitarian Social Welfare Function (SWF). It is well known that such an approach benefits people living in cities, at the cost of people living in remote areas. An often mentioned alternative is equity: providing the same service to people at every location. However, this gives so much focus on helping people in remote locations, that it usually leads to poor overall performance. Instead, we propose to use the so-called Bernoulli-Nash SWF. This may be viewed as an appealing compromise between the two solutions above. We formulate and solve models that maximize the Bernoulli-Nash social welfare. The most straightforward model maximizes coverage, but we also use more complex measures such as survival functions. We juxtapose the Bernoulli-Nash optimal solution with the Utilitarian optimum, and show how the results differ depending on the load of the system. Calculations are done for a realistic EMS region in the Netherlands.

This chapter is based on:
C.J. Jagtenberg, A.J. Mason and O.M. Dowson. Fairness in the ambulance location problem. *In preparation.*

5.1 Introduction

A key issue in EMS planning is the ambulance location problem: how and where to locate vehicles in order to effectively cover future demand. Much research has been focused on solving variants of this problem, and the majority has approached the problem from the same angle: their objective is to help as many patients as possible. For example, in overview papers from 2003 [22] and 2011 [71] practically all models aim to maximize the (expected) coverage.

Maximizing the number of people served seems natural in the context of ambulance planning. In fact, at first sight it seems hard to reason that we should help *fewer* people rather than more. However, as we will argue in this chapter, there may be reasons to consider ambulance configurations other than the one that maximizes the number of people served. For example, in order to serve as many people as possible given a fixed number of resources, a planner inadvertently moves resources to densely populated areas - at the cost of people living in rural areas. This certainly is an efficient use of resources, but the question arises how to distribute ambulances in a *fair* way.

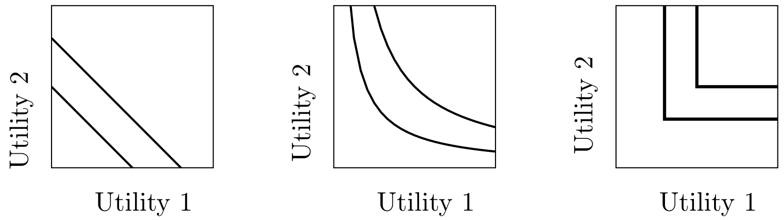
There are examples in literature of models that aim for equity. McLay et al. [48] review how equity can be modeled in the context of allocating public resources, and conclude that in general there is no single, best way to do so. However, when it comes to EMS problems they note that equity is almost always interpreted as 'having equal outcomes for all patients, regardless of where they are living'.[1] Therefore, the aim for equity often results in egalitarian (also known as maximin) models: models that maximize the level of service to the people that are the hardest to reach. It is not surprising that this generally leads to poor performance of the overall system. This is a major drawback of those models, and it would be hard to convince anyone to actually apply such solutions in practice. Furthermore, we disagree with the statement that 'equal outcomes for everyone' is the correct definition of fairness in an ambulance context. In fact, we argue that 'equal outcomes for everyone' is far from 'fair', because it is obvious that it requires much more resources to serve those people who live far away. Given these two extreme solutions (maximizing the number of people served, and equal service for everyone), we believe that a truly fair solution is some sort of compromise between them.

There exist a few papers on ambulance location problems that explicitly include a form of fairness in the objective. For example, in [31] the sum of 'envy' among all demand zones is minimized. In [32] the authors propose three bi-objective covering models, for which fairness is a secondary objective. They consider the following three options (1) minimize the maximum distance between each uncovered demand zone and its closest opened station, (2) minimize the number of uncovered rural demand zones, and (3) minimize the number of uncovered demand zones. Although these papers are all based on ideas similar to ours - that some form of fairness should be incorporated in the objective - none of these take our approach, which we will describe next.

In this chapter we will view ambulance location problems from the perspective of social welfare. Social welfare is measured as a function of the 'utilities' of individuals or subgroups of a society. That way, different social welfare functions (SWFs) represent different objectives on how to balance fairness and efficiency. For example, maximizing the total number of people served is equivalent with a so-called *utilitarian* SWF: that is, maximize the sum of the individual utilities. Alternatively, aiming for equal outcomes for everyone would correspond to an egalitarian SWF. In this chapter, we propose and investigate a third option, the so-called Bernoulli-Nash SWF. The Bernoulli-Nash SWF is defined as the *product* of individual utilities. These three different SWFs can be visualized by so-called *social indifference* curves: these are solutions which are equivalent in terms of

[1] For more background regarding equity in ambulance planning, see also [81]. They discuss several ideas regarding equity, e.g., server equity (as opposed to patient equity). We consider this an interesting alternative point of view, but not the focus of our work.

social welfare. For a small problem instance with just two individuals, these curves are shown in Figure 5.1. To the best of our knowledge, the Bernoulli-Nash SWF has not previously been applied to ambulance location problems. We juxtapose the Bernoulli-Nash optimal solution with the often-used utilitarian optimum, and show how the results differ depending on the load of the system.



(a) Utilitarian SWF **(b)** Bernoulli-Nash SWF **(c)** Egalitarian SWF

Figure 5.1 Social indifference curves for three different social welfare functions.

The Bernoulli-Nash SWF is related to the Nash bargaining problem [88], which was defined for two players. As Nash defines the problem, both players have to agree on the outcome, or otherwise a so-called *disagreement point* (denoted ξ) is reached. Let f and g be the utility functions for the two players. The optimal solution (obtained at the point θ) is defined relative to this ξ: it is the maximum of the SWF:

$$\arg\max_{\theta}(f(\theta) - f(\xi))(g(\theta) - g(\xi)).$$

As [19] describes it: "When the Nash bargaining solution is used, it is to predict what the result would be, under certain ideal circumstances, if specimens of *homo economicus* were to bargain optimally."

To interpret this in the context of ambulance planning, imagine these two players to be patients living in different locations - or two communities, together deciding on how to distribute their ambulances. It is not unreasonable to imagine this decision to be a bargaining process. Let us say that the disagreement point is that no ambulances will be acquired for the region, so we have $\xi = 0$. An optimal Bernoulli-Nash social welfare then comes down to the same as Nash's bargaining solution.

The definitions of SWFs are based on *utilities*; however, it is not immediately clear how the utility of a patient should be defined. This utility should somehow represent the happiness of that patient, depending on his location with respect to the locations of ambulances. Typically, their happiness will be a function of the (expected) ambulance response time, e.g., the probability that an ambulance will reach the patient within a certain time threshold. Other - more advanced - measures are also possible; our work includes three different definitions of utility.

The rest of this chapter is structured as follows. In Section 5.2 we define the problem. In Section 5.3 we define three different measures that can be used as utilities. Section 5.4 provides a small example that helps to build intuition for the problem. In Section 5.5 gives our optimization models, followed by a case study in Section 5.6. We finish with a discussion in Section 5.7.

5.2 Problem formulation

In this chapter we focus on the allocation of ambulances to a set of base stations with known locations. These ambulances respond to incidents that occur at *demand nodes.* Denote the set of demand nodes V. After completing service, the ambulance should return to its own home base. We address the question of which ambulance base locations to open as well as how to divide the vehicles over the bases. These questions may be addressed separately, but then obviously optimality is not guaranteed. Instead, we aim to find a base location for each vehicle, choosing from a large set of potential locations - many of which may be unused in the final solution. Note that multiple vehicles are allowed to have the same home base.

Our goal is to find a distribution of vehicles over bases that maximizes the Bernoulli-Nash SWF. The Bernoulli-Nash SWF is defined as the product of individual utilities. In the context of ambulance planning, we write the product as follows. Let u_i be the utility of a person at node i, and let d_i be the demand fraction at node $i, i \in V$. The Bernoulli-Nash SWF is then given by

$$\prod_{i \in V} u_i^{d_i}.$$

In this chapter, we compare the Bernoulli-Nash SWF to the often-used utilitarian SWF, which is denoted as

$$\sum_{i \in V} d_i u_i.$$

For completeness, we also state the egalitarian SWF:

$$\min_{i \in V} u_i.$$

Before we elaborate on definitions and interpretations of the utilities u_i, we find it useful to illustrate the differences between the three SWFs above. To that end, we introduce a small example of an ambulance location problem. Unlike the rest of this chapter, this particular example is not meant to be realistic. We use a simple utility measure that is good for demonstrative purposes because it helps to build intuition on (1) how the Bernoulli-Nash SWF relates to the two other SWFs, and (2) why we think the Bernoulli-Nash SWF is a somehow reasonable measure for positioning ambulances.

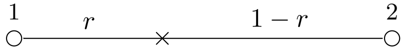


Figure 5.2 A toy example for the ambulance location problem

Example. Imagine two areas or villages, together acquiring one ambulance. We model these areas as nodes, labeled 1 and 2, and the ambulance may be positioned anywhere along the line between them. Without loss of generality, assume the distance between the two nodes is normalized to 1. The two nodes both contain a proportion of the demand, d_1 and d_2, such that $0 < d_1 < 1$ and $d_1 + d_2 = 1$. The position of the ambulance can be defined by its distance from node 1, let us call this distance r. This is depicted in Figure 5.2.

We define the utility u_i of an inhabitant of node i, $i \in \{1, 2\}$, to be equal to 1 *minus* the distance between i and the ambulance. This means, for example, that if the ambulance is placed at node 1, then $u_1 = 1$ and $u_2 = 0$. More generally, if the ambulance is located distance r from node 1, then $u_1 = 1 - r$ and $u_2 = r$.

Now let us compare the optimal solutions for the three SWFs. The utilitarian SWF is maximized when the ambulance is placed at the node with the most inhabitants. If $d_1 = d_2$, then all solutions are optimal from a utilitarian perspective. For the egalitarian SWF on the other hand, it is straightforward to see that this is maximized when $r = 0.5$. We finish with Bernoulli-Nash optimum, which can be found as follows. Recall that the Bernoulli-Nash SWF is given by $u_1^{d_1} \times u_2^{d_2}$. This means that if we place the ambulance at either of the two nodes (i.e., $r = 0$ or $r = 1$) the Bernoulli-Nash SWF is equal to zero. Therefore, for an optimal solution $0 < r < 1$ will hold. Furthermore, observe that maximizing $u_1^{d_1} \times u_2^{d_2}$ is equivalent with maximizing $d_1 log(u_1) + d_2 log(u_2)$. Denote this function $f(r)$, as the utilities can be expressed in terms of r:

$$f(r) = d_1 log(1 - r) + d_2 log(r).$$

The maximum of $f(r)$ is attained when the derivative is equal to zero:

$$f'(r) = \frac{-d_1}{1-r} + \frac{d_2}{r} = 0,$$

which is equivalent with

$$d_1 r = d_2(1 - r).$$

If we now use that $d_2 = 1 - d_1$ we can solve the equation:

$$r = d_2.$$

That is, the ambulance should be positioned between the two nodes, such that *the ratio of the distances is inversely proportional to the ratio of the demands.* In our opinion, this corresponds to a fair distribution of ambulances, that balances the distances depending on the ratio of the inhabitants, and thereby provides an attractive compromise between the utilitarian and egalitarian solution.

The simple utility function defined in the example above is not a common performance measure for ambulances. Therefore, we continue by introducing other, more realistic, utilities.

5.3 Utilities

In this section we discuss three different definitions for *utility* that are related to key performance indicators used by most ambulance practitioners and researchers. These three utilities will be used and compared throughout the rest of this chapter.

The utility of demand node $i \in V$ can be interpreted as a measure of how happy an inhabitant of i is with the ambulance configuration. Although ambulance literature typically does not use the term utility, several models exist that optimize for different quantities. Such a quantity is almost always a function of the ambulance response time.

A straightforward example of a utility is *single coverage* (also known as regional coverage). This quantity is defined in terms of a response time threshold (RTT). Let T denote the value of this threshold. Simply put, a demand node has (regional) coverage 1 if there is a vehicle positioned at most T minutes away. Otherwise, the node is said to have coverage 0. The early models in ambulance location literature typically used single coverage as their utility (e.g., [34, 110]). Note that this definition of coverage is somewhat shortsighted: (1) a single vehicle may not be enough to fully satisfy inhabitants of node i (because sometimes this vehicle will be busy serving other patients), and (2) one may argue that such a strict threshold is somewhat unrealistic: a vehicle that is slightly further than T away should be worth almost as much as a vehicle at distance T. It may be clear that single coverage is an overly simplified measure to use for our utility; we aim for a more sophisticated measure instead. We next describe how to overcome the issues described above.

First of all, we should account for ambulance unavailability: literature describes various ways to do this. The Maximum Expected Coverage Location Problem (MEXCLP) [36] uses a so-called *busy fraction*: a fixed parameter that represents the probability that any given ambulance is busy at any given time. In this model, ambulances are assumed to operate independently. It should be noted that modeling unavailability this way is somewhat of a simplification: in reality vehicles are not independent, and moreover, the busy probabilities might differ between vehicles. Other ways of modeling ambulance unavailability include Erlang loss models [96], scenarios [40] and simulation, although the latter is more common when planning in a dynamic context (e.g. [118]). We decided to model ambulance availability using a busy fraction. Although there exist alternatives that may be more accurate, incorporating this in the utility makes the model far more complex. We chose not to do this, since our objective - a product of utilities - is already a hard function to optimize. This is further addressed in the discussion (Section 5.7).

Second, the model becomes more realistic if we can relax the assumption that the utility is a 0-1 function of the response time. If instead we want to use some continuous function of the response time, the model becomes harder. However, it remains possible to solve it with modern solvers and hardware. We implemented our models using three different definitions of utility, two of which relax this 0-1 assumption. The rest of this section describes those three utilities.

5.3.1 Definitions

Deterministic coverage

The most straightforward utility function that we will consider is a coverage model. Throughout the chapter we refer to this utility function as *deterministic coverage*, where 'deterministic' refers to the underlying assumptions in the driving time model.

The coverage of demand node i can be defined in terms of how many ambulances are located within the RTT of i: let $k(i)$ denote this number of vehicles. The utility (coverage) of $i, i \in V$ is then given by the probability that at least one of these vehicles is idle, i.e., $c_i = 1 - q^{k(i)}$.

It is appealing to assume travel times are deterministic, for several reasons. First of all, it is quite difficult to accurately estimate a response time - let alone a whole distribution of response times. Second, stochastic travel times are harder to incorporate in optimization models. Doing this leads to less efficient solutions and scalability issues.

Stochastic coverage

As opposed to deterministic coverage (as discussed above), this section deals with coverage when travel times are stochastic. Although it is appealing to assume that travel times are deterministic, it may be argued that stochastic travel times are more realistic, see e.g. [54]. In this context, our values for τ will be interpreted as *expected* travel times.

The utility is then defined as the *probability* that an ambulance will reach the scene of the incident within the RTT. We can compute this probability if we assume that (1) ambulance unavailability may be modeled using a busy fraction, (2) the closest idle ambulance always responds, and (3) the distribution of the travel times is known.

As described in [17], stochastic travel times for Dutch ambulances may be approximated by a normal distribution with a coefficient of variation of 0.25. Therefore, we compute the probability that an ambulance with expected travel time τ arrives within 12 minutes (the RTT). The result is depicted in Figure 5.3.

Survival

While the majority of ambulance literature deals with response time threshold, it is sometimes argued that these do not adequately differentiate between consequences of different response times. That is, even if one uses a well-chosen

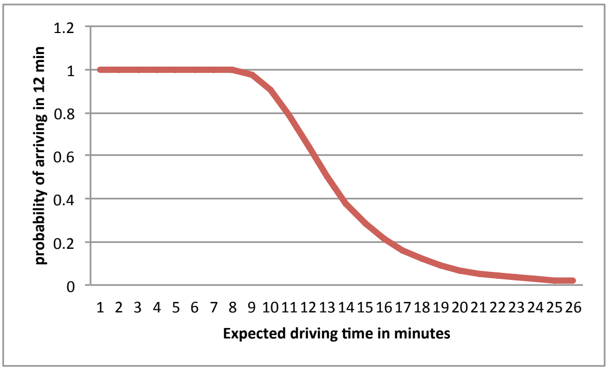


Figure 5.3 The probability that an ambulance will arrive within the time threshold, when driving times are normally distributed with a coefficient of variation of 0.25.

distribution of stochastic travel times, such a model still fails to accurately represent the happiness of a patient given his or her response time. Such discussions usually result in an argument for using a *survival function*: a monotonically decreasing function of the response time that returns the probability of survival for the patient.

We analyzed several survival functions mentioned in literature. As noted in [38], almost all of the published research relating survival rates to EMS response times focuses on cardiac arrest. Survival is typically interpreted as 'survival until discharge from the hospital'. For the purpose of this chapter, such practical considerations are of limited importance. Our main goal is to show that our model can find a solution resulting in maximum survival, the specific survival function used is mainly illustrative of the idea.

We implemented two different survival functions. The first was introduced by De Maio et al. in [73] and is given by:

$$f(\tau) = (1 + e^{0.679+0.262\tau})^{-1}. \tag{5.1}$$

The second (by Valenzuela et al. [113]) uses variables that measure the time from collapse to CPR (τ_{CPR}), and from collapse to defibrillation (τ_{defib}). The survival probability is then given by:

$$f(\tau_{CPR}, \tau_{defib}) = (1 + e^{-0.260+0.106\tau_{CPR}+0.139\tau_{defib}})^{-1}. \tag{5.2}$$

As in [38], we assume that CPR is performed by the responding EMS unit immediately upon arrival, and defibrillation is performed one minute after arrival. Equation 5.2 then becomes:

$$f(\tau) = (1 + e^{-0.260+0.106\tau+0.139(1+\tau)})^{-1}. \tag{5.3}$$

The two survival functions (5.1) and (5.3) are depicted in Figure 5.4.

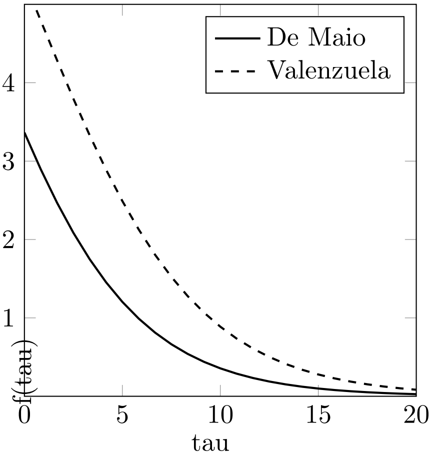


Figure 5.4 The two different survival functions. The horizontal axis represents the response time in minutes. The vertical axis is the probability of survival.

The results for the two different survival functions turn out to be quite similar. Therefore, we decided to only include results for one of them in this chapter. Our choice between the 'de Maio et al.' and the 'Valenzuela et al.' survival functions was mainly based on the following observation. The probability of survival is remarkably low: even if an ambulance is present right away, the probability of survival for the *most optimistic* function is still less than 55%. This may be accurate for cardiac arrest, but generally speaking one might hope that the survival probability of an ambulance request would be higher. Therefore, we decided to use the highest survival function among the two (Equation (5.3)).

5.3.2 Notation

This chapter compares two different SWFs as objectives. Furthermore, numerical work is done for three different utilities (as described in Section 5.3). When combined, this leads to six different objectives.

Our goal is to show the differences between these SWFs and the solutions that correspond to their optimum. We want to emphasize the difference between *objectives* and *models*. While a model has a certain objective, we can evaluate its solution with a different objective (that is precisely what we will do in Section 5.6). We next introduce notation in order to make the distinction between the six objectives and corresponding six models.

We will denote an objective as

$$\text{SWF}_{utility},$$

where SWF $\in$ {U, BN} means either the utilitarian resp. the Bernoulli-Nash social welfare. We denote utility $\in$ {cov, stoch, survival} to represent either deterministic coverage, stochastic coverage or survival according to the survival function by Valenzuela et al [113].

