Dynamic Ambulance Redeployment by Optimizing Coverage

Bachelor Thesis
Econometrics & Operations Research
Major Quantitative Logistics

Author: Dave Chi
Supervisor: Rutger Kerkkamp

Erasmus School of Economics
Erasmus University Rotterdam

June 30, 2016

ERASMUS UNIVERSITEIT ROTTERDAM

Abstract

The task of emergency medical services (EMS) is to timely provide service to those in urgent need of medical attention. In order to achieve this, ambulances must arrive at the emergency location within a time threshold imposed by the government. To facilitate this, sufficient base locations containing ambulances are spread across regions. An important concept is the notion of 'coverage'. Because each region must be sufficiently covered at any time, ambulances are constantly redeployed to different base locations. To solve the relocation problem of ambulances, a model called the MEXCLP is implemented that optimizes the total coverage. Two policies are to be distinguished: applying the static policy necessitates that idle ambulances always return to a predefined base location after finishing service and the dynamic policy works in conjunction with the dynamic MEXCLP to determine the location that optimizes the total coverage. By simulation, the performances of the EMS policies are measured. From the results, the dynamic policy always outperforms the static policy under any circumstance. Nevertheless, the static policy is useful as a benchmark.

This thesis provides a solution to the dynamic ambulance redeployment problem using methods from a related research. Furthermore, it provides analysis on performance levels for various research questions. The key performance indicator (KPI) is the expected fraction of late arrivals, which is to be minimized. Another relevant number is the expected average response time, which correlates with the expected fraction of late arrivals. This research shows that the fraction of late arrivals can be as low as four percent.

Contents

1 Introduction **1**
1.1 Problem description . . . 1
1.2 Related work . . . 2
1.3 Research questions . . . 3

2 Methods **4**
2.1 MEXCLP . . . 4
2.2 DMEXCLP . . . 6
2.3 Discrete Event Simulation . . . 6

3 Data **9**

4 Results and analysis **10**
4.1 Performance by changing the accident rate . . . 12
4.2 Sensitivity to the busy fraction . . . 13
4.3 Reallocation of base locations . . . 15
4.4 Different response time standards . . . 16
4.5 Non-constant accident rate . . . 17

5 Conclusion **20**

A Locations overview **I**

B Notation overview **II**

C Simulation algorithms **III**

D Optimized base locations **VII**

E Results for different time standards **VIII**

F Algorithms for non-constant accident rate **X**

References **XIII**

1 Introduction

1.1 Problem description

At any moment, emergencies arise when people get health problems, injuries or any other medical condition. It is important that aid comes sooner rather than later to help the people, who are in urgent need of medical attention. Ambulances must arrive at the location of the emergency as soon as possible. However, it is no easy task for emergency medical services (EMS) managers, who coordinate the ambulance system, to ensure that any inhabitant within a region can be reached and, most importantly, will be reached as soon as possible by an ambulance, given that they have to deal with limited resources and budget.

A related issue is that one needs to make sure that at any point in time each location in a region is sufficiently covered by ambulances, given that a region has ambulance bases at fixed locations and given a total number of ambulances. First, the concept of 'coverage' will be explained. When one speaks of coverage, one means that a location is surrounded by a sufficient number of ambulances in the neighbourhood, such that it can easily be reached by an ambulance within a specific time threshold. It is ideal to have this for each demand location. We want to avoid a situation in which no ambulance is able to respond to an accident[1] within the time threshold. It is also necessary that the number of ambulances stationed at a base location is sufficient, as an ambulance from a base location may be busy while another accident occurs at a demand location covered by the same base location. Furthermore, relocating ambulances to different base locations after finishing service may improve the total coverage, ensuring that accidents are more likely to be reached in time. This process is called the dynamic ambulance redeployment.

Accidents are unpredictable, since they occur randomly at an unknown rate. Moreover, the rate is dependent on time and location. It is more likely that an accident occurs in cities or during peak hours than at the country side or at night.