For the optimization models, we write

$$\text{model}_{utility}.$$

The utilitarian models are MEXCLP,[2] MEXSLP$_{stoch}$ and MEXSLP$_{survival}$. The models that optimize the Bernoulli-Nash social welfare are denoted as MaxFairness$_{utility}$ (because the Bernoulli-Nash SWF contains a form of fairness that the utilitarian SWF is lacking). As before, utility $\in$ {cov, stoch, survival}.

Next, we introduce a small problem instance for which we can optimize the social welfare by brute force. This allows us to provide some insights, before we continue with our optimization models.

5.4 A small example

For illustrative purposes, we analyze a fictional region with two demand nodes (demand $d_1 = 0.1$ and $d_2 = 0.9$). We take stochastic coverage as our utility function, and define the expected travel time between the two nodes to be 30 minutes. Both nodes are possible bases and our task is to place 2 ambulances in the region. Let the RTT to be twelve minutes, which - to the driving time distribution described in Section 5.3.1 - implies that the probability that an ambulance arrives on time while departing from the *other* node is ≈ 0.0082. Conversely, when the ambulance departs from the same node as where the incident is, the probability of being on time is ≈ 1. In this theoretical example we let the average busy fraction be $q = 0.3$.

For this problem instance, there are only three different solutions. We compute the utilitarian and the Bernoulli-Nash social welfare for each of those solutions to find the following optima: the Bernoulli-Nash optimal solution has one vehicle in each zone, while the utilitarian optimum[3] has two vehicles in the zone with the largest demand.[4] Table 5.1 shows the obtained social welfare of these two solutions, for both the Bernoulli-Nash and utilitarian SWF.

model	BN$_{stoch}$	U$_{stoch}$
MaxFairness$_{stoch}$	0.70172	70.17%
MEXSLP$_{stoch}$	0.56629	81.97%

Table 5.1 Max fairness vs MEXSLP, where the utility is stochastic coverage.

[2] Note that the utility of MEXCLP is always (deterministic) coverage, hence we do not have to write MEXCLP$_{cov}$ explicitly.

[3] This is equivalent to the optimal solution for the Maximum Expected Survival Location Problem (MEXSLP).

[4] The egalitarian optimum in this case also places one vehicle in each zone, but we will not focus on that.

The brute force approach that we used above obviously does not scale well. The next section introduces optimization models that allow us to compute optimal solutions for larger, realistic problem instances.

5.5 Methods

In this section we introduce the models used to optimize the Bernoulli-Nash SWF, which is the main contribution of this chapter. Our goal is to juxtapose this solution with the utilitarian optimum, hence, we also recap the corresponding utilitarian optimization models (MEXCLP [36] and MEXSLP [17]).

To define our models, we first introduce the notation in Table 5.2.

A	The set of ambulances.
V	The set of demand locations.
W	The set of possible base locations, $W \subseteq V$.
T	The response time threshold.
q	The busy fraction.
d_i	The fraction of demand in i, $i \in V$.
τ_{ij}	The driving time from i to j with siren turned on, $i, j \in V$.
n_j	The number of ambulances positioned at j, $j \in W$.

Table 5.2 Notation.

In the optimization models that we use, it is somewhat implicitly assumed that one always sends the closest idle ambulance.

As described in Section 5.3, we do numerical work for three different utilities. Optimizing for stochastic coverage or optimizing survival is done using the same model: they only differ in numerical input. It is also possible to use this same model for deterministic coverage; however, in this case a simpler and faster model is available. This results in four different models (two for each SWF), which we describe in the following subsections. We implemented them in Julia/JuMP [72], using Gurobi [49] as our solver.

5.5.1 Coverage optimization models

We next describe the optimization models that use coverage as utility.

Utilitarian SWF (MEXCLP)
Maximizing the utilitarian SWF with (deterministic) coverage as a utility is equivalent to the MEXCLP [36] (see also Section 1.2.1). MEXCLP maximizes the total coverage throughout the region.

Recall that d_i are parameters that represent the demand in zone i. We introduce variables as described in Table 5.3. The MEXCLP model uses parameters W_i, defined as the set of potential base locations that cover demand node $i (i \in V)$. That is, $W_i = \{j \in W : \tau_{ji} \leq T\}$. The MILP model can then be formulated as:

variable	for	range	meaning
c_i	$i \in V$	[0,1]	coverage of zone i
x_j	$j \in W$	$1, \ldots, \lvert A \rvert$	number of vehicles positioned at base j
y_{ik}	$i \in V$, $k = 1, \ldots, \lvert A \rvert$	{0,1}	there are at least k ambulances near zone i

Table 5.3 Interpretation of variables for the MILP formulation that maximizes coverage.

$$\text{Maximize} \sum_{i \in V} d_i c_i$$

subject to

$$c_i \leq \sum_{k=1}^{|A|} (1-q) q^{k-1} y_{ik}, \tag{5.4}$$

$$\sum_{j \in W_i} x_j \geq \sum_{k=1}^{|A|} y_{ik}, \quad i \in V,$$

$$\sum_{j \in W} x_j \leq |A|,$$

$$x_j \in \mathbb{N}, \quad j \in W,$$

$$y_{ik} \in \{0,1\}, \quad i \in V, k = 1, \ldots, |A|,$$

$$c_i \in [0,1], \quad i \in V.$$

Note that variables c_i are not strictly necessary to define this model. However, we included them for ease of reading (and this allows us to make a clear comparison with the model that maximizes the Bernoulli-Nash SWF). Note that for Equation (5.4) equality actually holds.

Bernoulli-Nash SWF
The optimization model that maximizes the Bernoulli-Nash SWF is given by:

$$\text{Maximize} \sum_{i \in V} d_i \text{log}(c_i) \tag{5.5}$$

subject to

$$c_i \leq \sum_{k=1}^{|A|} (1-q) q^{k-1} y_{ik},$$

$$\sum_{j \in W_i} x_j \geq \sum_{k=1}^{|A|} y_{ik}, \quad i \in V,$$

$$\sum_{j \in W} x_j \leq |A|,$$

$$x_j \in \mathbb{N}, \quad j \in W,$$

$$y_{ik} \in \{0, 1\}, \quad i \in V, k = 1, \ldots, |A|,$$

$$c_i \in [0, 1], \quad i \in V.$$

Since this is not a linear model, we approximate the logarithm in Equation (5.5) with piecewise linear functions. Thereto, we add a variable l_i for all demand nodes $i \in V$. The objective (5.5) is then replaced by

$$\text{Maximize} \sum_{i \in V} d_i l_i. \tag{5.6}$$

We start by introducing a few upper bounds on the value of l_i (for each $i \in V$), by adding lines that are tangent to the logarithm at different points, as depicted in Figure 5.5. These lines hold as upper bounds on the value of l_i. Then, we solve the MILP and analyze the result. If it turns out that $l_i > \log(c_i) + \varepsilon$ (here, ε is our tolerance), we add another constraint that bounds the value of l_i to the line tangent to $\log(c_i)$ at point $(c_i, \log(c_i))$. We continue until our piecewise linear approximation of the logarithm is accurate enough, i.e., all values of l_i are within tolerance of $\log(c_i)$.[5]

5.5.2 Survival optimization models

In this section we generalize the models from Section 5.5.1, using a so-called survival function. A survival function maps a response time to a survival probability. Every survival function $f(t)$ is monotonically decreasing, i.e., $f(t') \leq f(t)$ for all $t' > t$ (however, note that this is not a necessary condition for our model). Note that we can pre-compute all survival probabilities. Given the driving time t_{ji} from base j to demand zone i, we compute probabilities $p_{ji} = f(t_{ji})$ for all $j \in W, i \in V$. Hence, these probabilities are parameters of our model, *not* decision variables. As before, d_i are parameters that represent the demand in zone i, $i \in V$. We introduce variables as described in Table 5.4.

Utilitarian SWF (MEXSLP)
In the context of survival functions, a utilitarian's goal is to optimize the total survival probability. This is called the Maximal Expected Survival Location Problem (MEXSLP), and was first formulated in [38]. However, this formulation is not linear, and therefore does not scale well. Later, the same problem

[5] In our implementation, we used $\varepsilon = 10^{-5}$. For our case study with 217 nodes, this ensures that the objective value is approximated within 217 $\cdot 10^{-5} \approx 10^{-3}$ of the true value.

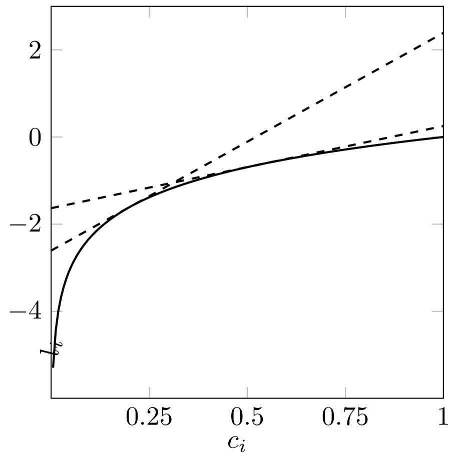


Figure 5.5 The value of l_i is bounded by linear functions (dashed lines), such that it is approximately equal to $\log(c_i)$.

variable	for	range	meaning
u_i	$i \in V$	[0,1]	utility for a patient in zone i
x_j	$j \in W$	$1, \ldots, \lvert A \rvert$	number of vehicles positioned at base j
z_{ijk}	$i \in V, j \in W,$ $k \in 1, \ldots, \lvert A \rvert$	$\{0, 1\}$	the k^{th} preferred ambulance for demand zone i is located at base j

Table 5.4 Interpretation of variables for the MILP formulation that maximizes survival probabilities.

was modeled as a MILP [17], and this is much faster to solve. Therefore, we implement a model that is similar to [17], albeit with some small changes for ease of reading and extending.

A difference with coverage models is that it is no longer sensible to precompute the set of bases that can reach zone i within the RTT. (In the coverage models this set was denoted W_i.) Instead, we need to keep track of the exact preference order of vehicles for each demand zone. Thereto, we introduce decision variables z (see Table 5.4). We next formulate the MEXSLP as a MILP.

$$\text{Maximize } \sum_{i \in V} d_i u_i \tag{5.7}$$

subject to

$$u_i \leq \sum_{k=1}^{|A|} (1-q) q^{k-1} \cdot p_{ji} \cdot z_{ijk}, \tag{5.8}$$

$$\sum_{j \in W} z_{ijk} = 1, \qquad i \in V, k = 1, \dots, |A|, \tag{5.9}$$

$$\sum_{k=1}^{|A|} z_{ijk} = x_j \qquad i \in V, j \in W, \tag{5.10}$$

$$\sum_{j \in W} x_j \leq |A|,$$

$$x_j \in \mathbb{N}, \quad j \in W,$$

$$z_{ijk} \in \{0,1\}, \quad i \in V, j \in W, k = 1, \dots, |A|,$$

$$u_i \in [0,1], \quad i \in V.$$

Constraint (5.9) ensures that only one vehicle can be the k^{th} favourite for a certain demand zone. Constraint (5.10) ensures that the number of vehicles at base j that are k^{th} favorite (for any k) is equal to the number of vehicles at base j in total. Equivalently, constraint (5.8) may be formulated with an equality sign. Note that, as before, variables u_i are not strictly necessary to implement this model, but we add them for ease of reading.

Note that if one chooses survival function f to be

$$f(t) = \mathbb{1}[t \leq T],$$

then the result is the same as the result for the models in Section 5.5.1. Therefore, the coverage optimization models may be viewed as special cases of the survival optimization models. A special case for which a more efficient formulation exists.

Bernoulli-Nash SWF

The model is identical to the MEXSLP, except we replace objective (5.7) with

$$\text{Maximize} \sum_{i \in V} d_i \text{log}(u_i) \tag{5.11}$$

As in Section 5.5.1, we deal with this nonlinear problem by creating a piecewise linear upper bound on the logarithm of u_i, for all $i \in V$. The model can then be solved with a MILP solver (Gurobi).

5.6 Computational results

We continue with a case study for which we computed numerical results. This section reports results based on an EMS region in the Netherlands. We first introduce the region, after which we discuss and compare the solutions of the different optimization methods.

5.6.1 Region

We apply our models to the province of Utrecht, which was described in Section 2.7.

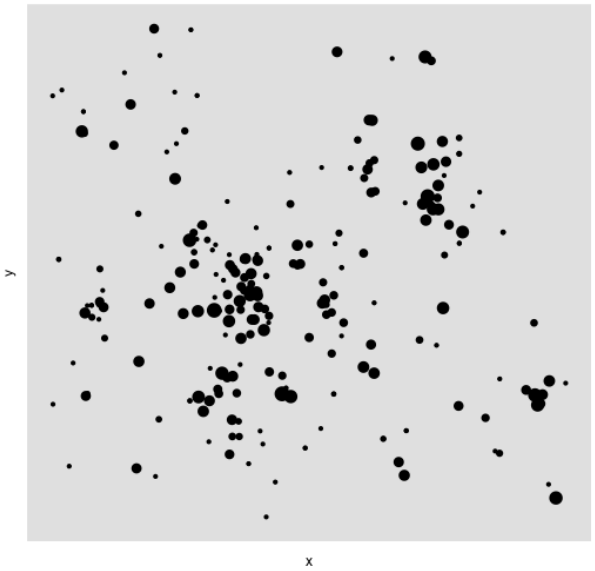


Figure 5.6 The 217 postal codes (demand nodes) of province Utrecht. The area of each node is scaled with the number of inhabitants. The driving time between two nodes that are furthest apart is approximately 58 minutes.

Utrecht is divided in 217 postal codes: these will be our demand nodes V. We will take the fraction of demand in a single node to be proportional to the number of inhabitants in that postal code. The driving times τ_{ij} for EMS vehicles between any two nodes $i, j \in V$ were estimated by the RIVM [66, Chapter 3]. The demand nodes are depicted in Figure 5.6.

For the purpose of this chapter, we want to place sixteen vehicles in the region. Recall that we want to optimize both which base locations to open, as well as how many vehicles to put at each base. Therefore, we want to consider more than just the nineteen existing base locations. However, using all 217 demand nodes as possible base locations might be quite a lot to handle, at least for our most complex models. To limit the computation time we choose a subset of 50 of these demand nodes to be our potential base locations. We want to make sure the set of potential base locations is well spread out over the region. Thereto, we formulate and solve a MIP: when two bases are opened within distance t (in minutes) from each other, we incur a penalty $e^{-0.5t}$. This implies that when two bases are opened close to one another, the corresponding penalty is very high. The objective of the MIP is to minimize the sum of these penalties. We add constraints that ensure the 19 currently existing base locations are included in

model	run time
MEXCLP	1 second
MaxFair$_{cov}$	10 seconds
MEXSLP$_{survival}$	9 minutes
MaxFair$_{survival}$	2-5 hours

Table 5.5 Run times for different optimization models. These run times are measured for the region Utrecht with 50 bases and sixteen ambulances.

the solution. This gives the 50 locations as depicted in Figure 5.7.

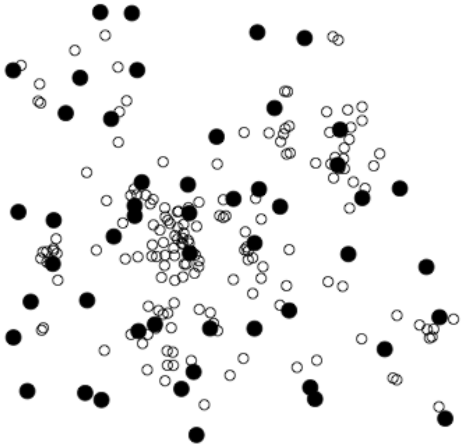


Figure 5.7 The demand locations (all nodes) and the 50 locations that we will use as possible bases (black nodes), as determined by our MIP.

5.6.2 Run times

We implemented and solved the six different optimization models for the region Utrecht as described above. The solve times are reported in Table 5.5. One immediately sees that the computational effort varies highly depending on the model. As expected, the coverage optimization models are more efficient than the survival models. Furthermore, the MaxFairness models are much harder than the MEXCLP/MEXSLP models. Note that the computation time of the MaxFairness models depend on ε. The tolerance of the difference between the linear approximation and the true value of the logarithm, as described in Section 5.5.1. The values reported in Table 5.5 are for $\varepsilon = 10^{-5}$.

5.6.3 Results

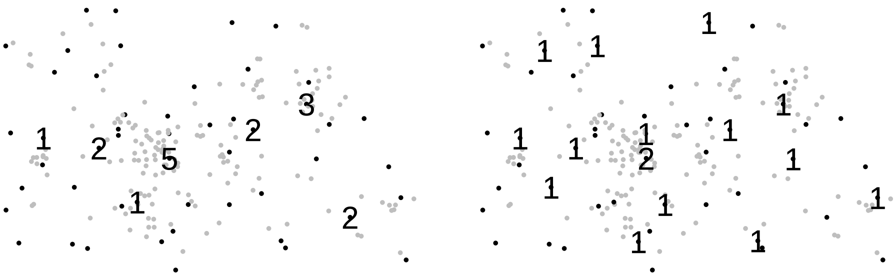
This section reports the results of the six different optimization models described in Section 5.5, applied to the region Utrecht defined in Section 5.6.1. We ran our models for several busy fractions (q), ranging from 0 to 0.9. We next show how the optimal solution for each model performs against all objectives, and highlight some of the differences between the solutions.

The Bernoulli-Nash social welfare may be computed as follows. If v represents the objective value of a MaxFairness model (e.g., the value of Equation (5.6)), then the Bernoulli-Nash social welfare is given by e^v. However, note that this is not an exact answer because of errors in the approximation of the logarithm. Instead, we explicitly calculated the product of the utilities whenever we report values of Bernoulli-Nash social welfare.

model	U_{cov}	BN_{cov}	U_{stoch}	BN_{stoch}	U_{surv}	BN_{surv}
MEXCLP	**0.614**	0	0.610	0.538	0.119	0.090
$\mathrm{MaxFairness}_{cov}$	0.518	**0.486**	0.534	0.515	0.104	0.093
MEXSLP_{stoch}	0.609	0	**0.616**	0.519	0.126	0.092
$\mathrm{MaxFairness}_{stoch}$	0.589	0	0.601	**0.565**	0.124	0.106
MEXSLP_{surv}	0.5923	0	0.6028	0.4842	**0.1305**	0.0899
$\mathrm{MaxFairness}_{surv}$	0.5766	0	0.5925	0.5551	0.1224	**0.1071**

Table 5.6 Performance of the optimal solution for each model against all objectives. These values correspond to $q = 0.75$. Note that the numbers on the diagonal correspond to the objective for that model, hence they are the maximum in each column.

The first thing to notice is that BN_{cov} is often zero. This means that in those solutions there is at least one postal code, however small, that cannot be reached within twelve minutes by any ambulance. Therefore the product of utilities is also zero.



(a) MEXCLP

(b) $\mathrm{MaxFairness}_{cov}$

Figure 5.8 The optimal solutions for the utilitarian solution and the MaxFairness solution, for $q = 0.75$. The grey nodes are demand points, the black nodes are possible base locations. The numbers represent the number of ambulances placed at each base. The utility in both cases is deterministic coverage.

Let us compare the two solutions that use deterministic coverage as their utility (i.e., MEXCLP and $\mathrm{MaxFairness}_{cov}$). We selected the solutions for $q = 0.75$, because differences become more clear for higher values of q. As Figure 5.8 shows, the two solutions are quite different. What stands out is the fact that MEXCLP places multiple vehicles at the same base (up to five), which is due to the high busy fraction: a single vehicle does not provide a lot of coverage. Hence, giving densely populated areas additional vehicles is preferred over giving areas of less demand a first vehicle. In contrast, MaxFairness$_{cov}$ shows more of a tendency to spread out over the region. Note that this same effect could already be seen on a smaller scale in the illustrative example from Section 5.4.

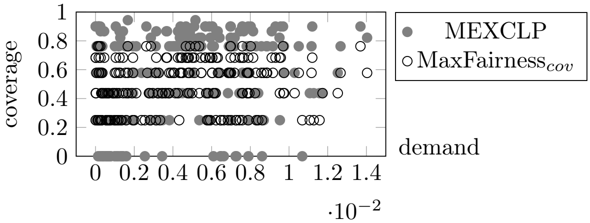


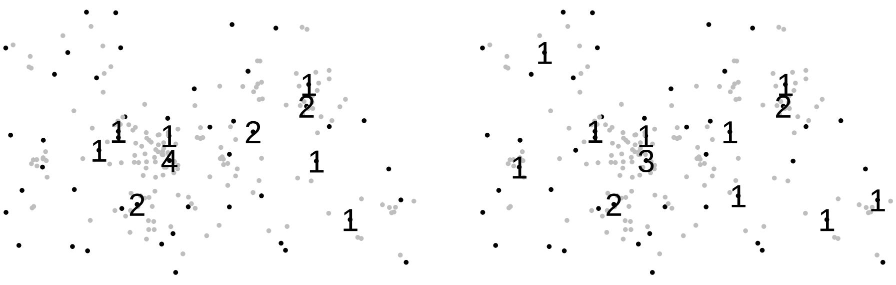
Figure 5.9 The coverage versus demand of each node, using deterministic driving times.

Let us further analyze the two different solutions depicted in Figure 5.8. Thereto, we compare the utilities (coverage) of individual demand nodes: in Figure 5.9 we plot the coverage versus the demand of each node. This shows that the MEXCLP solution gives some of the highest coverages, but also some of the lowest (even zero). The values for the MaxFairness$_{cov}$ solution are closer to one another. This is consistent with what might intuitively be considered 'fair'. Furthermore, note that the coverage only takes a few different values: this is due to the relatively simple definition of deterministic coverage.

Next, let us look at the stochastic counterpart of the previously described case. That is, instead of deterministic coverage, we take stochastic coverage as utility. The busy fraction remains 0.75. We again see that the Bernoulli-Nash optimum spreads the ambulances more than the utilitarian optimum (Figure 5.10). If we plot the stochastic coverage against the demand (Figure 5.11) we no longer see the clear discretization of values that we observed in the deterministic case.

For our next argument, imagine a utilitarian EMS manager positioning ambulances: that is, he would place them according to the MEXSLP solution. We investigate how much the fairness[6] of his solution can be improved by positioning the ambulances in a different way. This improvement is depicted in Figure 5.12 for different busy fractions. We show both the results for stochastic coverage and survival as utilities. Deterministic coverage is omitted, because the Bernoulli-Nash SWF of the utilitarian solution is often 0, and hence the ratio would become infinitely large. Figure 5.12 shows that for small busy fractions, the ratio is very close to 1, hence a utilitarian manager's choices are actually quite fair. However,

[6] the value of the Bernoulli-Nash SWF



(a) MEXSLP$_{stoch}$ **(b)** MaxFairness$_{stoch}$

Figure 5.10 The optimal solutions for the utilitarian solution and the MaxFairness solution, for $q = 0.75$. The grey nodes are demand points, the black nodes are possible base locations. The numbers represent the number of ambulances placed at each base. The utility is stochastic coverage.

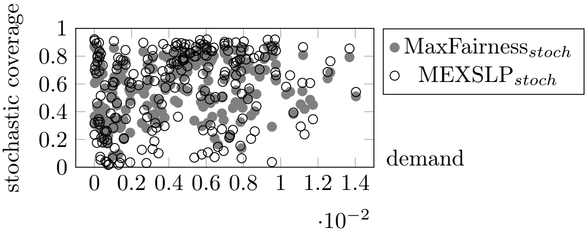


Figure 5.11 The coverage versus demand of each node, using stochastic driving times.

for higher values of q, the improvement factor increases up to 1.4.

As Figure 5.12 shows, the improvement factor is larger for the survival function than for the stochastic coverage. This can be explained by the fact that the survival is more rapidly declining with distance (compare Figure 5.4 to Figure 5.3). This increases the gap between a solution that places many vehicles on one base (as we have seen in Figure 5.8), and a solution that tends to spread vehicles.

We further compare the differences between the utilitarian and Bernoulli-Nash solutions. For now, let us focus on the stochastic coverage, i.e., we compare the MaxFairness$_{stoch}$ solution to the MEXSLP$_{stoch}$ solution. We investigate how each model performs under its own objective, as well as under the objective of the other model. Figure 5.13a shows these values for several values of q. First of all, note how surprisingly similar the values for BN$_{stoch}$ and U$_{stoch}$ are for low values of q. For higher values of q, we see that gap in the BN$_{stoch}$ is slightly bigger than the gap in U$_{stoch}$. This means that if one optimizes for fairness (maximize BN$_{stoch}$), the loss in coverage (U$_{stoch}$) is less than it would be vice versa. Furthermore, as before, we see that the gap between solutions widens as

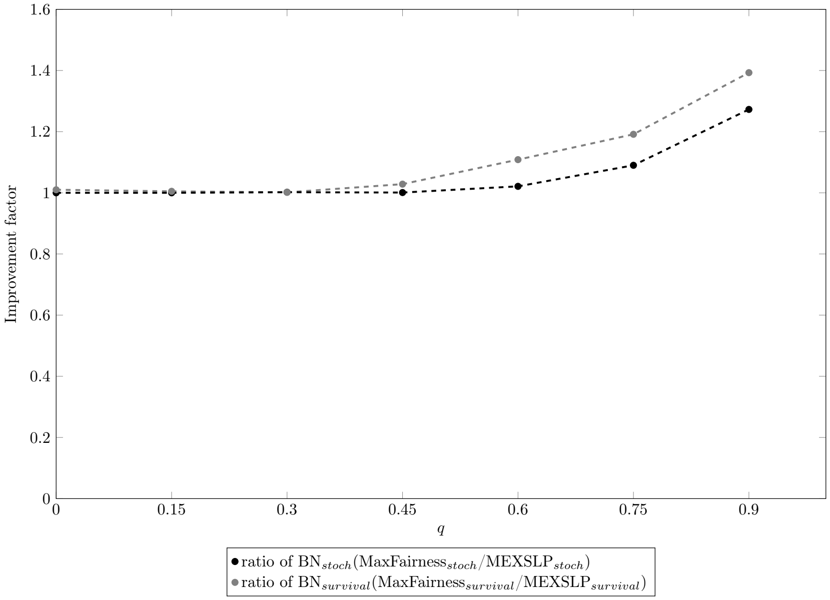
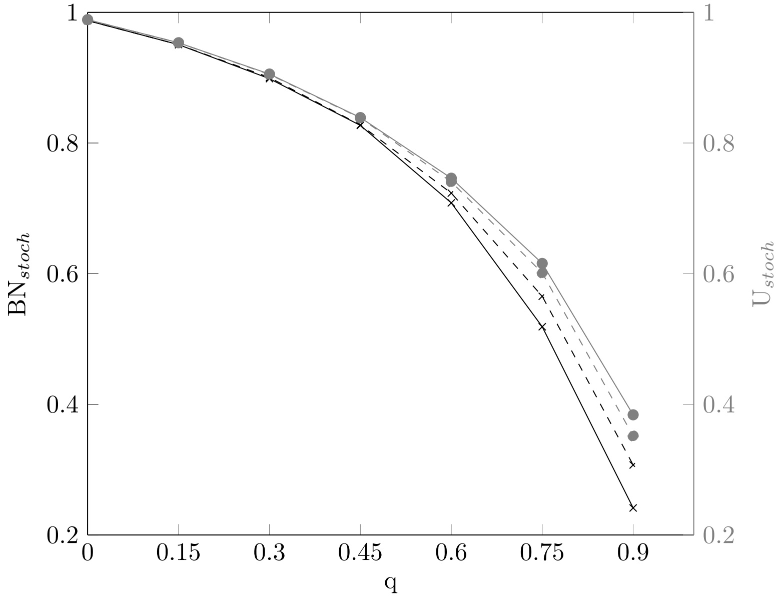
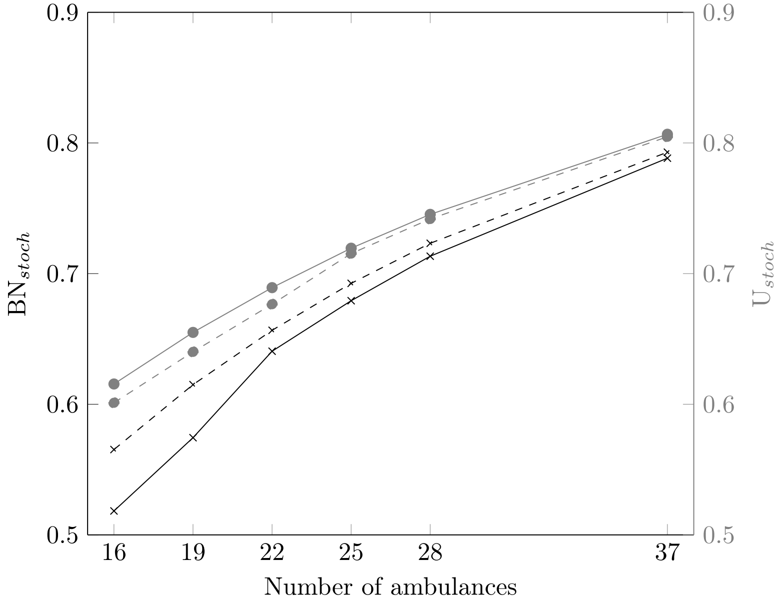


Figure 5.12 The relative improvement in the Bernoulli-Nash SWF, comparing the optimum to the utilitarian solution.



(a) Varying q, for 16 ambulances.



(b) Varying the number of ambulances, while keeping $q = 0.75$.

Figure 5.13 Comparing the objectives BN$_{stoch}$ and U$_{stoch}$, for both optimization models that use stochastic coverage as utility. Dashed lines represent the MaxFairness$_{stoch}$ solution. Solid lines represent the MEXSLP$_{stoch}$ solution.

the load of the system increases.

A quick conclusion might be that a high system load (large q) directly *causes* the gap between solutions that maximize utilitarian and Bernoulli-Nash SWFs. However, recall that throughout these computations we varied q and kept the number of ambulances equal to sixteen. This implies that when we increase q, fewer people can be served. This is not necessarily the case for all problem instances with a large q: one can imagine a very busy EMS region, where vehicles are almost always busy, yet there are so many ambulances that the overall performance is still very high.

To analyze what truly causes the differences between the MaxFairness$_{stoch}$ and the MEXSLP$_{stoch}$ solution - the value of q, or the total number of people that can be served - we performed additional computations. We fixed q at a value for which Figure 5.13a showed differences between the MaxFairness$_{stoch}$ and the MEXSLP$_{stoch}$ solution (we choose $q = 0.75$) and increased the number of ambulances. In Figure 5.13a we observe the range for which the MaxFairness$_{stoch}$ and the MEXSLP$_{stoch}$ solution appear very similar: roughly where $\mathrm{U}_{stoch} \geq 0.8$. Therefore, we start with 16 ambulances and increase this number until $\mathrm{U}_{stoch} \geq 0.8$. The results are depicted in Figure 5.13b. As before, this shows that the social welfare for both solutions are remarkably similar when $\mathrm{U}_{stoch} \approx 0.8$. This indicates that not the load of the system (q), but the utilitarian social welfare (U_{stoch}) determines in which cases the MaxFairness and the MEXSLP differ.

5.7 Discussion

This chapter approaches ambulance location problems from the perspective of social welfare. Our main contribution is that we introduced and implemented two models that maximize the Bernoulli-Nash social welfare. These solutions are juxtaposed against well-known utilitarian solutions. We showed numerical results for a realistic EMS region in the Netherlands. The differences between the utilitarian and Bernoulli-Nash optima turned out to depend on the total number of people that can be served. When at least 80% of the population can be served within the RTT, we found that both solutions are remarkably similar. This is a somewhat surprising, but reassuring result. For a coverage smaller than 80%, the utilitarian and Bernoulli-Nash optima start to show their differences. Generally speaking the Bernoulli-Nash optimum tends to spread vehicles throughout the region, whereas the utilitarian optimum clusters vehicles in areas with high demand. We conclude that as the total coverage of the system decreases, it becomes more important to explicitly think about fairness.

When we translate our result to implications for ambulance providers in practice, the outcome depends on the EMS region. For example, an ambulance provider in the Netherlands typically serves 95% of the most urgent requests within 15 minutes [89]. Our results indicate that, even if they aimed for a utilitarian optimum without thinking about fairness, their solution will be rather fair. For other EMS providers - e.g. in the UK where ambulances typically reach

75% of their most urgent requests within eight minutes [90] - there will be more differences between the most efficient and the most fair solution. In such cases, the political debate about fairness in ambulance care deserves more attention.

The trade-off between efficiency and fairness remains unavoidable. While classical ambulance equity models - that use a egalitarian approach - are not suitable to use in practice because of their poor overall performance, we believe the Bernoulli-Nash optimum provides a reasonable alternative to the utilitarian optimum. This means that, besides the theoretical importance of our work, our solutions could truly be worth considering for ambulance providers who believe fairness should play a role in their decisions. Additionally, the Bernoulli-Nash social welfare of current practice could be evaluated, and used as a measure to detect how far current solutions are from a maximally fair solution.

The models in Section 5.5.2 implicitly define the situation in which there are no ambulances available to have survival probability zero. Alternatively, one might extend these models with a nonzero survival probability in case no ambulance is available. This probability may depend on the demand node i, therefore denote it δ_i. This affects the constraints as follows: one should add $+\ q^{|A|} \cdot \delta_i$ to the right-hand side of (5.8).

In [17] the authors add a decision variable and constraint to limit the total number of base locations used. We chose to leave such a constraint out of this chapter, because it might distract the reader from the main topic. However, it could be added without further complicating the models.

As already described in Section 5.3, using a busy fraction to model ambulance availability is an approximation of the true system dynamics. One might suggest to relax the assumptions of independent vehicles all having and the same busy probability, by using a more advanced model. This is done in Hypercube Queuing Models (HQM) [69] (which are compared to MEXCLP in [9]). However, considering that our most complex model already takes five hours to solve, we chose not to make the problem harder. Other researchers that optimized a form of fairness while using hypercube correction factors faced computational difficulties - even for small cases - and needed to resort to tabu search [31], hence losing a guarantee of finding a globally optimal solution.

6

Improving fairness by time-sharing ambulances

Most papers on the ambulance location problem aim to maximize the total number of inhabitants covered. Such a problem often has one optimum, but there may exist several near-optimal solutions. These have a similar overall performance but differ on a smaller scale, such as individual villages. This raises the question: are we making 'arbitrary' choices in terms of who gets coverage and who does not? In this chapter we propose to share time between several good ambulance configurations in the interest of fairness. We first argue that the Bernoulli-Nash social welfare corresponds to a form of fairness. We formulate a nonlinear optimization model that computes the time shares such that the Bernoulli-Nash social welfare is maximized. Our approach consists of a novel combination of simulation and optimization. We include a case study for a realistically sized ambulance provider in the Netherlands.

This chapter is based on:
C.J. Jagtenberg and A.J. Mason. Improving fairness in ambulance planning by time-sharing. *In preparation.*

6.1 Introduction

An important aspect of EMS is positioning vehicles to provide a high level of service. This is the objective of the ambulance location problem: how and where to locate vehicles in order to effectively cover demand.

An overview of ambulance location models in literature can be found in [22] and [71]. The majority of these models use mixed integer linear programming to maximize the (expected) coverage, where coverage of a region refers to an ambulance being able to arrive at that region within a specified response time threshold.

Researchers tend to compute the 'one and only' optimum to such ambulance location problems, but several near-optimal solutions may exist. These alternative solutions have a similar overall performance, but differ in terms of individual villages or areas. This leads to 'arbitrary' choices in terms of who gets coverage and who does not. If we allow different configurations at different times, we might be able to reach a long-term average performance that is almost equal to the optimum, but more fair.

The ambulance location problem is usually solved to find one permanent solution (a configuration that is to be followed at all times), but there are a few exceptions. These exceptions define some aspect of the problem to vary over time, and compute different solutions for different time periods. Examples include [15, 95]: both are extensions of MEXCLP that incorporate temporal varying demands. Another time-dependent model is introduced in [103], which is an extension of the Double Standard Model. In [103] it is not the demand, but the travel time that varies over time. Note that this is different from our approach: we suggest to switch between different configurations, not because the circumstances change, but in the interest of fairness.

To the best of our knowledge, this is the first work that suggests to share time between several good ambulance configurations in the interest of fairness. Similar to the work in Chapter 5, we define fairness in terms of the Bernoulli-Nash social welfare function (SWF). However, in this chapter we investigate how this can be used in the context of time sharing. This leads to a nonlinear optimization model, which is the main contribution of this chapter. Furthermore, we implemented this model for a realistic EMS region in the Netherlands, and show how the problem can be approached using a novel combination of simulation and optimization.

The rest of this chapter is structured as follows. Section 6.2 describes related work that illustrates the use of Bernoullli-Nash social welfare in time sharing. Section 6.3 shows the Bernoulli-Nash optima for small ambulance location instances - small enough to compute the optima by hand or brute force. Section 6.4 describes the optimization model that allows us to maximize the Bernoulli-Nash social welfare for realistically-sized problem instances. We include a case study in Section 6.5 and finish with our discussion in Section 6.7.

6.2 Preliminaries and related Work

This section defines fairness in terms of social welfare. For completeness, we recap the three different SWFs that were introduced in Chapter 5. Furthermore, this section shows how these functions can be applied in a context of time sharing.

6.2.1 Social welfare

Social welfare is measured as a function of the 'utilities' of individuals or subgroups of a society. For example, a commonly used function is the *utilitarian* SWF. Let u_i be the utility of a person in subgroup i, and let d_i be the number of people in subgroup i. If we let V denote the set of subgroups, the utilitarian social welfare, f_{U} is then given by

$$f_{\mathrm{U}} = \sum_{i \in V} d_i u_i.$$

Solutions that maximize utility do so by maximizing the sum of the individual utilities. This is a common choice for social welfare in ambulance planning: practically all models mentioned in the literature overview in [22, 71] aim to maximize the total (expected) coverage, i.e., they seek a utilitarian solution where utility is defined as coverage.

An alternative SWF is egalitarian, or 'Rawlsian', social welfare. This is defined as the minimum utility over all sub-groups, regardless of the size of the subgroup.

The model we will use is the Bernoulli-Nash SWF. This is defined as the *product* of individual utilities, and consequently is more egalitarian than a utilitarian measure in that it is more sensitive to the utility of the worse off individuals. Using the notation that we introduced above, this can be written as

$$f_{\text{BN}} = \prod_{i \in V} u_i^{d_i}.$$

The Bernoulli-Nash social welfare f_{BN} corresponds to a form of fairness, as we will demonstrate next.

6.2.2 A radio time sharing example

In this section we illustrate why it makes sense to use the Bernoulli-Nash social welfare for time sharing. We do this by recapping an example found in [85, Example 3.6]. Consider a group of n agents working together in a common space, where the radio must be turned on to one of five available stations. The agents have different tastes in music, and it is up to the manager to decide how the time is shared fairly between the five stations. That is, the manager has to decide timeshares λ_i, $i = 1, \ldots, 5$, such that $\lambda_i \geq 0$ and $\lambda_1 + \lambda_2 + \ldots + \lambda_5 = 1$. In this example, an agent can either like or dislike a station, i.e., we set her *utility* for a certain station at 1 or 0.

A Basic Example

In its simplest form, each agent likes exactly one station and dislikes the other four. Let d_i denote the number of fans of station i, with $d_1 + \cdots + d_5 = n$. Note that in this example, the utility of a person that likes station i is equal to λ_i. A utilitarian manager would choose to play the station with the largest support all the time. Note that if there are several such stations, mixing between them is optimal as well. An egalitarian manager, however, would do the opposite: play each station $\frac{1}{5}$th of the time.[1] Note that this ensures everyone is happy 20 percent of the time.