In this thesis, the studied region is the RAV of Utrecht, one of the largest ambulance providers of the Netherlands. The RAV is the regional ambulance service in the Netherlands. Given the base locations and the number of ambulances, the dynamic ambulance redeployment problem will be solved. The research objective is to minimize the expected fraction of late ambulance arrivals by optimizing the coverage for each demand location. The reason for

[1] In this thesis, accidents will be used to refer to demand for ambulances.

this objective is because the aim of EMS managers is to provide good service to citizens, and this will be attained when ambulances arrive in time when they are needed. This, in turn, is achieved by having sufficient coverage in each demand location by ambulances.

1.2 Related work

Many insights on this topic can be gained from other papers. The reference framework [1] of the RIVM, the Dutch National Institute for Public Health and the Environment, describes methods that have been used to improve the base locations and the number of ambulances for every RAV in the Netherlands. Data acquired from the reference framework (e.g. the travel times between two demand locations) will be used for the research in this thesis.

Furthermore, the heuristic described in Jagtenberg et al. [2] will be used to solve the dynamic ambulance redeployment problem, for which implementation of both an integer linear programming problem and a simulation is required. Only the most urgent emergencies, called the A1 emergencies in the Netherlands, are discussed in the thesis. For these emergencies, ambulances must arrive at the demand location within 15 minutes after the call has been made. The base locations, number of ambulances and travel times between every demand location are known in advance.

Two different policies are discussed in Jagtenberg et al. [2]. An ambulance that has served an accident either must return to the home base it has been assigned to, which is the rule applied under a static policy, or may relocate to another base if it gives a coverage improvement, which is the rule under a dynamic policy. In case of a dynamic policy, an assumption is made that repositioning of an ambulance is only allowed after an ambulance has served an accident. This is a reasonable assumption, as it would be very inefficient for ambulance drivers to relocate from base to base in any occasion and it would cost a lot of fuel. The restriction ensures that the number of trips remains the same, which also convinces the EMS managers that the dynamic policy is a good alternative to the static policy. The goal is to minimize the expected fraction of late arrivals, which will be referred to as the key performance indicator (KPI). The static policy results will be used as benchmark.

The integer linear programming problem that needs to be solved is the maximum expected covering location problem (MEXCLP) [3]. For this model, the demand locations and the base locations are required. A variation of this model [2] is applied to the dynamic policy, involving relocation of an ambulance to a different base location by looking for the best coverage improvement.

Furthermore, an overview of statistics [4] of the health care department in the Netherlands provides insight on the performance of the Dutch medical service. In this overview, statistical data can be found, such as the number of A1 accidents, the number of accidents that required a transport and the number of ambulances. The information will be used as data for this research.

1.3 Research questions

In this thesis, mainly the methods described in Jagtenberg et al. [2] are implemented to obtain the research objective and to analyze the results. The methods are easy to implement and very lenient to adjustments for some further analysis.

The MEXCLP requires a possible set of base locations. Hence, to solve our problem we use our current base locations in the RAV of Utrecht. An interesting extension to research is to see the consequence of relaxing our set of base locations by including all demand locations in the set of possible base locations. Our assumption is that every location in the RAV of Utrecht can potentially serve as a base location. So, we use this input to solve the MEXCLP, after which we can derive a new set of base locations. For the results of this analysis, we refer to Section 4.3.

As already indicated, ambulances need to arrive at accident locations within a certain time threshold. For this research, we use the time threshold that is applied by the Dutch authority to determine the KPI. As mentioned before, the time threshold for the A1 emergencies in the Netherlands is 15 minutes. This, however, is not a global standard. Some other countries use a much stricter time threshold, such as 10 or 12 minutes. Therefore, Section 4.4 discusses the effects on the performance by applying different time thresholds as a Dutch standard requirement.