The Bernoulli-Nash social welfare can be viewed as a compromise between the two solutions above. We seek to maximize $f_{\text{BN}} = \prod_i \lambda_i^{d_i}$, which is the same as maximizing $\sum_i d_i \log(\lambda_i)$. If we maximize this under the constraint

[1] assuming each station has at least one fan

$\sum_i \lambda_i = 1$, it leads to a solution of $\lambda_i^* = d_i/n$, i.e., the time share of each station is proportional to the number of its fans.

An elaboration
In a slightly elaborated version of the example above, some agents are flexible, in the sense that they like more than one radio station. The utility matrix is given by:

$$\begin{array}{ccccccc}
 & & & & \textit{Station} & & \\
 & & A & B & C & D & E \\
 & 1 & 1 & 0 & 0 & 0 & 0 \\
 & 2 & 0 & 1 & 0 & 0 & 0 \\
\textit{Agent} & 3 & 0 & 0 & 1 & 1 & 0 \\
 & 4 & 0 & 0 & 0 & 1 & 1 \\
 & 5 & 0 & 0 & 1 & 0 & 1
\end{array}$$

A utilitarian manager would choose to share the time between stations c, d and e, such that there are always as many people as possible listening to a station they like. Unfortunately for Agents 1 and 2, they never get to listen to a station of their choice. An egalitarian manager would select $\lambda_a = \lambda_b = \frac{2}{7}$, $\lambda_c = \lambda_d = \lambda_e = \frac{1}{7}$, such that every agent likes the music $\frac{2}{7}$th of the time.

The Bernoulli-Nash collective utility function again offers a compromise between the two solutions above, as it recommends to play each station $\frac{1}{5}$th of the time. To see this, observe that outcomes a and b are symmetrical, hence will receive the same time share x. Similarly, c, d and e are allocated the same time share y. The Bernoulli-Nash maximization problem can then be written as

$$\text{maximize } f_{\text{BN}} = x^2(2y)^3 \quad \text{s.t. } x, y \geq 0, 2x + 3y = 1,$$

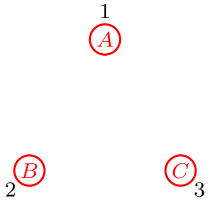
which indeed leads to $x^* = y^* = \frac{1}{5}$.

Next, we show how the concept of Bernoulli-Nash social welfare translates to the ambulance location problem.

6.3 Motivating examples

In this section we consider two small instances of the ambulance location problem. In both instances, the demand is distributed over three locations or *nodes*, labelled A, B and C. Each node i contains a fraction d_i of the total demand. There is one ambulance, which we can position in one of three *bases* (labelled 1, 2 and 3). The decision variables are λ_b, the fraction of time that the ambulance should spend at each base b.

We define the utilities in terms of a response time threshold, where the utility of a person living in a certain demand node is 0 or 1, depending on whether or not there is an ambulance stationed at a base that is closer than the response



(a) Bases and nodes for example 1.

	A	B	C
Base 1	1	0	0
Base 2	0	1	0
Base 3	0	0	1

(b) Coverage of nodes by bases.

Figure 6.1 A set of 3 bases, 1, 2, 3, and demand nodes A, B, C, (left) and their associated coverages (right), for example 1. The circles represent demand nodes; the numbers are the base locations. From each base, exactly one demand location can be reached within the response time threshold.

time threshold. This utility is also known as *single coverage.*

Example 1

Consider the case where each node can be reached by one base, and each base can reach exactly one demand node. An example of this is shown in Figure 6.1. The corresponding utility (i.e., coverage) matrix is given in Figure 6.1b. A utilitarian optimum would be to permanently place the ambulance at the base where it serves the biggest demand. Although this is efficient, it is clearly far from fair.

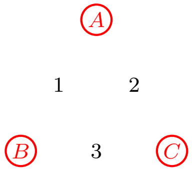
Instead, consider the case where the ambulance can spend some fraction of time λ_i at each base i, $i = 1, 2, 3$. We assume that we can ignore time spent driving between bases, perhaps because the ambulance moves between bases during periods of zero call demand, or perhaps because such moves happen infrequently. Thus, we assume that λ_i gives the probability of the ambulance being at base i when any emergency call arrives. The utility of an individual at some node is now given by the fraction of time that they are covered by an ambulance. The problem of maximizing the Bernoulli-Nash social welfare for this system is given by:

$$\begin{aligned} \max \quad & f_{\text{BN}} = \lambda_1^{d_A} \lambda_2^{d_B} \lambda_3^{d_C} \\ \text{subject to} \quad & \lambda_1 + \lambda_2 + \lambda_3 = 1, \\ & 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1. \end{aligned}$$

Solving this gives a solution $\lambda_1 = d_A, \lambda_2 = d_B$ and $\lambda_3 = d_C$ that maximizes the Bernoulli-Nash social welfare. Just as in the basic case of Section 6.2, the proportion of time spent in each node is proportional to the number of inhabitants served. This is consistent with what we might consider to be a fair distribution. Next, we show what happens if the situation becomes slightly more complex.

Example 2

Consider the case where each node can be reached by two bases, and each base



(a) Bases and nodes for example 2.

	A	B	C
Base 1	1	1	0
Base 2	1	0	1
Base 3	0	1	1

(b) Coverage of nodes by bases.

Figure 6.2 A set of three bases, 1, 2, 3, and demand nodes A, B, C, (left) and their associated coverages (right), for example 2. Each base can reach the two closest demand locations within the response time threshold.

can reach the two closest nodes. An example of this is shown in Figure 6.2.

In this case, our Bernouli-Nash measure is now maximized by solving

$$\begin{aligned} \max \quad & f_{\text{BN}} = (\lambda_1 + \lambda_2)^{d_A} (\lambda_1 + \lambda_3)^{d_B} (\lambda_2 + \lambda_3)^{d_C} \\ \text{subject to} \quad & \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ & 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1 \end{aligned}$$

The solution depends on how the demand is distributed over A, B and C. For the simple case $d_A = d_B = d_C = \frac{1}{3}$ the optimal solution is $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$. This corresponds to a social welfare of $f_{\text{BN}} = \frac{2}{3}^{1/3} \cdot \frac{2}{3}^{1/3} \cdot \frac{2}{3}^{1/3} = \frac{2}{3}$. For comparison, consider a case where one of the nodes has fewer inhabitants than the rest: $d_A = \frac{1}{5}, d_B = d_C = \frac{2}{5}$. The optimal solution in this case is $\lambda_1 = \lambda_2 = 0.2, \lambda_3 = 0.6$, and the corresponding social welfare is $f_{\text{BN}} = 0.4^{1/5} \cdot 0.8^{2/5} \cdot 0.8^{2/5} \approx 0.696$. (Note that this is slightly higher than in the case where the demand is distributed equally over the nodes.) This example shows that the Bernoulli-Nash SWF favours areas with high demand, but does not leave areas with low demand completely uncovered.

6.4 Optimization model

This section formally describes the ambulance system we wish to consider and details the model that optimizes the Bernoulli-Nash social welfare for this system. We assume that demand for ambulances in a region can be modelled using a set of demand zones (or nodes) V. Each node $i \in V$ has a nonnegative demand fraction d_i, such that $\sum_{i \in V} d_i = 1$. There is a fixed number of ambulances available, which may be stationed at a given set of base locations.

We define a *configuration* to be an allocation of ambulances to bases, where we allow a base to hold multiple ambulances. In this problem, we only wish to consider a predefined set C of possible ambulance configurations which we assume the user has constructed.

Our optimisation model requires, as input, the utility u_{ic} of each demand zone $i \in V$ for each configuration $c \in C$. This utility should be a nonnegative number that represents the happiness of an inhabitant of node i under configuration c. Without loss of generality, we assume that the values u_{ic} are normalized, so they lie between 0 and 1.

The utility measure can be defined in whatever manner is suitable for the practical situation at hand and that incorporates the local rules or laws governing response times. The utility is typically some function of the response time. The most common measure might be the expected fraction of incidents in each demand zone that would be served within the response threshold time under some ambulance configuration. Alternative utilities may include, for example, an average response time or a survival probability [17, 54]. In order to determine these utility values, one needs (at a minimum) an estimated driving time for an EMS vehicle with lights and sirens between any base and demand location. Note that travel times for a more comprehensive set of starting locations may be required if we allow a vehicle to be dispatched to a call while still on the road as a result of serving a previous call. Throughout this chapter, we will assume that such estimates are available.

Once an appropriate utility measure has been chosen, the corresponding u_{ic} values for all $i \in V$, $c \in C$, can be computed in a preparatory phase for the set of candidate ambulance configurations generated by the user. This can be done in several ways. For example, one can use one of the many available ambulance location models [22, 71] to estimate utilities at each node for each configuration. Depending on the complexity of the chosen model, this can give a reasonable estimate for u_{ic}. However, we note that these models always include some simplification and cannot completely capture the complex processes in EMS. Alternatively, the values u_{ic} may be estimated by simulation. We consider this a more reliable estimate because it allows explicit modelling of complex issues such as ambulance availability and the fact that ambulances may be dispatched to calls while on the road. Practical limitations such as labor legislation may also be included. Additionally, a trace-driven simulation (using historic call records) would further benefit accuracy because using a trace avoids modelling of the incident arrival process - and the potential corresponding errors.

Once the values u_{ic} have been determined, they can be used as input for the optimization model. The goal is to determine the fraction of time λ_c to spend in each configuration $c \in C$. We allow $\lambda_c = 1$, which indicates that we have chosen just a single configuration for the system. As before, we interpret configuration c's time share, λ_c, as giving the probability of the system being in configuration c when an ambulance is dispatched to a call. Given a solution $(\lambda_1, \lambda_2, ..., \lambda_{|C|})$, the long-term average utility that a person living in node i receives is given by $\sum\limits_{c \in C} \lambda_c u_{ic}$. Therefore, the Bernoulli-Nash social welfare is given by

$$f_{\text{BN}}(\lambda_1, \lambda_2, ..., \lambda_{|C|}) = \prod_{i \in V} \left(\sum_{c \in C} \lambda_c u_{ic} \right)^{d_i}. \tag{6.1}$$

We can now form the following non-linear Bernoulli-Nash Time Sharing

(BNTS) optimization model:

$$\text{BNTS:} \quad \max \quad f_{\text{BN}} = \prod_{i \in V} \left(\sum_{c \in C} \lambda_c u_{ic} \right)^{d_i} \tag{6.2}$$

$$\text{s.t.} \quad \sum_{c \in C} \lambda_c = 1, \tag{6.3}$$

$$0 \leq \lambda_c \leq 1 \quad \forall c \in C \tag{6.4}$$

Before concluding this section, it is useful to consider how f_{BN} should be interpreted. Clearly this interpretation depends on the user's choice of u_{ic} values. One possible natural choice is to let u_{ic} be the probability of a call at node i being served on time (i.e. within the response time target) under configuration c. If we further assume that each individual will make one request for an ambulance, then f_{BN} is the probability that *all* the resulting response times will be within the response time target. By maximizing this probability f_{BN}, we are seeking a system that best delivers for everyone. This is very different to the traditional utilitarian objective which will leave some areas uncovered if this helps more people than it disadvantages.

6.5 Computational results

This section reports the details of a case study in which we compute the Bernoulli-Nash optimal time shares, applying the BNTS optimization model from Section 6.4 to a realistic EMS region. Our numerical work concerns the province of Utrecht, which was described in Section 2.7. The ambulance provider for Utrecht uses nineteen base locations (see Figure 3.2). Utrecht consists of 217 postal codes, see Figure 6.3. The centroids of these regions form our set of demand nodes V. We define the fraction of demand d_i in a single node to be proportional to the number of inhabitants in that postal code.

We consider a fleet of nineteen vehicles that we wish to distribute over the nineteen base locations that exist in this region. In this case study, we only want to consider configurations that have a good overall performance, that is, we want to consider a set of configurations that are near-optimal to the utilitarian ambulance location problem. One way to do this would be to use solutions to one or several integer programming models for the ambulance location problem. However, these solutions can only be considered optimal with respect to the model itself, which is always a simplification of the problem. Instead, we use a combination of simulation and local search to find locally optimal configurations for the utilitarian problem.

We next describe both the simulation model and our local search procedure in more detail.

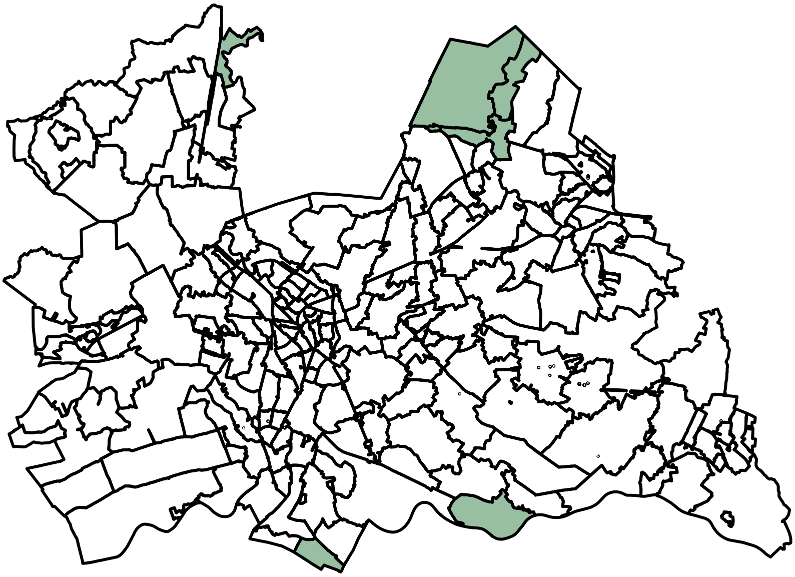


Figure 6.3 Utrecht consists of 217 postal codes; the centroids of these polygons are used to represent demand nodes. The shaded regions cannot be reached from any base within the response time threshold. (As discussed in Section 6.5.1, these are removed from the problem.)

6.5.1 Simulation

We used a discrete event simulation model that was previously described in [55]. The system keeps track of all incidents and vehicles. Vehicle travel occurs on a network that contains nodes for demand nodes (being postal code centroids), hospitals and ambulance bases. The simulation uses deterministic lights-and-siren driving times between nodes as estimated by the RIVM [66, Chapter 3]. Travel speeds without lights and sirens are assumed to be 10% slower.

The simulation generates events for an incident occurring, an ambulance arriving at the scene of the incident, an ambulance leaving for a hospital, an ambulance arriving at a hospital, and an ambulance becoming idle. We drew incident arrival times from a Poisson distribution with an average inter-arrival time of 6.4 minutes, and drew the incident location based on the demand distribution, i.e., an incident occurs at node i with probability d_i, for $i \in V$. Each incident is served by the closest idle ambulance available at that time - including ambulances that are currently on the road returning to base. We approximate the location of those ambulances on the road by using the longitude and latitude of each node, and assume that ambulances travel with constant speed in a straight line between them. Given the estimated travel time between the ambulance's origin and destination, as well as the time that has passed since the vehicle left its origin, we then compute its longitude and latitude and round this to the nearest node in V.

After an ambulance arrives at the scene of an incident, it spends a random amount of time there. This time is drawn from an exponential distribution with an expectation of twelve minutes. After this time, it is decided whether or not the patient needs treatment at a hospital (according to a Bernoulli distribution with probability 0.8). If not, the ambulance becomes available at the scene of the incident. Otherwise, the ambulance drives to the nearest hospital,[2] and spends an additional drop-off time there (drawn from a Weibull distribution with an expectation of 15 minutes). Eventually the ambulance becomes available at the hospital location.

When an ambulance becomes available, we check if there are any unattended incidents left in the queue. If so, the ambulance is immediately dispatched to the first call in the queue. Otherwise, the ambulance stays idle, and is sent to its base location[3].

Using this model, we evaluate any given ambulance configuration by simulating 5,000 hours of EMS events. Recall that in the Netherlands, ambulances should arrive within fifteen minutes. Typically, three minutes are reserved for handling the call, therefore we use a response time threshold of twelve minutes. We keep track of the observed utility u_{ic} of each demand zone $i \in V$ by measuring the fraction of calls there that are reached within 12 minutes.

Using the Bernoulli-Nash SWF raises the theoretical issue that this measure will always have value 0 if there are demand nodes that cannot be reached on time. To avoid this, we slightly adapted our problem instance by removing those demand nodes that are more than 12 minutes away from any base (see Figure 6.3). Another more nuanced approach might be to use a measure, such as survival probability, that recognises there is a non-zero value in an ambulance arriving even if it is outside the response time threshold. We also note that undefined utility values can arise in practice, because our simulation runs may not generate any calls in regions with low population counts. A u_{ic} value of 1 was used to handle such cases.

The purpose of this simulation model is twofold. First of all, it is used to evaluate intermediate solutions in the local search. Second, we use it to determine the utility per demand node for those solutions that we consider good enough to be in C.

6.5.2 Local optima

We want to position nineteen vehicles on the nineteen bases of region Utrecht. As it is possible to position more than one vehicle on a base, there are many different configurations to choose between. We search for solutions with high coverage, i.e., where the total fraction of calls reached within the response time threshold is high. To find these, we implemented a hill climbing algorithm which

[2] We use a set of ten hospitals, excluding private clinics, that existed in the region in 2013.

[3] Note that the ambulance might not arrive at this base location, because it may be dispatched to be new call before reaching its destination.

starts with a random distribution of nineteen vehicles over the nineteen base locations. In each iteration, we change the home base of one of the vehicles. This possible solution is evaluated by simulation, and the solution is accepted if it increases the number of on-time responses. Note that vehicles are assumed to be identical, hence many solutions are equivalent. Therefore, before we evaluate a solution, we check whether an equivalent solution has been simulated before. This reduces the total computation time, and allows a local optimum to be found in approximately eight hours.

We repeat this procedure multiple times, using different starting solutions. This lead to eleven different configurations (each of which has a high utilitarian coverage), which together constitute our set of configurations C. A selection of these local optima is depicted in Appendix 6.A.

6.5.3 Results

This section describes the results of the optimization model (6.2), using the configurations generated using our local search and their corresponding utilities as input. The optimization model was solved using the Ipopt solver [115] (an interior point method solver from COIN-OR) and the model was implemented in Julia/JuMP [72].

Maximizing the Bernoulli-Nash social welfare leads to the following time shares: $\lambda_3 = 0.1984$, $\lambda_8 = 0.1864$, $\lambda_9 = 0.2351$ and $\lambda_{11} = 0.3800$. The other configurations are not used. This Bernoulli-Nash solution has a social welfare $f_{\text{BN}}^{\max} = 0.8525$. To better understand this solution, it is helpful to contrast it with more traditional utilitarian solutions. We do this next.

6.6 Multi-objective optimization model

In this section we investigate how the Bernoulli-Nash social welfare is related to the often-used utilitarian social welfare[4]. This is done using the same set of eleven near-optimal configurations as before. However, we now consider both the Bernoulli-Nash social welfare and the utilitarian social welfare to be of interest, and so we have a multi-objective optimization problem. We term this the Bernoulli-Nash Utilitarian Time Sharing (BNUTS) problem:

$$\text{BNUTS: maximize} \quad \begin{pmatrix} f_{\text{U}} \\ f_{\text{BN}} \end{pmatrix} = \begin{pmatrix} \sum\limits_{i\in V} d_i \sum\limits_{c\in C} \lambda_c u_{ic} \\ \sum\limits_{i\in V} \big(\sum\limits_{c\in C} \lambda_c u_{ic} \big)^{d_i} \end{pmatrix} \tag{6.5}$$

$$\text{subject to} \quad \sum_{c\in C} \lambda_c = 1, \tag{6.6}$$

$$0 \leq \lambda_c \leq 1 \quad \forall c \in C. \tag{6.7}$$

[4] Note that since we defined utilities in terms of a response time threshold, the utilitarian social welfare is equal to the total coverage of the region.

We can use the ε method [37] to compute Pareto-optimal solutions for this BNUTS problem. Formally, we say that a solution $\lambda' = (\lambda'_1, \lambda'_2, ..., \lambda'_{11})$ *dominates* another solution $\lambda = (\lambda_1, \lambda_2, ..., \lambda_{11})$ if either $f_{\text{U}}(\lambda') > f_{\text{U}}(\lambda),\ f_{\text{BN}}(\lambda') \geq f_{\text{BN}}(\lambda)$ or $f_{\text{U}}(\lambda') \geq f_{\text{U}}(\lambda),\ f_{\text{BN}}(\lambda') > f_{\text{BN}}(\lambda)$. We say that a solution λ is *efficient* if there is no other solution that dominates it. We can obtain a discretized set of efficient solutions by solving a sub-problem BNUTS(ε_{BN}) which maximizes f_{U} subject to a lower bound $f_{\text{BN}} \geq \varepsilon_{\text{BN}}$, as follows:[5]

$$\begin{aligned} \text{BNUTS}(\varepsilon_{\text{BN}})\text{: maximize} \quad & \sum_{i \in V} d_i \sum_{c \in C} \lambda_c u_{ic} && (6.8) \\ \text{subject to} \quad & \sum_{i \in V} (\sum_{c \in C} \lambda_c u_{ic})^{d_i} \geq \varepsilon_{\text{BN}}, && (6.9) \\ & \sum_{c \in C} \lambda_c = 1, && (6.10) \\ & 0 \leq \lambda_c \leq 1 \quad \forall c \in C. && (6.11) \end{aligned}$$

We solved BNUTS(ε_{BN}) for $\varepsilon_{\text{BN}} \in \{0.8525, 0.8512, 0.8498, 0.8484, 0.8390\}$, where the first $\varepsilon_{\text{BN}} = 0.8525$ value is the single-objective maximum $f_{\text{BN}}^{\max}$ for the Bernoulli-Nash solution found in Section 6.5.3, and the other values were chosen experimentally to give a good characterisation of the Pareto front. We observe that solving for $\varepsilon_{\text{BN}} = f_{\text{BN}}^{\max}$ gives a lexicographically optimal efficient solution in which f_{BN} achieves its best possible value. We found the other lexicographic solution, being the efficient solution where f_{U} achieves its maximum $f_{\text{U}}^{\max} = 0.9100$, by simply choosing the configuration $c \in C$ giving the largest f_{U} and then observing that only one such configuration existed and thus the solution was efficient.

The efficient solutions we found are summarized in Table 6.1, which shows for example that configuration 8 has the highest total coverage. Furthermore, maximizing the Bernoulli-Nash social welfare f_{BN} requires four different configurations to be combined, but relaxing the bound on the Bernoulli-Nash social welfare f_{BN} reduces this to three or two configurations. These solutions result in the Pareto frontier depicted in Figure 6.4.

We observe that our efficient solutions make use of only a small number of ambulance configurations. Such solutions may appeal to an ambulance organisation as it may be easier to operate a system with fewer configurations. To explore this, we examine the system performance when the two policies that receive the largest time shares in the Bernoulli-Nash optimum are combined. (These are configurations 9 and 11.) To that end, we construct different solutions by taking time share $\lambda_9 \in \{0, 0.1, 0.2, \dots, 1\}$ and $\lambda_{11} = 1 - \lambda_9$. We compute the utilitarian and Bernoulli-Nash social welfare of each solution. These are depicted in Figure 6.5. The solution with $\lambda_9 = 0$ was omitted from the graph because it has a Bernoulli-Nash social welfare of 0. We note that solutions with $\lambda_9 > 0.4$ are

[5] We actually solved an equivalent problem in which (6.9) was modified by taking the logarithms of both sides, and converting the sense to $\leq$.

f_{U}	0.9074	0.9094	0.9098	0.9099	0.9100	0.9100*
f_{BN}	0.8525*	0.8511	0.8498	0.8484	0.8390	0.8337
λ_3	0.1984	0.3459	0.3354	0.1286	0.0008	
λ_8	0.1865	0.4459	0.6646	0.8714	0.9992	1.000
λ_9	0.2351					
λ_{11}	0.3800	0.2082				

Table 6.1 Six solutions that are Pareto efficient. Only non-zero λ_i values are shown. Where an objective function value f_{U} or f_{BN} is marked *, it indicates that this is the maximum possible value for this objective, and thus this column denotes a lexicographic solution. Note that some values differ in decimal point values that are not shown.

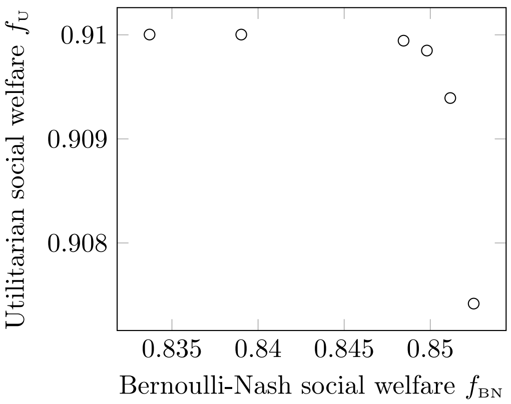


Figure 6.4 Pareto efficient solutions for the region Utrecht, with respect to the utilitarian social welfare f_{U} and Bernoulli-Nash social welfare f_{BN}.

dominated (in a multi-objective sense) by solutions with $\lambda_9 \leq 0.4$, and so there would be no value in operating such solutions.

Figure 6.6 shows the results for a similar analysis in which we consider the three configurations (configurations 3, 9 and 11) used most frequently in the Bernoulli-Nash solution. We consider all combinations for which $\lambda_3, \lambda_9, \lambda_{11} \in \{0, 0.1, 0.2, \ldots, 1\}$ and $\lambda_3 + \lambda_9 + \lambda_{11} = 1$. For each combination, we compute the utilitarian and the Bernoulli-Nash social welfare. These are depicted in Figure 6.6. Note that the utilitarian social welfare f_{U} changes linearly with λ_3, λ_9 and λ_{11}, as it is a convex combination of the associated f_{U} values. The Bernoulli-Nash social welfare has a more complex dependence on λ_3, λ_9 and λ_{11} leading to an apparently convex surface which results in some of the solutions being dominated by other solutions.

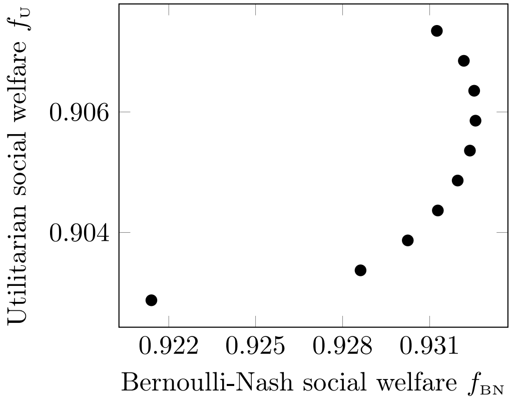
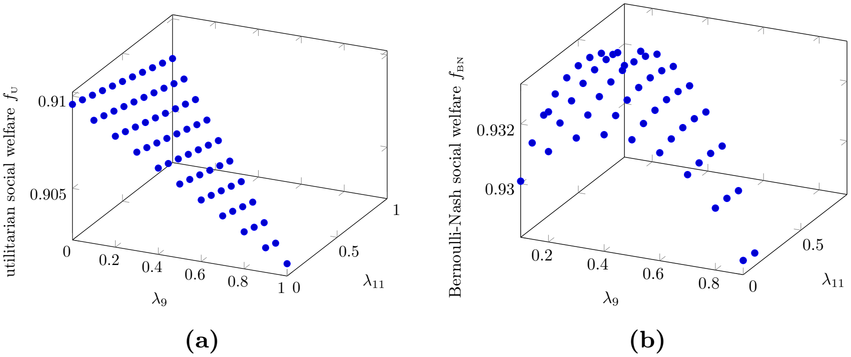


Figure 6.5 Sharing time between policy 9 and policy 11. λ_9 varies between 0.1 (highest point) and 1 (lowest point).



(a) **(b)**

Figure 6.6 The utilitarian and Bernoulli-Nash social welfare for time sharing policy 3, 9 and 11 in different ratios. Solutions with a welfare of 0 were omitted.

6.7 Discussion

In this chapter we viewed the ambulance location problem from a time sharing perspective. We proposed to mix between different ambulance configurations to increase the fairness of the system. The optimal mix is found by computing the maximum of the Bernoulli-Nash social welfare, a nonlinear optimization problem. In our case study, we show a novel approach that combines simulation and optimization. This section discusses alternatives for the choices made in our approach, as well as possibilities for extending this work. There is no single best way to define utilities. In the small examples in Section 6.2, we simply used a 0-1 function indicating if there is an ambulance positioned within reach of the demand node, also known as *single coverage*. Conversely, in our case study in Section 6.5, we defined the performance in terms of a probability of being reached within a response time threshold. Which utility one wants to use may depend on the region and the local rules applicable.

A positive aspect of our problem formulation is that computing the utilities for each demand node and configuration can be done in a preparatory phase, before starting the optimization. This allows for performance indicators that are too hard to compute, to be estimated by simulation instead (as we showed in Section 6.5).

Our problem formulation makes it possible to incorporate results from practice: if an ambulance provider has applied a certain configuration in practice in the past, the performance indicators may be derived from the historical data.

This paper only considers *static* ambulance configurations. That is, in a certain configuration each ambulance has its own *home base*, from which it always responds. Alternatively, one may consider using dynamic ambulance redeployment policies (e.g., [2, 45, 77, 78, 102]). It is also possible to do time sharing in this setting: one can compute the Bernoulli-Nash optimum combining several dynamic policies, or a mix between dynamic and static configurations.

One way to extend our model is to apply robust optimization. This would be a way to handle estimation errors for the values of u_{ic}. (For an introduction to robust optimization, see [14].)

Appendices

6.A Locally optimal configurations

This appendix shows two of the near-optimal ambulance configurations for the region Utrecht found by local search. The configurations are depicted in Figure 6.7 and the corresponding coverage of the demand nodes is shown in Figure 6.8. Note that we plotted the *squared* coverage because it shows the differences between configurations better. Figure 6.8 shows that the Northwest and Southwest corners of the region receive relatively poor coverage in configuration 5, while configuration 9 performs worse in the Northeast.

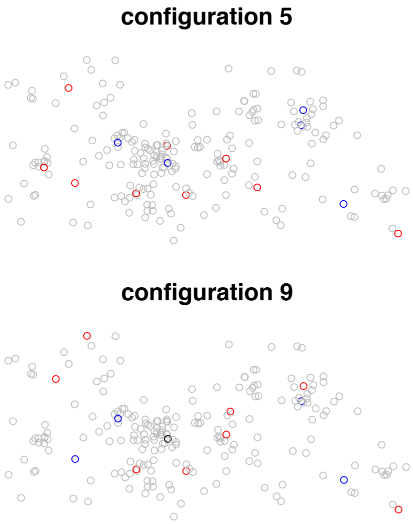


Figure 6.7 A graphical representation of two near-optimal configurations. Each node represents a postal code. Grey: 0 ambulances, Red: 1 ambulance, Blue: 2 ambulances, Black: 3 ambulances

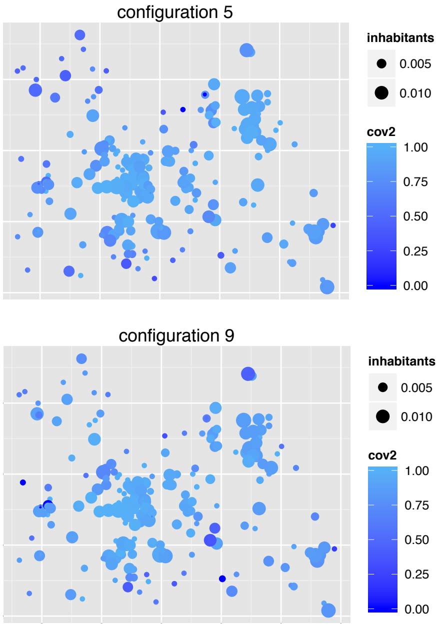


Figure 6.8 Showing the squared fraction of on time arrivals per postal code in Utrecht, as observed for each configuration in a simulation of 5,000 hours. The area of the node scales with the fraction of inhabitants in that node. Nodes that cannot be reached from any base were not depicted.

7

Analysis of Smith's rule in stochastic machine scheduling

In a landmark paper from 1986, Kawaguchi and Kyan show that scheduling jobs according to ratios weight over processing time - also known as Smith's rule - has a tight performance guarantee of $(1+\sqrt{2})/2 \approx 1.207$ for minimizing the weighted sum of completion times in parallel machine scheduling. This chapter proves the counter-intuitive result that the performance guarantee of Smith's rule is not better than 1.243 when processing times are exponentially distributed.

This chapter is based on:
C.J. Jagtenberg, U. Schwiegelshohn and M. Uetz. Analysis of Smith's rule in stochastic machine scheduling. *Operations Research Letters* 41:570–575, 2013.

7.1 Introduction

Minimizing the weighted sum of completion times on m parallel, identical machines is an archetypical problem in the theory of scheduling. In this problem, we are given n jobs which have to be processed non-preemptively on m machines. Each job j comes with a processing time p_j and a weight w_j, and when C_j denotes job j's completion time in a given schedule, the goal is to compute a schedule that minimizes the total weighted completion time $\sum_j w_j C_j$. In the classical 3-field notation for scheduling problems [47], the problem is denoted $\mathrm{P} \mid \mid \sum w_j C_j$. For a single machine, a simple exchange argument shows that scheduling the jobs in order of nonincreasing ratios w_j/p_j gives the optimal schedule [107]. Greedily scheduling the jobs in this order on parallel machines is known as WSPT rule, weighted shortest processing times first, or Smith's rule. On parallel identical machines, WSPT is known to be a $\frac{1}{2}(1+\sqrt{2})$-approximation, and this bound is tight [60]. The computational tractability of the problem was finally settled by showing the existence of a PTAS [106], given that the problem is strongly NP-complete if m is part of the input [42].

In this chapter, we consider the stochastic variant of the problem. It is assumed that the processing time p_j of a job j is not known in advance. It becomes known upon completion of the job. Only the distribution of the corresponding random variable P_j, or at least its expectation $\mathbb{E}[P_j]$, is given beforehand. More specifically, we assume that the processing times of jobs are governed by independent, exponentially distributed random variables. That is to say, each job comes with a parameter $\lambda_j > 0$, and the probability that its processing time exceeds t equals

$$\mathbb{P}\left[P_j > t\right] = e^{-\lambda_j t}\,.$$

We denote this by writing $P_j \sim \exp(\lambda_j)$. Exponentially distributed processing times somehow represent the cream of stochastic scheduling, in particular when juxtaposing stochastic and deterministic scheduling: The exponential distribution is characterized by the memoryless property, that is,

$$\mathbb{P}\left[P_j > s + t \;|P_j > s\right] = \mathbb{P}\left[P_j > t\right].$$

So for any non-finished job it is irrelevant how much processing it has already received. This is obviously a decisive difference to deterministic scheduling models, and puts stochastic scheduling apart. Next to that, the model with exponentially distributed processing times is attractive because it makes the stochastic model analytically tractable.

In the stochastic setting with the objective to minimize $\mathbb{E}[\sum w_j C_j]$, the analogue of Smith's rule is greedily scheduling the jobs in order of non-increasing ratios $w_j/\mathbb{E}\left[P_j\right]$, also called WSEPT (weighted shortest expected processing time first) [92]. For a single machine, this is again optimal [101]. For parallel machines, it has been shown that the WSEPT rule achieves a performance bound of $(2 - 1/m)$ within the class of all non-anticipatory stochastic scheduling policies [84]. Here, the considered metric is the expected performance of WSEPT relative to that of an (unknown) optimal non-anticipatory scheduling policy. We refer to [83] for the precise definition on non-anticipatory stochastic scheduling policies. For the purpose of this chapter, it suffices to know that non-anticipatory stochastic scheduling policies are, at any given time t, only allowed to use information that is available at that time t. Obviously, this is also the case for WSEPT, as the distributions P_j, thus particularly expected processing times $\mathbb{E}\left[P_j\right]$ are even available beforehand.

The major purpose of this chapter is to establish the first lower bound for the $(2-1/m)$ performance guarantee of [84] for exponentially distributed processing times. In fact, we are not aware of any result in this direction. The only result known to us is an instance showing that WSEPT can miss the optimum by a factor 3/2, but then for arbitrary processing time distributions [111, Ex. 3.5.12]. Our main result is the following.

Theorem 1. *When scheduling jobs with exponentially distributed processing times on parallel, identical machines in order to minimize $\mathbb{E}[\sum w_j C_j]$, the performance guarantee of Smith's rule is no better than α with $\alpha > 1.243$.*

To obtain our result, we carefully adapt and analyze the worst-case instance of [60]. Note that the originality of this result lies in the fact that $1.243 >$

$\frac{1}{2}(1+\sqrt{2}) \approx 1.207$. Hence, stochastic scheduling with exponentially distributed processing times has worse worst-case instances than deterministic scheduling. This result may seem counterintuitive, as Pinedo correctly claims the following.

> "It is intuitively acceptable that a deterministic problem may be NP-hard while its counterpart with exponentially distributed processing times allows for a very simple policy to be optimal." [92]

An example for this intuition is given by the problem to minimize the makespan on parallel identical machines: While the problem is NP-hard in deterministic scheduling, the version with exponentially distributed processing times is solved optimally by the LEPT policy (longest expected processing times first) [116]. For the minsum objective considered here, the picture is as follows: For unit weights where $w_j = 1$, the SPT rule is optimal for minimizing $\sum_j C_j$ in the deterministic setting [92], and also SEPT (shortest expected processing time first) is optimal for minimizing $\mathbb{E}[\sum_j C_j]$ when processing times are exponentially distributed [23]. For exponentially distributed processing times and weights that are agreeable in the sense that there exists an ordering such that $w_1 \geq \cdots \geq w_n$ and $w_1\lambda_1 \geq \cdots \geq w_n\lambda_n$, scheduling the jobs in order $1, 2, \ldots, n$ is optimal [57], while the corresponding deterministic problem is NP-hard, and in particular, WSPT is not optimal.

That is to say, there are examples where the stochastic version with exponentially distributed processing times is computationally easier than the deterministic version of the same problem, under the realm of minimizing expected performance. Our result shows that with arbitrary weights, the situation is different. Next to this qualitatively new insight, our analysis also sheds light on phenomena in stochastic scheduling which are interesting on their own.

The chapter is organized as follows. In Section 7.2, we briefly review and visualize the worst-case instance presented in [60]. We explain the intuition behind the stochastified instance of [60] in Section 7.3. Then we derive four technical lemmas about scheduling jobs with exponentially distributed processing times, and finally prove the claimed lower bound for the performance of Smith's rule. We end with a discussion in Section 7.4.

7.2 Recap of the Kawaguchi & Kyan instance

We briefly summarize the instance from [60] that achieves the bound $(1+\sqrt{2})/2$ for deterministic scheduling, as the instance we propose is a stochastic variant thereof. Let n be the number of jobs and m the number of machines. Denote the processing time of job j by p_j and its weight by w_j. The (deterministic) instance is then given by:

$$\begin{aligned}
&m = h + \lfloor (1+\sqrt{2})h \rfloor, && \\
&n = mk + h, && \\
&p_j = w_j = 1/k && \text{for} \quad 1 \leq j \leq mk, \\
&p_j = w_j = 1+\sqrt{2} && \text{for} \quad mk+1 \leq j \leq mk+h\,.
\end{aligned}$$

Here, h denotes an integer, and k is an integer that can be divided by $\lfloor(1+\sqrt{2})h\rfloor$. Notice that $w_j/p_j = 1$ for all jobs j. This means that any list schedule is in fact a WSPT schedule. Let us refer to the first mk jobs as short jobs, and the remaining h jobs as long jobs.