Throughout the day, the rate at which accidents occur is not constant, since more accidents happen at certain hours. Our initial simulation model assumes the accident rate to be constant, which is far from true. Therefore, we modify the simulation model to incorporate the changes in the accident rate throughout the day. We assume accidents occur according to a piecewise constant non-homogeneous Poisson process. The results are presented in Section 4.5.

The contents of this thesis are structured as follows. In Section 2, the methods and the corresponding notations are introduced. This section explains the MEXCLP and our simulation model. Section 3 features the data we use. The results to our methods, the analysis and the extensions are presented in Section 4. Lastly, in Section 5, we end with the conclusion.

2 Methods

Before we explain the methods for solving the dynamic ambulance redeployment problem, some notations are introduced to formulate our model.

First, let V be the set of demand locations in the RAV of Utrecht. A demand location corresponds to a four digits postal code[2], meaning that in order to determine the position, we take the average of the coordinates over all six digits postal codes locations of which the first four digits are the same. Each demand location $i \in V$ has its population d_i and one may assume that d_i corresponds to the demand of the location. Let the demand fraction of location i be $g_i = \frac{d_i}{\sum_i d_i}$, for $i \in V$. Furthermore, accidents in the region comprised of every location in V follow a Poisson process with a rate λ. Thus, accidents at location i occur at rate $g_i\lambda$.

Let $W \subseteq V$ denote the set of base locations, which is a subset of the set of demand locations, since base locations are located at specific four digits postal code locations. Moreover, let $H \subseteq V$ be the set of hospitals, which is a subset of the set of demand locations for the same reason as for the base locations. Furthermore, A is the set of ambulances and τ_{ij} are the travel times between locations $i \in V$ and $j \in V$.

The complete overview of demand locations, base locations and hospital locations in the RAV of Utrecht can be viewed in Appendix A.

As stated before, our objective is to minimize the expected fraction of late arrivals, that is, accidents served later than T time units. To determine this fraction, several simulations are performed. For each simulation, the number of accidents and the number of accidents reached within time threshold T are tracked. To calculate the fraction of late arrivals, the number of accidents that are reached later than T is divided by the number of accidents.

In the following section, the concept of MEXCLP is explained.

2.1 MEXCLP

The MEXCLP is an integer linear programming problem that searches for the best static solution of initial base location for each ambulance. The MEXCLP solution gives for each demand location the number of ambulances that must cover the location and for each base the number of ambulances it holds, such that the total coverage is maximized. A good coverage by ambulances prevents that accidents are served later than a specific time threshold T. Therefore, optimizing the coverage reduces the expected fraction of late arrivals, the KPI, which aligns with our objective.

[2] In the Netherlands, postal codes start with four digits followed by two letters.

In this model, there is a set of possible base locations W, and for each base location a number of ambulances must be distributed. Let $|A|$ be the total number of ambulances available and let q denote the busy fraction, the probability an ambulance is unavailable, which is assumed to be the same for all ambulances. The busy fraction is estimated by dividing the expected load of the system by the total number of available ambulances. Moreover, let W_i be the set of bases within range of demand location $i \in V$, that is, the bases of which the travel time is within T time units of a demand location.

Let the decision variables be the variables x_j, the number of ambulances at each base, for $j \in W$, and the binary variables y_{ik}, where y_{ik} is 1 if demand location $i \in V$ is covered by k ambulances and 0 otherwise. This value k is determined using the travel times τ_{ij}.

Lastly, the expected covered demand is introduced, which is defined as $E_k = d_i(1 - q^k)$. This number is the expected coverage when there are k ambulances within range of demand location $i \in V$. We want to find the number of ambulances $k - 1$ for which adding another ambulance yields the highest increase in coverage. This improvement can be calculated by taking the difference between E_k and E_{k-1}. The marginal contribution of the kth ambulance is then defined as $E_k - E_{k-1} = d_i(1 - q)q^{k-1}$.

The MEXCLP is formulated as:

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

$$\text{s.t.} \sum_{j \in W_i} x_j \geq \sum_{k=1}^{p} y_{ik}, \quad i \in V, \tag{2}$$

$$\sum_{j \in W} x_j \leq |A|, \tag{3}$$

$$x_j \in \mathbb{N}, \quad j \in W, \tag{4}$$

$$y_{ik} \in \{0, 1\}, \quad i \in V, k = 1, \ldots, p. \tag{5}$$

The objective function is defined in Equation (1), which is the sum of marginal contributions over all demand locations. Equation (2) ensures that the number of ambulances at base locations in range of demand location $i \in V$ is at least as high as the number of ambulances that covers location i. Equation (3) states that the sum of the number of ambulances at each base location is at most the number of available ambulances. Furthermore, x_j is an integer and y_{ik} a binary number, as indicated by Equations (4) and (5) respectively. Hence, the MEXCLP is used to optimize the total coverage.

2.2 DMEXCLP

Beside the static MEXCLP, Jagtenberg has introduced an algorithm that implements the MEXCLP for the dynamic problem. An ambulance becomes idle after it has served an accident. Each time an ambulance becomes idle, the DMEXCLP algorithm searches for the next base location it needs to drive to. For each base location, the coverage improvement is calculated if it is sent to that location. The chosen destination is the location that results in the maximum total coverage. Furthermore, the algorithm does not just send an ambulance to a base location that has the least number of ambulances, as this does not necessarily means a better coverage. Therefore, the DMEXCLP proves to be very useful to solve the dynamic relocation of ambulances. The DMEXCLP algorithm can be found in Appendix C as Algorithm 1.

2.3 Discrete Event Simulation

In order to determine the performance for a policy, a discrete event simulation (DES) needs to be performed, which stores and updates relevant statistical values. For this simulation, the results acquired from the MEXCLP and the algorithm for the DMEXCLP are used.

Before explaining the components, the DES model will be shortly described. It is a system that is modeled in terms of its state at each point in time. At discrete points in time, events occur that may change the system state. After an event has occurred, the system state and the simulation clock are updated. The simulation clock is a variable that keeps track of the current value of the simulated time, which continues to proceed until it has reached the end time of the simulation. The DES also keeps track of statistical counters, which are variables containing valuable information about the system's performance.

The components of a DES model are the initialization routine, in which the variables are initialized, the timing routine, in which the next event time is established and simulation clock is updated, the event routines, which performs in several steps the events, the library routines, which are used to generate random variables, the report generator, which reports and summarizes results at the end of the simulation, and the main program, which ties all routines together and executes them in the right order.

The state space is defined by using the destinations of all idle ambulances, which can only be base locations. If an ambulance is already stationed at a base location, its destination is the base location it is currently at. Because each ambulance is identical or exchangable, the state space $\mathcal{S}$ can be defined as the set of states $s = \{n_1, \ldots, n_{|W|}\}$, where n_j is the number of idle ambulances that has destination base location j, for $j \in W$. The system state changes if the number of idle ambulances changes.

There are two events that may change the system state s: the moment an accident occurs and the moment an ambulance finishes the service. In case an accident occurs, the closest idle ambulance gets sent to the accident location. If this happens, the number of idle ambulances reduces by one, therefore changing the system state. It may occur that there are no idle ambulances available to immediately serve the accident. In this case, the accident is stored in a first-come-first-serve queue and the system state remains unchanged. In case an ambulance finishes the service, it becomes free to serve other accidents, therefore the number of idle ambulances increases by one. However, if the queue is not empty, the ambulance that became just available will immediately drive to the accident location that is first in queue, meaning its status remains busy.

The next accident times for each demand location is generated at the start of the simulation using the exponential distribution. By multiplying the accident rate λ with the respective demand fraction g_i of each location $i \in V$, $|V|$ smaller rates are obtained which are used to generate the next accident time at the respective demand location.