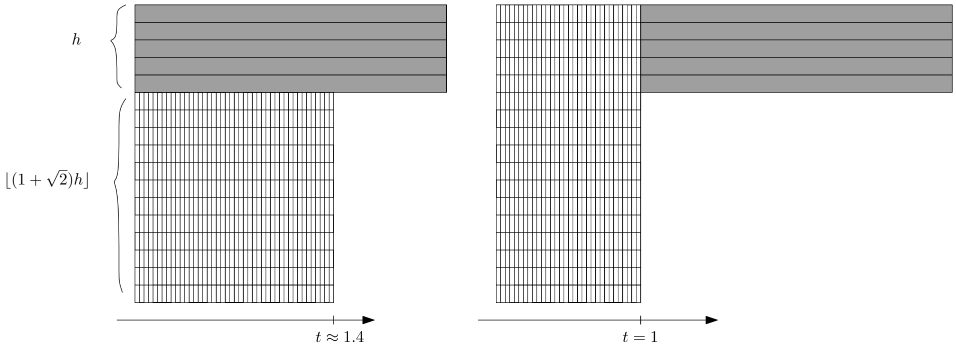


Figure 7.1 Two different WSPT schedules, one with optimal objective value v^* on the left, and one with suboptimal value v on the right, respectively.

Let v^* be the total weighted completion time of a schedule where the long jobs are processed first, and v be the total weighted completion time of a schedule in which all short jobs are processed first. Figure 7.1 depicts these two schedules. The schedule on the left of Figure 7.1 has objective value v^*. Here the last jobs of length $1/k$ finish at time $1 + h/\lfloor(1+\sqrt{2})h\rfloor \approx 1.4$ (for large values of h and k). The schedule on the right of Figure 7.1 has value v, and it finishes the last jobs of length $1/k$ exactly at time 1. In Figure 7.1 we used $h = 5$ and $k = 32$. It can be verified (see [60]) that $v = (1+\sqrt{2})(2+\sqrt{2})h + (m/2)(1+1/k)$ and $v^* = (1+\sqrt{2})^2 h + (m/2)(m/\lfloor(1+\sqrt{2})h\rfloor + 1/k)$. The ratio v/v^* then tends to $(1+\sqrt{2})/2$ as $h \to \infty$ and $k \to \infty$.

7.3 The stochastic Kawaguchi & Kyan instance

We find it particularly instructive to consider the stochastic analogue of the instance presented by Kawaguchi and Kyan [60], even though other instances might lead to comparable results. That said, we keep all parameters the same as in Section 7.2, except that the processing times of long jobs will be $P_j \sim \exp(1/(1+\sqrt{2}))$, and the processing times of short jobs will be $P_j \sim \exp(k)$. So the expected processing times of long and short jobs are identical to the deterministic processing times in the worst case example in [60].

The crucial insight when stochastifying the instance by Kawaguchi and Kyan is the following. The non-optimal schedule with value v is essentially identical to the expected situation in stochastic scheduling. However, we will argue that the optimal schedule with value v^* will have a significantly different expected realization with exponentially distributed processing times. We start by sketching the main differences between the deterministic schedules and the expected stochastic schedules in Section 7.3.1. Then in Section 7.3.2 we derive some technical lemmas about the behaviour of jobs with exponentially distributed processing times, and finish the analysis in Section 7.3.3.

7.3.1 Intuition of the analysis

Suppose we start all h long jobs first and greedily fill up the remaining machines with short jobs. As we will formally prove in Lemma 1, we expect the i^{th} long job to finish at time

$$t_i = \sum\nolimits_{j=1}^{i} \frac{1+\sqrt{2}}{h-j+1}\,.$$

For a given finite number of machines, the schedule will look like depicted in Figure 7.2. The crucial point is that the average expected time that machines

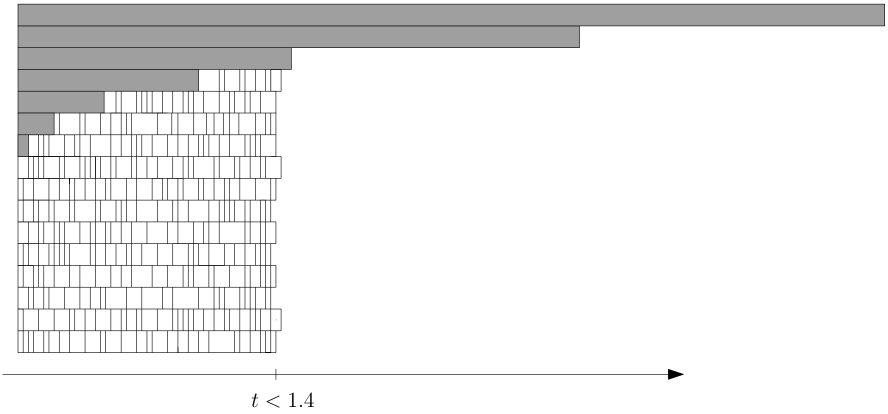


Figure 7.2 Schedule with value v^*: all long jobs start at time 0, yet some of these machines are expected to become available for processing short jobs.

finish processing short jobs will be smaller than in the deterministic case. This happens because many long jobs finish much earlier, and the late finishing of few long jobs does not matter for the short jobs. Hence, the overall contribution of the short jobs will decrease when compared to the deterministic case, while the contribution of long jobs remains exactly the same.

Suppose on the other hand that we first start all short jobs. The set of short jobs is not likely to produce the ideal rectangle as it did in the deterministic case. However, the gap between the time the first machine runs out of short jobs and the time the last machine runs out of short jobs can be made arbitrarily small, by letting k, the inverse of the expected processing time of short jobs, be large. In this situation, the expected cost of the schedule is almost the same as the cost in the deterministic case. This is illustrated in Figure 7.3.

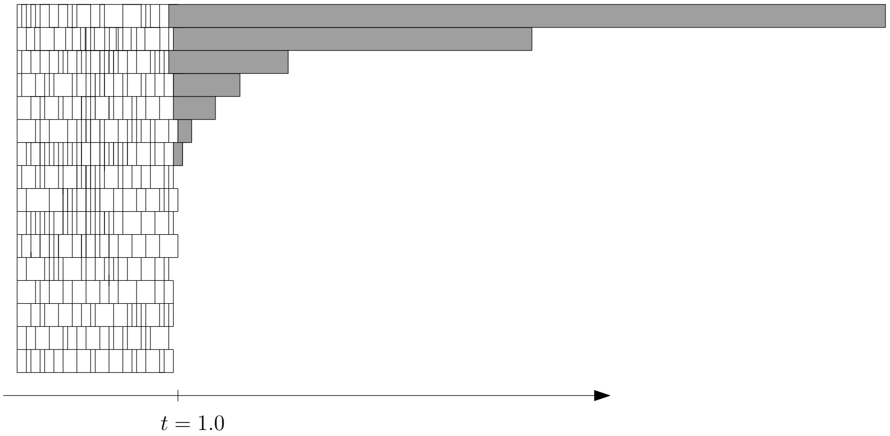


Figure 7.3 Schedule with value v: long jobs scheduled only after short jobs, yet expected to start at almost equal times.

In other words, in the stochastic setting the performance guarantee of WSEPT deteriorates because the expected value for the optimal policy (long jobs first) decreases in comparison to the deterministic case, while the expected value for the suboptimal policy (short jobs first) remains almost the same.

7.3.2 Preliminaries for memoryless jobs

In order to formalize the idea from Section 7.3.1, we first state some technical observations which are needed later in the analysis. Here, λ is an arbitrary positive parameter. We denote by

$$H_n := \sum_{i=1}^{n} \frac{1}{i}$$

the nth harmonic number, where we define $H_0 := 0$. The first lemma gives an estimate on expected job completion times for parallel jobs with $P_j \sim \exp(\lambda)$.

Lemma 1. *When scheduling in parallel $h \leq m$ jobs on m machines with i.i.d. exponential processing times $P_j \sim \exp(\lambda)$, the expected number of machines that are idle at a given time t, denoted $m(t)$, is bounded as follows,*

$$m(t) \geq (m - h) + \lfloor (1 - e^{-\lambda t})h \rfloor \,.$$

Proof. The first completion time is distributed as the minimum of h independent $\exp(\lambda)$ distributions. This is an $\exp(h\lambda)$ distribution, hence it is expected at time $t_1 = \frac{1}{h\lambda}$. After the first job completion, we have $h - 1$ jobs remaining. Since the exponential distribution is memoryless, the next completion is expected a time $\frac{1}{(h-1)\lambda}$ later, so $t_2 = \frac{1}{h\lambda} + \frac{1}{(h-1)\lambda}$. By continuing this argument we find that the

ith job completion is expected at time

$$t_i \;=\; \sum_{j=1}^{i} \frac{1}{(h-j+1)\lambda} \;=\; \frac{1}{\lambda} \sum_{j=h-i+1}^{h} \frac{1}{j} \;=\; \frac{1}{\lambda}\,(\,H_h \;-\; H_{h-i})\,. \tag{7.1}$$

We now use that $H_i \,-\, \ln(i)$ is positive and monotonically decreasing in i [63]. Hence we may conclude that

$$t_i \le \frac{1}{\lambda}\,(\ln(h) \,-\, \ln(h-i)) \;=\; \frac{1}{\lambda}\ln\left(\frac{h}{h-i}\right)\,,$$

which yields

$$i \ge (1-e^{-\lambda t_i})h\,. \tag{7.2}$$

Note that $m(t_i) = (m-h)+i$, for $i = 1,\ldots,h$, by definition. Hence, (7.2) yields

$$m(t_i) \ge (m-h) + (1-e^{-\lambda t_i})h, \tag{7.3}$$

for $i = 1, 2\ldots, h$. Together with the fact that $m(t)$ is integer valued, (7.3) yields

$$m(t) \ge (m-h) + \lfloor(1-e^{-t\lambda})h\,\rfloor$$

for all $t \ge 0$. □

Note that the last job is expected to finish at time $\Theta(\log h)/\lambda$. Nevertheless, the average expected completion time of the jobs is $1/\lambda$; see also Figure 7.2 for an illustration.

Lemma 2. *Let $s \le t$ and consider $k(t-s)$ jobs with i.i.d. processing times $P_j \sim \exp(k)$ and weights $w_j = 1/k$, scheduled on a single machine from time s on. Then for all $\varepsilon > 0$ there exists k large enough so that*

$$\mathbb{E}\left[\textstyle\sum_j w_j C_j\right] \le \int_s^t x\,dx + \varepsilon\,.$$

Proof. Assuming w.l.o.g. that $\frac{1}{k}|(t-s)$, we have expected job completion times at times $s+1/k$, $s+2/k$, ..., $s+k(t-s)/k = t$. We therefore calculate rather straightforwardly that $\mathbb{E}\left[\sum_j w_j C_j\right] = \frac{1}{2}(t^2-s^2) + \frac{1}{2k}(t-s)$, so for $k \ge \frac{t-s}{2\varepsilon}$ the claim is true. □

The next lemma is concerned with the expected total weighted completion time of short jobs that succeed a set of long jobs.

Lemma 3. *Suppose we first schedule h i.i.d. long jobs with processing times $P_j \sim \exp(\lambda)$ on m machines, where $h \leq m$. We then greedily schedule $m\,k$ i.i.d. short jobs, with processing times $P_j \sim \exp(k)$ and weights $w_j = 1/k$, where k is large. Let v_{short} be the expected weighted sum of completion times of the short jobs. Then for k large enough,*

$$v_{short} \leq \int_0^{T'} f(t)\, t\; dt,$$

where $f(t) := (m-h)+(1-e^{-\lambda t})h-1$ and T' is defined so that $\int_0^{T'} f(t)\, dt = m$.

Proof. First, define T as the average expected machine completion time for machines that process short jobs. We know that when scheduling the short jobs greedily, the schedule is expected to look like illustrated in Figure 7.2.

We analyze a scheduling policy π that is inferior to greedy scheduling, that is, it yields an expected value for the total weighted completion times of short jobs $v^{\pi}_{short} \geq v_{short}$. The proof then follows by verifying the claimed upper bound for v^{π}_{short}.

We define π as follows: Let $[i]$ be the ith machine that becomes available to execute short jobs, $t_{[i]}$ be the expected time for that to happen, and for simplicity of notation assume that $i = [i]$. We know that $t_i = 0$ for $i = 1, \ldots, m-h$, and $t_{m-h+i} = \sum_{\ell=0}^{i-1} 1/((h-\ell)\lambda)$ for $i = 1, \ldots, h$. Policy π schedules fixed sets of jobs per machine, in the order in which they become available. More precisely, on machine i, we schedule a fixed set J_i of $k(T - t_i)$ short jobs. By definition of T as the average expected machine completion time for machines that process short jobs, we will have run out of short jobs for all machines i with $t_i > T$. For these machines, we therefore redefine $t_i = T$. Policy π is indeed inferior in contrast to greedy scheduling, as it lacks the load balancing towards the end of the schedule. That is, there is positive probability that a machine is left idle although other machines have yet unscheduled jobs, which cannot happen when scheduling the short jobs greedily. Yet note that, by definition, the expected machine completion times equal T for all machines that process short jobs.

By Lemma 2, we know that under π it holds for the short jobs on machine i that

$$\sum_{j \in J_i} w_j C_j \leq \int_{t_i}^{T} t\, dt + \varepsilon_i\,,$$

for any $\varepsilon_i > 0$. Now we sum over all machines, where we let $\varepsilon_i = 0$ for all machines i that become available while there are no more short jobs. We get

$$v^{\pi}_{short} \leq \sum_{i=1}^{m} \int_{t_i}^{T} t\, dt + \varepsilon_i = \int_0^T m(t) t\, dt + \varepsilon\,, \tag{7.4}$$

where $m(t)$ is defined as the expected number of machines at time t that are available for processing short jobs, and $\varepsilon := \sum_i \varepsilon_i$.

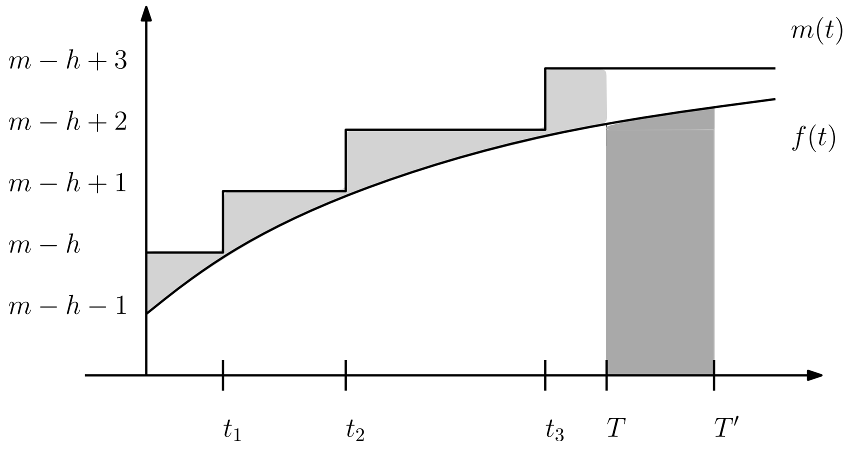


Figure 7.4 Illustration of functions $m(t)$, $f(t)$, and values T and T'.

Now $f(t) = (m-h)+(1-e^{-\lambda t})h-1$, and Lemma 1 yields $m(t) > f(t)$ for all $t \geq 0$. The functions $f(t)$ and $m(t)$ are illustrated in Figure 7.4. By definition of T' we have $m = \int_0^T m(t)\,dt = \int_0^{T'} f(t)\,dt$, which implies that the two grey areas in Figure 7.4 are equal in size. Also note that $m(t) - f(t)$ is nonnegative for all $t \geq 0$. Therefore,

$$\begin{aligned}\int_0^T (m(t) - f(t))t\,dt &< T\int_0^T (m(t) - f(t))\,dt \\ &= T\int_T^{T'} f(t)\,dt \\ &< \int_T^{T'} f(t)t\,dt\,.\end{aligned}$$

Here, the first inequality follows from $m(t) - f(t) \geq 0$, the equality from $\int_0^T m(t)\,dt = \int_0^{T'} f(t)\,dt$, and the last inequality from $f(T) \geq 0$ and f being monotone non-decreasing. We conclude from the previous inequalities that there exists some constant $\eta > 0$ so that

$$\int_0^T m(t)t\,dt \;+\eta \leq \int_0^{T'} f(t)t\,dt\,. \tag{7.5}$$

Therefore, by choosing $\varepsilon \leq \eta$, we may conclude from (7.4) and (7.5), that

$$v^{\pi}_{short} \leq \int_0^T m(t)\,t\,dt + \varepsilon \leq \int_0^{T'} f(t)\,t\,dt\,.$$

□

Intuitively, the expression $\int_0^{T'} f(t)\,t\;dt$ equals the total weighted completion time for infinitesimally short jobs with total expected processing m, scheduled on "machines" with availability $f(t)$. As $m(t) \geq f(t)$, the actual availability of machines for short jobs is higher. We bound the contribution of the jobs that are processed in the light grey area of Figure 7.4 by the contribution they would have if they were processed in the dark grey area.

Finally, the next lemma makes a statement about the machine completion times when scheduling a block of (short) jobs, as illustrated in Figure 7.3.

Lemma 4. *Suppose we schedule $m\,k$ i.i.d. short jobs with processing times $P_j \sim \exp(k)$ greedily on m machines. Then the average expected machine completion time equals 1, and for any $\delta > 0$ there exists k large enough such that the earliest expected machine completion time is at time $t \geq 1 - \delta$.*

Proof. The claim about the average expected machine completion time is clear, because the total expected processing is m. For the second claim, consider the first time, say t, that a machine runs out of jobs. We know from Lemma 1 that the last machine that runs out of jobs is expected to be at time $t + \sum_{i=1}^{m-1} \frac{1}{i\,k}$. For m large enough, we have $\sum_{i=1}^{m-1} \frac{1}{i\,k} \leq \frac{1}{k}\,[\ln(m) + \gamma]$. Here,

$$\gamma := \lim_{i \to \infty} (H_i - \ln i) \approx 0.57721$$

denotes the Euler-Mascheroni constant [41]. Of course, the average expected machine completion time must be less than the last expected machine completion time. Therefore, we have $1 \leq t + \sum_{i=1}^{m-1} \frac{1}{ik} \leq t + \frac{1}{k}[\ln(m) + \gamma]$. If we now let $k \geq (\ln(m) + \gamma)/\delta$, we get $1 \leq t + \delta$. □

7.3.3 Lower bound on performance of Smith's rule

Let v^* denote the expected objective value $\mathbb{E}\left[\sum_j w_j\, C_j\right]$ for the policy that first schedules all long jobs. Similarly, let v denote the expected objective value for the policy that starts long jobs only when there is no short job left to be scheduled. Both policies are WSEPT, hence the ratio v/v^* is a lower bound for the approximation ratio of Smith's rule in stochastic machine scheduling with exponentially distributed processing times. We choose h sufficiently large, and k, a multiple of $\lfloor (1+\sqrt{2})h \rfloor$, we may choose arbitrarily large in comparison to h (i.e., $k >> h$). In fact, we can choose these two parameters in such a way that all our technical lemmas from Section 7.3.2 do apply.

The optimal policy, v^*. We split v^* up into the contribution of long jobs v^*_{long} and the contribution of short jobs v^*_{short}. So

$$v^* = v^*_{long} + v^*_{short}\,.$$

The value v^*_{long}: We start all h long jobs at time 0. Their expected completion time is $1 + \sqrt{2}$ each. Hence the contribution of the long jobs is simply given by

$$v^*_{long} \;=\; h(1+\sqrt{2})^2\,, \tag{7.6}$$

which is the same as in the deterministic case.