In the accident event routine, the time an ambulance finishes the service is generated. The finish time is generated by cumulating the service time and adding it to the current time t. Let the time an accident occurs correspond to the time a call has been received by the emergency room. Then it takes time τ_{answer} before an ambulance is on the road to drive to the demand location. The nearest ambulance is sent from location i to demand location j, taking up travel time τ_{ij}, for i, $j \in V$. After the ambulance has arrived at the demand location, the medical employees provide service that takes up time $\tau_{onscene}$, which is generated using the exponential distribution with rate λ_s. Afterwards, it is decided whether the patient needs a transport to the hospital for further service. This is decided by using a coin flipping method. With probability p, a transport to the hospital is needed, and with probability $(1-p)$, no transport is needed. If a transport is not required, the ambulance immediately becomes idle. Otherwise, it drives from the current location j to the nearest hospital location $k \in H$, taking up travel time τ_{jk}. After the ambulance arrives at the hospital, the time it remains there is denoted by $\tau_{hospital}$, after which it becomes idle. $\tau_{hospital}$ is generated using the Weibull distribution with shape parameter λ_h and scale parameter α. The finish time is therefore $t + \tau_{answer} + \tau_{ij} + \tau_{onscene} + \tau_{jk} + \tau_{hospital}$ with probability p and $t + \tau_{answer} + \tau_{ij} + \tau_{onscene}$ with probability $(1-p)$.

The response time is the time between the moment a call is received by the emergency room and the moment an ambulance arrives at the accident scene. Note that $\tau_{answer} + \tau_{ij}$ is the response time of an accident. Since our objective is to minimize the number of accidents served later than T time units, it is beneficial to have a low average response time.

When an ambulance becomes idle, it may either get send to its original base location under the static policy, which is determined by solving the MEXCLP, or to the base location that yields the best coverage improvement under the dynamic policy. In the latter case, the DMEXCLP algorithm is performed, which gives as output the best base location. When travelling to a base, the travel speed of the ambulance is v times the speed when it drives to an accident location, $0 < v < 1$.

An idle ambulance that is on the way to a base location and is the closest ambulance available does not need to first arrive at the base when an accident occurs. To determine the current location of this idle ambulance, we make use of linear interpolation. First, the fraction of the travel is calculated by the formula $\frac{\text{current time}-\text{start time}}{\text{end time}-\text{start time}}$, where start time is the time the ambulance becomes idle and end time is the time the ambulance arrives at the destination. Secondly, the coordinates of the current position are calculated using this fraction and the coordinates of the origin and the destination. Lastly, we calculate the distance between the current position and each demand location to determine the nearest demand location. The nearest demand location is assumed to be the current position of the idle ambulance, since only distances between demand locations are known. When an accident occurs, the simulation model first determines the closest base location that contains available ambulances and afterwards checks whether the location of the idle ambulance that is still on the way is closer to the accident location.

The performance measures of interest are the expected fraction of late arrivals (the KPI) and the expected average response time. These measures are good indicators to decide whether our current policy maintains a good service, since these evaluate how many accidents are served within time and how long it takes to arrive at the accident scene. Further statistical measures are the expected number of accidents occurring, the expected number of accidents reached within T time units, the expected number of accidents served, the expected number of times a transport is required and the number of accidents that have been in queue. Since the ambulance redeployment is a continuous process, it would be desirable not to start with an empty system, i.e. no accidents have occurred at all. For this reason, our simulation has a warm up period of h hours. In the first h hours, none of the statistical counters are updated yet.

An overview of the definitions of notations can be found in Appendix B and algorithms for each component of the simulation in Appendix C.

3 Data

As we want to replicate the research from Jagtenberg et al. [2], we mostly use the same data. According to the statistical overview [4], we have a total of 217 demand locations and 11 base locations. The postal codes that are used to derive the coordinates and the population of each demand location are obtained from the RIVM. In Jagtenberg et al. [2], on average 9.5 accidents occur within an hour in the RAV of Utrecht, which is equivalent to an accident rate of $\lambda = \frac{9.5}{3600}$, as the time unit in our simulation model is in seconds. This means accidents at location i occur with rate $\frac{9.5}{3600}g_i$. Furthermore, 8 hospitals are found in the RAV of Utrecht.

The number of ambulances is also acquired from the research of Jagtenberg [2]. In total, 19 ambulances are used. The time before an ambulance is sent to an accident location τ_{answer} is 3 minutes. The travel times are deterministic and are acquired from the RIVM. The probability p an accident need a transport is derived from the data in the statistical overview [4] by calculating the fraction of the number of transports to the total number of accidents. For this number, the data of the year 2010 is used. In 2010, the number of accidents that needed a transport in the RAV of Utrecht amounts 58961 and the total number of accidents amounts 85152, so p is approximately 0.69. For $\tau_{onscene}$ we use an exponential distribution with rate $\frac{1}{720}$ and for $\tau_{hospital}$ we use a Weibull distribution with shape parameter $\frac{1}{1080}$ and scale parameter 1.5. The time threshold T, as already indicated before, is 15 minutes, which means an ambulance has at most 12 minutes to be on the road driving to an accident location.

Like in the research of Jagtenberg, the busy fraction q is equal to 0.3. If an idle ambulance is on the road driving to a base location, its travel speed is 0.9 times the speed when it drives to an accident location. The initialization of the system state is done by using the result obtained from the MEXCLP, after solving it by using our fixed set of base locations. For generating random variables, the Mersenne Twister random generator is used with a fixed seed value of 12345, which is set before the simulation runs. Lastly, a warm up period of 5 hours is used.

4 Results and analysis

Performing the DES model yields results for both the static and dynamic policies.

Table 1 shows the results for both the static and dynamic policies for 500 simulation hours, 10 runs, $\lambda = \frac{9.5}{3600}$ and $q = 0.3$. The corresponding standard deviations are in the columns next to the results. From the standard deviations, we see that we can use these results as performance measures, as the deviations are relatively not high. It can be seen that the expected number of accidents occurring is approximately the same as the expected number of accidents served in the same time window, meaning the time interval between accidents occurring and the time interval between ambulances finishing a service are approximately the same. The expected number of accidents that are handled within 15 minutes is higher under the dynamic policy, which coincides with the lower expected fraction of late arrivals (the KPI).

Performance measures	Static policy		Dynamic policy	
	Mean	St. Dev.	Mean	St. Dev.
No. of accidents occurring	4762.5	50.4276	4762.5	50.4276
No. of accidents handled on time	4233.3	35.1822	4357.4	41.9582
No. of accidents served	4763.2	50.1947	4763.0	49.9355
No. of times transported	3303.2	64.4012	3303.2	64.4012
No. of accidents in queue	0.0	0.0	0.0	0.0
Fraction of late arrivals	0.1111	0.0053	0.0850	0.0056
Average response time (in seconds)	608.51	3.6574	578.02	3.9015

Table 1: Performance of the static policy and the dynamic policy with $\lambda = \frac{9.5}{3600}$ and $q = 0.3$ for 500 simulation hours and 10 runs.

One remarkable observation is that there has been no accident put in the queue in every run. This is due to the high number of ambulances we use, namely 19. Since the time interval between accidents occurring and the time interval between ambulances finishing a service are approximately the same, we always have an ambulance becoming idle for every accident that occurs before all ambulances are busy. We have confirmed that, when we use lower numbers of ambulances, some accidents are put in the queue.

Furthermore, Figure 1 on page 11 shows a comparison in the performance between the static and the dynamic policy. The performances of each of the 10 simulation runs have been plotted. It can be clearly seen from both Table 1 and Figure 1 that implementing the dynamic MEXCLP results in a better performance than the static policy. A relative decrease of 23.5%

in the expected fraction of late arrivals can be gained by performing the DMEXCLP, which is a significant improvement.