The value v^*_{short}: Just like in the proof of Lemma 3 denote by $m(t)$ the expected number of machines at time t that is available for processing short jobs, and T be the average expected machine completion time for machines that process short jobs. We now use Lemma 3 where

$$f(t) \;=\; (m-h) + (1 - e^{-t/(1+\sqrt{2})})h - 1\,.$$

Following the proof of Lemma 3, we need to compute a value $T' \geq T$ large enough so that $\int_0^{T'} f(t)\,dt \geq m$. We have not attempted to solve this analytically, but one can check numerically that for $m = h + \lfloor(1+\sqrt{2})h\rfloor$ and $h \to \infty$,

$$T' = 1.2933 \tag{7.7}$$

suffices to process the short jobs when machine availabilities are governed by function $f(t)$ rather than the true value $m(t)$. Then v^*_{short}, the expected weighted sum of completion times for all mk short jobs, can be bounded using Lemma 3. We thus find, for h and k sufficiently large,

$$v^*_{short} \leq \int_0^{T'} f(t)t\,dt\,. \tag{7.8}$$

With (7.7) and (7.8) we can calculate

$$v^*_{short} \leq\; 2.266h - 0.836\,. \tag{7.9}$$

Combining (7.6) and (7.9) gives

$$v^* \;=\; v^*_{long} \;+\; v^*_{short} \leq (1+\sqrt{2})^2 h + 2.266h - 0.836\,. \tag{7.10}$$

The worst case policy, v. Now we switch to the case where we first schedule all the short jobs. Again split the objective value into the two parts contributed by the short and long jobs, respectively,

$$v = v_{short} + v_{long}\,.$$

The value v_{short}: We have m machines working on mk jobs with processing times $P_j \sim \exp(k)$. According to Lemma 4, on average a machine is expected to finish with these jobs at time 1, and for any $\delta > 0$, we can find k large enough so that no machine is expected to finish before time $1-\delta$. Hence, the average expected completion time of the set of short jobs on each machine is at least $(1-\delta)/2$. Therefore, for any $\varepsilon > 0$, there is k large enough so that, by choosing $\varepsilon = m\delta$,

$$v_{short} \;\geq\; m/2 \;-\; \varepsilon/2\,. \tag{7.11}$$

The value v_{long}: Remember that the schedule is expected to look like depicted in Figure 7.3. Using Lemma 4 again, we know that long jobs are expected to start no earlier than $1-\delta$, for any $\delta > 0$. So by assuming they all start at this time, we get a lower bound for their completion times. If all long jobs start at $1-\delta$, the average expected completion time is $2-\delta+\sqrt{2}$. Multiplying this by the weight and summing over all h long jobs, for any $\varepsilon > 0$ there is k large enough so that

$$v_{long} \geq (2+\sqrt{2})\,(1+\sqrt{2})h \;-\; \varepsilon/2\,, \tag{7.12}$$

by choosing $\delta = \varepsilon/(2h(1+\sqrt{2}))$. With (7.11) and (7.12) we now have

$$v = v_{short} + v_{long} \geq m/2 \;+\; (2+\sqrt{2})\,(1+\sqrt{2})h \;-\; \varepsilon\,. \tag{7.13}$$

The Performance Bound.
Finally, let α be the approximation ratio of Smith's rule for exponentially distributed processing times. Then

$$\alpha \;\geq\; \frac{v}{v^*}\,.$$

Remember that $m = h + \lfloor(1+\sqrt{2})h\rfloor$. Now for carefully chosen $k >> h$, and taking $h \to \infty$, equations (7.10) and (7.13) give

$$\frac{v}{v^*} \geq \frac{m/2 \;+\; (2+\sqrt{2})\,(1+\sqrt{2})h - \varepsilon}{(1+\sqrt{2})^2h + 2.266h - 0.836} \;>\; 1.229\,.$$

So we conclude that $\alpha > 1.229$. Note that this is strictly larger than the approximation ratio for WSPT in the the deterministic case, which is ≈ 1.207.

Optimizing the parameters.
What remains to be done is to optimize over the parameters of the instance to improve the obtained lower bound. To that end, recall that the considered instance has h long jobs and $m = h + \lfloor(1+\sqrt{2})h\rfloor \approx 3.4h$ machines, and long jobs have processing times $P_j \sim \exp(\frac{1}{1+\sqrt{2}}) \approx \exp(0.41)$. However, these parameters are optimized for the deterministic instance. Taking slightly more long jobs, namely by letting $m = 2.3h$, with somewhat shorter processing times, namely $P_j \sim \exp(0.56)$, we obtain a ratio of at least 1.2436, which finally proves Theorem 1.

7.4 Discussion

For minimizing the weighted sum of completion times in parallel machine scheduling, Smith's rule is known to have a tight performance guarantee of $(1+\sqrt{2})/2 \approx 1.207$. This chapter proved the first lower bound for the stochastic version of this problem, when processing times are exponentially distributed. We showed that in this case the performance guarantee of Smith's rule is no better than 1.243. Note that $1.243 > 1.207$, hence, stochastic scheduling with exponentially distributed processing times has worse worst-case instances than deterministic scheduling. This may be considered surprising since there are known examples for which a stochastic scheduling problem with exponentially distributed processing times is computationally easier than the deterministic version of the same problem. We also found instances (not discussed in this chapter) - with comparable building blocks and features - where WSPT is always optimal for the deterministic case, while WSEPT is not necessarily optimal for the stochastic counterpart with exponentially distributed processing times. The numerical calculations in this chapter have been performed using Wolfram Mathematica.

Improvements in the ratio 1.243 might be possible. Yet, the upper bound $(2-1/m)$ seems out of reach. This leaves the question to improve the upper bound on the performance guarantee for WSEPT; in that respect, it is interesting to note that the analysis of [84] does not explicitly exploit the exponential distribution; it is valid in more generality.

Bibliography

[1] L. Aboueljinane, E. Sahin, and Z. Jemai. A review on simulation models applied to emergency medical service operations. *Computers & Industrial Engineering*, 66(4):734–750, 2013.

[2] R. Alanis, A. Ingolfsson, and B. Kolfal. A Markov chain model for an EMS system with repositioning. *Production and Operations Management*, 22(1):216–231, 2013.

[3] R. Aringhieri, G. Carello, and D. Morale. Ambulance location through optimization and simulation: the case of milano urban area. *XXXVIII Annual Conference of the Italian Operations Research Society Optimization and Decision Sciences*, pages 1–29, 2007.

[4] D. Bandara, M.E. Mayorga, and L.A. McLay. Optimal dispatching strategies for emergency vehicles to increase patient survivability. *International Journal of Operational Research*, 15(2):195–214, 2012.

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

[6] T.C. van Barneveld, S. Bhulai, and R.D. van der Mei. A dynamic ambulance management model for rural areas: Computing redeployment actions for relevant performance measures. *Health Care Management Science*, 18:1 – 22, 2015.

[7] T.C. van Barneveld, C.J. Jagtenberg, S. Bhulai, and R.D. van der Mei. Real-time ambulance relocation, assessing real-time redeployment strategies for ambulance relocation. *Under review*, 2015.

[8] T.C. van Barneveld, R.D. van der Mei, and S. Bhulai. Compliance tables for an EMS system with two types of medical response units. *Computers and Operations Research*, 80:68–81, 2017.

[9] R. Batta, J.M. Dolan, and N.N. Krishnamurthy. The maximal expected covering location problem: Revisited. *Transportation Science*, 23(4):227–287, 1989.

[10] BBC. Top Gear. Series 22, Episode 3, 2015.

[11] V. Bélanger, A. Ruiz, and P. Soriano. Recent advances in emergency medical services management. *Technical Report CIRRELT-2015-28*, 2015.

[12] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(4):679–684, 1957.

[13] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[14] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, 2009.

[15] P.L. van den Berg and K.I. Aardal. Time-dependent MEXCLP with start-up and relocation cost. *European Journal of Operational Research*, 242(2):383–389, 2015.

[16] P.L. van den Berg, J.T. van Essen, and E.J. Harderwijk. Comparison of static ambulance location models. *Proceedings of the 3th International IEEE Conference on Logistics Operations Management*, 2016.

[17] P.L. van den Berg, G.J. Kommer, and B. Zuzáková. Linear formulation for the maximum expected coverage location model with fractional coverage. *Operations Research for Health Care*, 8:33–41, 2016.

[18] O. Berman. Repositioning of distinguishable urban service units on networks. *Computers & Operations Research*, 8(2):105–118, 1981.

[19] K. Binmore. *Game Theory and the Social Contract, Volume 1.* The MIT Press, 1994.

[20] R. Bjarnason, P. Tadepalli, and A. Fern. Simulation-based optimization of resource placement and emergency response. *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference, Pasadena*, 2009.

[21] G.E.P. Box. *Empirical Model-Building and Response Surfaces.* Wiley, 1987.

[22] L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. *European Journal of Operational Research*, 147(3):451–463, 2003.

[23] J.L. Bruno, P.J. Downey, and G.N. Frederickson. Sequencing tasks with exponential service times to minimize the expected flowtime or makespan. *Journal of the Association for Computing Machinery*, 28:100–113, 1981.

[24] S. Budge, A. Ingolfsson, and D. Zerom. Empirical analysis of ambulance travel times: The case of Calgary emergency medical services. *Management Science*, 56 (4):716–723, 2010.

[25] M. van Buuren, K. Aardal, R.D. van der Mei, and H. Post. Evaluating dispatch strategies for emergence medical services: TIFAR simulation package. *Proceedings of Winter Simulation Conference 2012, Berlin, Germany*, 2012.

[26] M. van Buuren, G.J. Kommer, R.D. van der Mei, and S. Bhulai. A simulation model for emergency medical services call centers. *Proceedings of Winter Simulation Conference 2015 (WSC 47), Huntington Beach, CA, USA*, 2015.

[27] M. van Buuren, C.J. Jagtenberg, T.C. van Barneveld, R.D. van der Mei, and S. Bhulai. Ambulance dispatch center pilots proactive relocation policies to enhance effectiveness. *Under review*, 2016.

[28] G. Cady. JEMS 200 city survey, JEMS 2001 annual report on EMS operational & clinical trends in large, urban areas. *Journal of Emergency Medical Serices*, 27 (2):46–71, 2002.

[29] G. Carter, J. Chaiken, and E. Ignall. Response areas for two emergency units. *Operations Research*, 20(3):571–594, 1972.

[30] N. Channouf, P. L'Ecuyer, A. Ingolfsson, and A.N. Avramidis. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science*, 10(1):25–45, 2007.

[31] S. Chanta, M.E. Mayorga, M.E. Kurz, and L.A. McLay. The minimum p-envy location problem: a new model for equitable distribution of emergency resources. *IIE Transactions in Healthcare Systems Engineering*, 1(2):101–115, 2011.

[32] S. Chanta, M.E. Mayorga, and L.A. McLay. Improving emergency service in rural areas: a bi-objective covering location model for ems systems. *Annals of Operations Research*, 221(1):133–159, 2011.

[33] D. Chi. *Dynamic Ambulance Redeployment by Optimizing Coverage.* Bachelor's thesis, Delft University of Technology, https://thesis.eur.nl/pub/34157, 2016.

[34] R.L. Church and C.S. Revelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32:101–118, 1974.

[35] J. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.

[36] M.S. Daskin. A maximum expected location model: Formulation, properties and heuristic solution. *Transportation Science*, 7:48–70, 1983.

[37] M. Ehrgott. *Multicriteria Optimization.* Springer, 2005.

[38] E. Erkut, A. Ingolfsson, and G. Erdoğan. Ambulance location for maximum survival. *Naval Research Logistics (NRL)*, 55(1):42–58, 2008.

[39] J.T. van Essen. *Flowing Through Hospitals.* PhD thesis, University of Twente, 2013.

[40] T.J. van Essen, J.L. Hurink, S. Nickel, and M. Reuter. Models for ambulance planning on the strategic and the tactical level. 2013.

[41] L. Euler. De progressionibus harmonicis observationes. *Commentarii academiae scientiarum Petropolitanae*, 7:150–161, 1740.

[42] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, New York, 1979.

[43] Gemeentelijke Gezondheidsdienst Flevoland. `www.ggdflevoland.nl`.

[44] M. Gendreau, G. Laporte, and F. Semet. Solving an ambulance location model by tabu search. *Location Science*, 5(2):75–88, 1997.

[45] M. Gendreau, G. Laporte, and F. Semet. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, 27:1641–1653, 2001.

[46] M. Gendreau, G. Laporte, and F. Semet. The maximal expected relocation problem for emergency vehicles. *Journal of the Operational Research Society*, 57:22 – 28, 2006.

[47] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[48] O. Grandstrand. Modeling equity for allocating public resources. In M.P. Johnson, editor, *Community-Based Operations Research*, chapter 4, pages 97–118. Springer New York, 2011.

[49] Gurobi. `www.gurobi.com`.

[50] M. Hofmeijer. *Dynamic Ambulance Redeployment with uncertain driving times.* Bachelor's thesis, Delft University of Technology, https://thesis.eur.nl/pub/34164, 2016.

[51] K. Hogan and C.S. Revelle. Concepts and applications of backup coverage. *Management Science*, 34:1434–1444, 1986.

[52] A. Ingolfsson. *Operations Research and Health Care Policy*, chapter 6, pages 105–128. Springer New York, 2013.

[53] A. Ingolfsson, E. Erkut, and S. Budge. Simulation of single start station for Edmonton EMS. *Journal of the Operational Research Society*, 54(7):736–746, 2003.

[54] A. Ingolfsson, S. Budge, and E. Erkut. Optimal ambulance location with random delays and travel times. *Health Care Management Science*, 11(3):262–274, 2008.

[55] C.J. Jagtenberg, S. Bhulai, and R.D. van der Mei. An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care*, 4:27–35, 2015.

[56] J.P. Jarvis. Optimal assignments in a Markovian queueing system. *Computers & Operations Research*, 8(1):17–23, 1981.

[57] T. Kämpke. On the optimality of static priority policies in stochastic scheduling on parallel machines. *Journal of Applied Probability*, 24:430–448, 1987.

[58] V. Kapelos. Code Red: An investigation into Edmonton's ambulance system. *Global News*, January 12, 2012.

[59] R.M. Karp. On-line algorthms versus off-line algorithms: How much is it worth to know the future? *International Federation for Information Processing Congress*, 12(1):416–429, 1992.

[60] T. Kawaguchi and S. Kyan. Worst case bound on an LRF schedule for the mean weighted flow-time problem. *SIAM Journal on Computing*, 15:1119–1129, 1986.

[61] R.B.O. Kerkkamp. Facility location models in emergency medical service: Robustness and approximations. Master's thesis, Delft University of Technology, 2014.

[62] V.A. Knight, P.R. Harper, and L. Smith. Ambulance allocation for maximal survival with heterogeneous outcome measures. *Omega*, 40:918–926, 2012.

[63] D. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd ed.) Section 1.2.7: Harmonic Numbers, pp. 75–79.* Addison-Wesley, 1997.

[64] P. Kolesar, W. Walker, and J. Hausner. Determining the relation between fire engine travel times and travel distances in New York City. *Operations Research*, 23(4):614–627, 1975.

[65] G.J. Kommer and S.L.N. Zwakhals. *Referentiekader Spreiding en Beschikbaarheid Ambulancezorg 2008.* RIVM Briefrapport 270192001/2008, 2008.

[66] G.J. Kommer and S.L.N Zwakhals. Modellen referentiekader ambulancezorg 2008 documentatie rijtijden- en capaciteitsmodel. Rapport 270412001/2011, Rijksinstituut voor Volksgezondheid en Milieu (RIVM), 2011.

[67] G.J. Kommer and S.L.N. Zwakhals. *Referentiekader Spreiding en Beschikbaarheid Ambulancezorg 2013.* RIVM Briefrapport 270412003/2013, 2013.

[68] M. Kuhn and K. Johnson. *Applied Predictive Modeling.* Springer New York, 2013.

[69] R.C. Larson. A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, 1(1):67–95, 1974.

[70] L. Lee, T. Lau, and Y. Ho. Explanation of goal softening in ordinal optimization. *IEEE Transactions on Automatic Control*, 44(1):94–99, 1999.

[71] X. Li, Z. Zhao, X. Zhu, and R. Wyatt. Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, 74(3):281–310, 2011.

[72] M. Lubin and I. Dunning. JuMP: A modeling language for mathematical optimization. *INFORMS Journal on Computing*, 27(2):238–248, 2015.

[73] V.J. De Maio, I.G. Stiell, G.A. Wells, and D.W. Spaite. Optimal defibrillation for maximum out-of-hospital cardiac arrest survival rates. *Annals of Emergency Medicine*, 42:242–250, 2003.

[74] M. Manasse, L.A. McGeoch, and D. Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11(2):208–230, 1990.

[75] A.J. Mason. Simulation and real-time optimised relocation for improving ambulance operations. In *Handbook of Healthcare Operations Management*, volume 184, chapter 11, pages 289–317. Springer New York, 2013.

[76] E.S. Matteson, M.W. McLean, E.B. Woodard, and S.G. Henderson. Forecasting emergency medical service call arrival rates. *Annals of Applied Statistics*, 5(2B): 1379–1406, 2011.

[77] M.S. Maxwell, M. Restrepo, S.G. Henderson, and H. Topaloglu. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing*, 22:226–281, 2010.

[78] M.S. Maxwell, S.G. Henderson, and H. Topaloglu. Tuning approximate dynamic programming policies for ambulance redeployment via direct search. *Stochastic Systems*, 3(2):322–361, 2013.

[79] M.S. Maxwell, E.C. Ni, C. Tong, S.R. Hunter, S.G. Henderson, and H. Topaloglu. A bound on the performance of an optimal ambulance redeployment policy. *Operations Research*, 62(5):1014–1027, 2014.

[80] L.A. McLay and M.E. Mayorga. A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions*, 45(1):1–24, 2013.

[81] L.A. McLay and M.E. Mayorga. A dispatching model for server-to-customer systems that balances efficiency and equity. *Manufacturing and Service Operations Management*, 15(2):205–220, 2013.

[82] E. Melachrinoudis, A.B. Ilhan, and H. Min. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34(3):742–759, 2007.

[83] R.H. Möhring, F.J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *ZOR - Zeitschrift für Operations Research*, 28:193–260, 1984.

[84] R.H. Möhring, A.S. Schulz, and M. Uetz. Approximation in stochastic scheduling: The power of LP-based priority policies. *Journal of the Association for Computing Machinery*, 46:924–942, 1999.

[85] H.J. Moulin. *Fair Division and Collective Welfare.* The MIT Press, 2003.

[86] R. Nair and E. Miller-Hooks. Evaluation of relocation strategies for emergency medical service vehicles. *Journal of the Transportation Research Board*, 2137:63 – 73, 2009.

[87] J. Naoum-Sawaya and S. Elhedhli. A stochastic optimization model for real-time ambulance redeployment. *Computers & Operations Research*, 40(8):1972–1978, 2013.

[88] J.F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[89] Ambulancezorg Nederland. *Ambulances In-Zicht 2014*, 2014.

[90] NHS. Ambulance quality indicator. `http://www.ambulancestats.co.uk`.

[91] S.N. Parragh, K.F. Doerner, and R.F. Hartl. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks*, 54(4):227–242, 2009.

[92] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Prentice-Hall, Upper Saddle River (NJ), second edition, 2002.

[93] W.B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* John Wiley & Sons, 2011.

[94] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, New York, NY, USA, 1994.

[95] J.F. Repede and J.J. Bernardo. Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. *European Journal of Operational Research*, 75:567–581, 1994.

[96] M. Restrepo, S. Henderson, and H. Topaloglu. Erlang loss models for the static deployment of ambulances. *Health Care Management Science*, 12(1):67–79, 2009.

[97] C. ReVelle and K. Hogan. The maximum availability location problem. *Transportation Science*, 23(3):192–200, 1989.

[98] Rijksinstituut voor Volksgezondheid en Milieu. `www.rivm.nl`.

[99] U. Ritzinge, J. Puchinge, and R.F. Hartl. Dynamic programming based metaheuristics for the dial-a-ride problem. *Annals of Operations Research*, 236(2): 341–358, 2016.