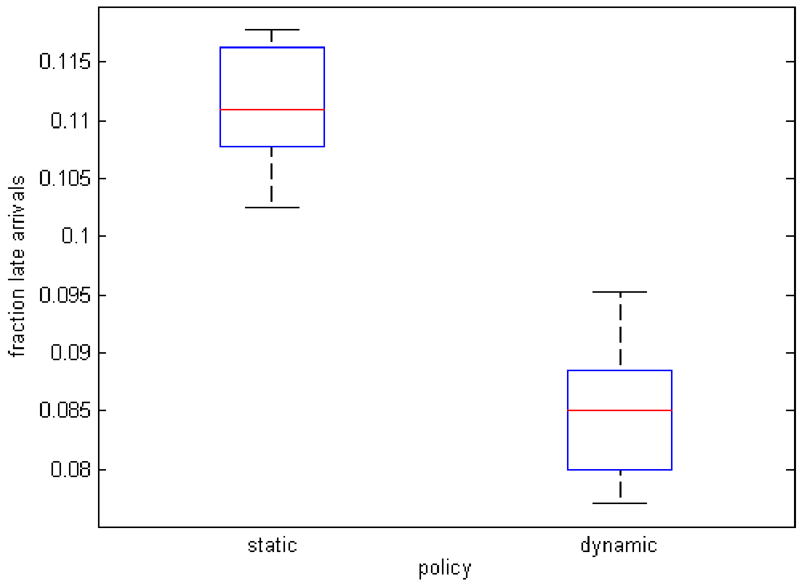

Figure 1: Comparison between the performance of the static policy with the performance of the dynamic policy using $\lambda = \frac{9.5}{3600}$ and $q = 0.3$ for 500 simulation hours and 10 runs.

Figure 2 on page 12 shows a comparison in the response times between the static policy and the dynamic policy, for 1 simulation and 2500 simulation hours. Like Table 1, it shows that the average response times are lower under the dynamic policy than under the static policy, as the line of the dynamic policy is constantly above the line of the static policy and it approaches faster to 1.

Overall, from these results we can conclude that the dynamic policy greatly improves the performance of the EMS system. We may, however, want to check whether changing a parameter like the accident rate will affect this result.

In the next subsections, some further analysis are discussed. First, in Section 4.1, we check the effect of changing the accident rate λ on the expected fraction of late arrivals and the expected average response time of both policies and compare them. Afterwards, in Section 4.2, the sensitivity of the dynamic policy on the busy fraction q is investigated, by performing the simulation for several values of q. Section 4.3 introduces a new set of

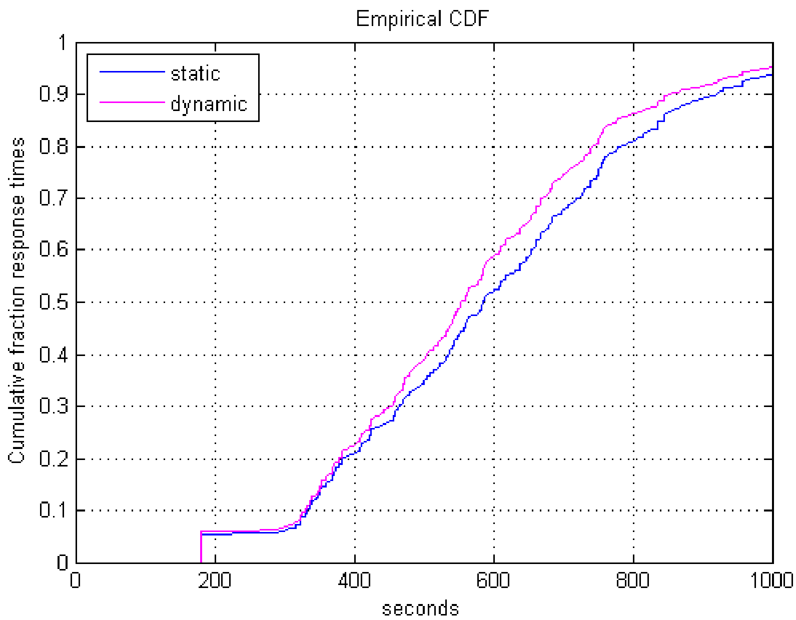

Figure 2: Comparison in the response times between the static policy with the dynamic policy using $\lambda = \frac{9.5}{3600}$, $q = 0.3$, 2500 simulation hours and 1 run.