[100] J. Røislien, P.L. van den Berg, T. Lindner, E. Zakariassen, K. Aardal, and J.T. van Essen. Exploring optimal air ambulance base locations in Norway using advanced mathematical modelling. *Injury Prevention*, http://dx.doi.org/10.1136/injuryprev-2016-041973, 2016.

[101] M.H. Rothkopf. Scheduling with random service times. *Management Science*, 12: 703–713, 1966.

[102] V. Schmid. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, 219:611–621, 2012.

[103] V. Schmid and K.F. Doerner. Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research*, 207(3): 1293–1303, 2010.

[104] Z. Shen, Q.-C. Zhao, and Q.-S. Jia. Quantifying heuristics in the ordinal optimization framework. *Discrete Event Dynamic Systems*, 20(4):441–471, 2010.

[105] P.N. Skandalakis, P. Lainas, O. Zoras, J.E. Skandalakis, and P. Mirilas. "To afford the wounded speedy assistance": Dominique Jean Larrey and Napoleon. *World Journal of Surgery*, 30(8):1392–1399, 2006.

[106] M. Skutella and G.J. Woeginger. A PTAS for minimizing the total weighted completion time on identical parallel machines. *Mathematics of Operations Research*, 25:63–75, 2000.

[107] W.E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.

[108] Stokhos. `www.stokhos.nl`.

[109] C. Swoveland, D. Uyeno, I. Vertinsky, and R. Vickson. Ambulance location: A probabilistic enumeration approach. *Management Science*, 20(4):686–698, 1973.

[110] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19:1363 – 1373, 1971.

[111] M. Uetz. *Algorithms for Deterministic and Stochastic Scheduling.* PhD thesis, Institut für Mathematik, Technische Universität Berlin, Germany, 2002.

[112] Personal communication with ambulance provider Utrecht (RAVU).

[113] T.D. Valenzuela, D.J. Roe, S. Cretin, D.W. Spaite, and M.P. Larsen. Estimating effectiveness of cardiac arrest intervention - a logistic regression survival model. *Circulation*, 96:3308–3313, 1997.

[114] J.L. Vile, J.W. Gillard, P.R. Harper, and V.A. Knight. Predicting ambulance demand using singular spectrum analysis. *Journal of the Operational Research Society*, 63(11):1556–1565, 2012.

[115] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[116] G. Weiss and M. Pinedo. Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability*, 17:187–202, 1980.

[117] D.M. Williams. 2008 JEMS 200 city survey: The future is your choice. *Journal of Emergency Medical Services*, 34(2):36–51, 2009.

[118] Y. Yue, L. Marla, and R. Krishnan. An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *Proceedings of AAAI Conference on Artificial Intelligence*, July 2012.

[119] J. van der Zee. *Dynamic Ambulance Redeployment and Ambulance Dispatching.* Bachelor's thesis, Delft University of Technology, https://thesis.eur.nl/pub/34162, 2016.

[120] O. Zhang. *Simulation Optimisation and Markov Models for Dynamic Ambulance Redeployment.* PhD thesis, University of Auckland, 2012.

[121] O. Zhang, A.J. Mason, and A.B. Philpott. *Simulation and optimisation for ambulance logistics and relocation.* Presented at the INFORMS 2008 Conference, 2008.

[122] L. Zhen, K. Wang, H. Hu, and D. Chang. A simulation optimization framework for ambulance deployment and relocation problems. *Computers & Industrial Engineering*, 72(1):12–23, 2014.

Summary

In emergency situations where every second counts, the timely presence of an ambulance can be a matter of life or death. This importance justifies research to improve the logistics of Emergency Medical Services (EMS). This thesis introduces several models of EMS processes, and displays a variety of applications of Operations Research techniques to ambulance planning problems. Naturally, our research is focused on reducing response times. We deal with various planning stages in the EMS process, aiming to use a given number of resources (e.g., vehicles or personnel) in either a fair or efficient way. Some classical problems are viewed from a new angle, which leads to new theoretical insights as well as practically applicable solutions. The main contribution of this dissertation lies in the models and methods presented, verified by realistic case studies for a Dutch ambulance provider.

The first part of this thesis deals with EMS dispatching: deciding which ambulance to send to which incident. Many researchers and practitioners use the 'closest idle' policy without questioning it, but this is not necessarily optimal: instead, we could choose an ambulance such that remaining idle vehicles are in a good position with respect to expected incidents in the near future. In Chapter 2 we find such alternative dispatch policies using two methods: a MDP-based solution and a heuristic. The heuristic behaves similarly to the policy obtained from our MDP, but is more scalable. We validate both policies by simulating an urban EMS region and show a significant performance improvement when compared to the closest idle method. This sheds new light on the popular belief that the closest idle policy is near-optimal. Although we do not advise all EMS managers to immediately discard the closest idle dispatch method, we do show that the typical argument – that it would not lead to large improvements in the fraction of late arrivals – should be changed.

While the displayed dispatch policies in Chapter 2 clearly outperform the closest idle policy, the optimal policy remains unknown. Therefore, we continue in Chapter 3 by providing a bound on the performance of an optimal dispatch policy. This is done by introducing a benchmark model (referred to as the *offline* dispatch model): deciding which ambulance to dispatch when all incidents are known in advance. We show how to calculate the optimal offline dispatch decisions, and the corresponding performance serves as a bound for any - including the optimal - online policy. We perform a worst case analysis which shows that the so-called competitive ratio of the dispatch problem is unbounded; that is, even an optimal online dispatch algorithm can perform arbitrarily bad compared to the offline solution. However, when we consider the average case, the gap between existing solutions and the offline optimum turns out to be much smaller: a case study

for a large ambulance provider in the Netherlands shows that the closest idle policy obtains a fraction of late arrivals that is approximately 2.7 times that of the optimal offline policy. What is perhaps most surprising is that our dispatch heuristic from Chapter 2 manages to reduce this gap to approximately 1.9, i.e., it closes roughly half of the gap between 'closest idle' and the offline optimum. This work constitutes the first quantification of the gap between online and offline dispatch policies.

Chapter 4 considers dynamic ambulance repositioning: proactively relocating idle vehicles in order to reduce response times. When an ambulance completes service for a patient, we allow it to be sent to one of the existing base locations. In order to compute where to send these idle vehicles, we propose a heuristic that scales to large EMS regions with many vehicles. Simulations show that this method significantly improves the fraction of late arrivals compared to the scenario in which each vehicle always returns to its home base. Furthermore, not only the performance at the response time threshold is improved, but the whole distribution of response times is shifted to the left. As our method is intuitive and easy to implement, it also serves as a suitable base for extensions. The practical relevance of this heuristic was demonstrated by the implementation in a decision support tool used by the EMS region Flevoland, the Netherlands.

Chapters 5 and 6 introduce several models to improve the fairness in ambulance logistics. Rather than simply maximizing the number of people served, we consider the distribution over the different areas where people live. To that end, we view ambulance optimization models from a social welfare perspective. We analyze existing ambulance planning models and show that they tend to maximize either the number of people served (called *utilitarian* social welfare) or maximize the service to the person who is worst off (called *egalitarian* social welfare). We propose a third option: the so-called Bernoulli-Nash social welfare. In Chapter 5, a new facility location model is introduced. This allows us to compute where to open ambulance bases and how to distribute vehicles over those bases, such that the Bernoulli-Nash social welfare is maximized. In several case studies we compare our Bernoulli-Nash optimal solution with the often-used utilitarian optimum. In Chapter 6 we take a different approach: we argue that classical ambulance planning models may have several near-optimal solutions. These have a similar overall performance but differ on a smaller scale, such as individual villages. We propose to avoid the 'arbitrary' choice in terms of who gets coverage and who does not, by sharing time between several good ambulance configurations. We formulate an optimization model that computes the time shares such that, again, the Bernoulli-Nash social welfare is maximized. In this chapter we use a combination of simulation and optimization.

Chapter 7 considers a stochastic machine scheduling problem: the scheduling of jobs for which the processing times are not known in advance. Instead, the processing times are governed by independent exponentially distributed random variables. In particular, we analyze the performance of the Weighted Shortest Expected Processing Times first (WSEPT) rule - also known as *Smith's rule* -

for minimizing the expected weighted sum of completion times. In this setting, WSEPT has a known upper bound of $(2 - 1/m)$, and in this chapter we prove the first lower bound to be 1.243. This result is particularly surprising when juxtaposed with the deterministic counterpart of this problem: there, Smith's rule is known to be a $\frac{1}{2}(1+\sqrt{2})$ – approximation. Note that $1.243 > \frac{1}{2}(1+\sqrt{2}) \approx 1.207$, hence our result indicates that stochastic scheduling with exponentially distributed processing times has worse worst-case instances than deterministic scheduling.

Samenvatting

In noodsituaties waarbij elke seconde telt, kan het tijdig ter plaatse zijn van een ambulance het verschil maken tussen leven en dood. Om die reden is het belangrijk om te onderzoeken hoe de logistiek van medische eerste hulpdiensten verbeterd kan worden. Dit proefschrift introduceert modellen voor verschillende planningsfasen en processen in de ambulancezorg, met als doel de beschikbare middelen zo efficiënt of eerlijk mogelijk te benutten. Verder laten we voorbeelden zien van hoe Operations Research technieken kunnen worden toegepast op ambulancelogistiek, waarbij de focus ligt op het verbeteren van de responstijd. We bekijken klassieke problemen vanuit een nieuwe hoek, wat leidt tot nieuwe theoretische inzichten en praktisch toepasbare oplossingen. De voornaamste bijdrage van dit proefschrift ligt in de modellen en methoden die gepresenteerd worden, geverifieerd door realistische case studies voor een Nederlandse ambulancedienst.

Het eerste deel van dit proefschrift gaat over de uitgifte van ambulances: bepalen welke ambulance naar welk incident gestuurd wordt. Uitgifte is een relatief onderbelicht onderwerp in de literatuur, en veel onderzoekers gebruiken de 'dichtstbijzijnde vrije wagen' zonder hier bij stil te staan. Echter, als we onze doelfunctie definiëren aan de hand van een normtijd waarbinnen een ambulance ter plaatse moet zijn, kan er een betere prestatie worden geboekt wanneer we een andere wagen dan de dichtstbijzijnde sturen: we kunnen er dan een kiezen zodanig dat de overblijvende vrije wagens in een goede positie staan ten opzichte van verwachte incidenten in de toekomst. In Hoofdstuk 2 geven we twee manieren om zo een alternatief uitgiftebeleid te vinden: een Markov beslissingsprobleem en een heuristiek. De heuristiek gedraagt zich vergelijkbaar met de oplossing van het Markov beslissingsprobleem, maar schaalt beter. We valideren deze twee oplossingen door een dichtbevolkte ambulanceregio te simuleren en laten zien dat er een behoorlijke prestatiewinst kan worden gerealiseerd vergeleken met het sturen van de dichtstbijzijnde vrije wagen. Dit schijnt nieuw licht op de algemeen aangenomen stelling dat de dichtstbijzijnde vrije wagen een (bijna) optimale keuze is. Hoewel we ambulancediensten niet adviseren om onmiddellijk ons uitgiftebeleid in de praktijk over te nemen, laten we wel zien dat het veelgebruikte argument - dat het niet zou leiden tot een grote verbetering in de fractie incidenten waarbij men te laat komt - veranderd moet worden.

Hoewel de uitgifte regels die we in Hoofdstuk 2 afleiden duidelijk beter presteren dan de 'dichtstbijzijnde vrije wagen' regel, blijft het optimum onbekend. Daarom geven we in Hoofdstuk 3 een grens voor de prestatie van een optimaal uitgiftebeleid. Dit doen we door een alternatief model te introduceren (het *offline* model): bepalen welke ambulance gestuurd moet worden als alle incidenten van tevoren bekend zijn. We laten zien hoe we onder deze omstandigheden de optimale uitgifte keuzes kunnen maken, en de bijbehorende prestatie geldt als een grens voor elk - inclusief het optimale - online uitgiftebeleid. We analyseren het slechtst mogelijke geval, en laten zo zien dat de competitieve verhouding van het uitgifteprobleem onbegrensd is; dat wil zeggen, zelfs een optimaal online uitgiftebeleid kan het willekeurig slecht doen in vergelijking met de offline oplossing. Echter, als we een gemiddeld geval beschouwen, blijkt het gat tussen bestaande uitgifteregels en het offline optimum veel kleiner te zijn: een studie voor een grote Nederlandse ambulancedienst laat zien dat het sturen van de dichtstbijzijnde vrije wagen ertoe leidt dat men 2.7 keer vaker te laat komt dan in het optimale offline geval. Wat misschien nog het meest verrassend is, is dat onze heuristiek uit Hoofdstuk 2 het gat tussen de ‘dichtstbijzijnde vrije’ en het offline optimum grofweg halveert tot een factor 1.9. Dit is de eerste kwantificering van het gat tussen online en offline uitgifteregels.

Hoofdstuk 4 beschouwt dynamisch ambulance management: het proactief verplaatsen van vrije wagens met als doel een betere dekking van de regio te realiseren. Wanneer een ambulance vrijkomt, staan we toe dat deze naar een van de beschikbare standplaatsen wordt verplaatst. Om uit te rekenen waar deze vrije wagens het beste heen kunnen, stellen we een heuristiek voor die schaalt tot grote ambulancedienst met veel wagens. Simulatie laat zien dat onze methode de fractie te laat gearriveerde ambulances significant vermindert ten opzichte van een scenario waarin elke wagen altijd terugrijdt naar zijn *eigen* standplaats. Verder blijkt dat niet alleen de prestatie op de normtijd verbetert, maar dat de hele verdeling van responstijden naar links opschuift. Doordat onze methode intuïtief en gemakkelijk te implementeren is, is deze ook geschikt als basis voor uitbreidingen. De praktische relevantie van deze heuristiek blijkt uit de implementatie in software gebruikt door GGD Flevoland.

Hoofdstukken 5 and 6 introduceren verschillende modellen die tot doel hebben de eerlijkheid in ambulancelogistiek te verbeteren. In plaats van simpelweg het aantal mensen dat op tijd geholpen kan worden te maximaliseren, bekijken we ook de verdeling over de verschillende gebieden waar mensen wonen. Daartoe beschouwen we ambulance optimalisatiemodellen vanuit het perspectief van sociale welvaart. We analyseren bestaande ambulance planningsmodellen en laten zien dat deze typisch ofwel de utilistische, ofwel de maximin sociale welvaart optimaliseren. Wij stellen een derde optie voor: de zogenaamde Bernoulli-Nash sociale welvaart. In Hoofdstuk 5 wordt een nieuw locatiemodel geïntroduceerd. Dit maakt het mogelijk om te berekenen waar standplaatsen geopend moeten worden zodanig dat de Bernoulli-Nash sociale welvaart maximaal is. In verschillende casussen vergelijken we de Bernoulli-Nash optimale oplossing met het veelgebruikte utilistische optimum. In Hoofdstuk 6 gebruiken we een andere aanpak: we redeneren dat klassieke ambulance planningsmodellen vaak veel bijna-optimale oplossingen hebben. Deze presteren gemiddeld genomen vergelijkbaar, maar verschillen worden op kleinere schaal zichtbaar, zoals individuele dorpen of wijken. We stellen voor om de schijnbaar willekeurige keuze ‘wie wordt er gedekt en wie niet’ te vermijden door op verschillende tijden ambulances in een andere (goede) configuratie op te stellen. We formuleren een optimalisatiemodel dat de tijdsverhouding tussen de verschillende configuraties bepaalt, zodat (wederom) de Bernoulli-Nash sociale welvaart maximaal is. In dit hoofdstuk gebruiken we een combinatie van simulatie en optimalisatie.

Hoofdstuk 7 gaat over een stochastisch machine roosterprobleem: het roosteren van taken waarvoor de duur niet van tevoren bekend is. In plaats daarvan wordt de duur van taken bepaald door onafhankelijke, exponentieel verdeelde toevalsvariabelen. In het bijzonder analyseren we de prestatie van de Gewogen Kortste Verwachte Duur Eerst (GKVDE) regel - ook bekend als *Smith's regel* - voor het minimaliseren van de gewogen som van tijden waarop taken voltooien. In deze setting is bekend dat GKVDE een bovengrens heeft van (2-1/m), en in dit hoofdstuk bewijzen we de eerste ondergrens: 1.243. Dit resultaat is verrassend wanneer we het afzetten tegen de deterministische variant van het probleem: daarvan is bekend dat Smith's regel een $\frac{1}{2}(1+\sqrt{2})$ – approximatie is. Merk op dat $1.243 > \frac{1}{2}(1+\sqrt{2}) \approx 1.207$, ons resultaat duidt er dus op dat stochastisch roosteren met exponentieel verdeelde taakduren slechtere worst-case instanties heeft dan deterministisch roosteren.

About the author

Caroline Jagtenberg was born in Grubbenvorst, the Netherlands, in 1987. She started her studies in 2005, combining two Bachelor's programs: Mathematics and Physics & Astronomy at Utrecht University. She continued with a Master's degree in Mathematical Sciences, also at Utrecht University, including a semester abroad at Lund University, Sweden. After finishing her Master's program cum laude, she spent another semester at Monash University, Australia, studying various topics in computer Science.

In 2011, she started working as a software engineer at ORTEC in Gouda. There, she developed the back end of a routing application, and later switched focus to workforce planning. During the time of this research, Caroline remained employed at ORTEC for two days a week.

In 2012, Caroline started her PhD at the Stochastics group at CWI (the Dutch national research institute for mathematics and computer science) in Amsterdam. Her research was part of a larger research project called *REPRO*: from Reactive to Proactive planning of ambulance services. In the last year of her PhD program, Caroline spent four months at the Engineering Science department of the University of Auckland in New Zealand.

List of publications

Peer reviewed publications

C.J. Jagtenberg, U. Schwiegelshohn and M. Uetz. Analysis of Smith's rule in stochastic machine scheduling. *Operations Research Letters* 41:570–575, 2013.

C.J. Jagtenberg, S. Bhulai and R.D. van der Mei. An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care* 4:27–35, 2015.

C.J. Jagtenberg, S. Bhulai and R.D. van der Mei. Dynamic ambulance dispatching: is the closest-idle policy always optimal? To appear in *Health Care Management Science.*

C.J. Jagtenberg, S. Bhulai and R.D. van der Mei. Optimal ambulance dispatching. To appear in N.M. van Dijk & R.J. Boucherie (Eds.) *Markov Decision Processes in Practice*, chapter 8, pages 259–257. Springer.

C.J. Jagtenberg, P.L. van den Berg and R.D. van der Mei. Benchmarking online dispatch algorithms for Emergency Medical Services. To appear in *European Journal of Operational Research.*

T.C. van Barneveld, C.J. Jagtenberg, S. Bhulai and R.D. van der Mei. Real-time ambulance relocation. Assessing real-time redeployment strategies for ambulance relocation. *Submitted.*

M. van Buuren, C.J. Jagtenberg, T.C. van Barneveld, R.D. van der Mei and S. Bhulai. Ambulance dispatch center pilots proactive relocation policies to enhance effectiveness. *Submitted.*

C.J. Jagtenberg, A.J. Mason and O.M. Dowson. Fairness in the ambulance location problem: maximizing the Bernoulli-Nash social welfare. *In preparation.*

C.J. Jagtenberg and A.J. Mason. Improving fairness in ambulance planning by time-sharing. *In preparation.*

Professional publications

C.J. Jagtenberg and R.D. van der Mei. Mensenlevens redden met Operations Research: dynamisch ambulance management. *Stator*, 2015.

K. Aardal, T.C. van Barneveld, P.L. van den Berg, S. Bhulai, M. van Buuren, J.T. van Essen, C.J. Jagtenberg, G.J. Kommer, G.A.G. Legemaate and R.D. van der Mei. Van reactieve naar proactieve planning van ambulancediensten. *Nieuw Archief voor Wiskunde*, 2015.