base locations, after we solve an MEXCLP for which every demand location can potentially serve as a base location. The solution is used as an initialization of the system state in our simulation model to check its performance. Section 4.4 investigates the effect in the performance of our model under the dynamic policy when different time standards T are applied. Lastly, in Section 4.5, the performance of our policies are checked after using different accident rates within a day, assuming that accidents occur according to a non-homogeneous Poisson process.

4.1 Performance by changing the accident rate

Since there is chance that the static policy may outperform the dynamic policy for a different accident rate, we perform simulations using different accident rates. The simulation is performed for 1000 simulation hours and 10 runs and the busy fraction q is hold fixed at 0.3. Figure 3 on page 13 shows a chart with the absolute performance of both the static policy and the dynamic policy for various accident rates. It can be seen that the dynamic policy always outperforms the static policy, for every λ.

However, it can also be seen from Figure 3 that the difference in the absolute performance between the two different policies is very small at extreme time intervals between accidents. This can be easily seen in the figure, as the first and last points of both policies are almost equally positioned. At most, the static policy performs just as good as the dynamic policy.

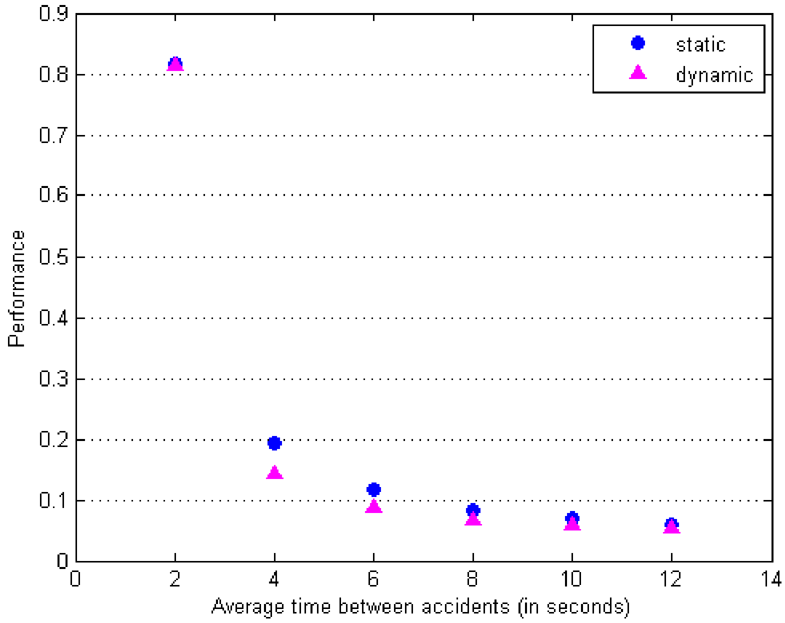

Figure 3: Comparison in the absolute performance between the static policy and the dynamic policy for $q = 0.3$, 1000 simulation hours and 10 runs, when different accident rates are used.

Figure 4 on page 14 shows the relative improvement in the expected fraction of late arrivals when the dynamic policy is applied over the static policy. This value peaks when the average time between accidents is around 4 minutes, which is around 26.1%, after which the improvement increasingly decreases for a higher average time between accidents.

4.2 Sensitivity to the busy fraction

Now, we investigate the sensitivity of the dynamic policy to the busy fraction q. We hold the accident rate λ fixed at $\frac{9.5}{3600}$ and we simulate 10 runs of 1000 simulation hours each for every value of q. Table 2 shows the performance measures for different values of q. As can be seen in the table, the expected fraction of late arrivals is not very sensitive to the value of q for lower values

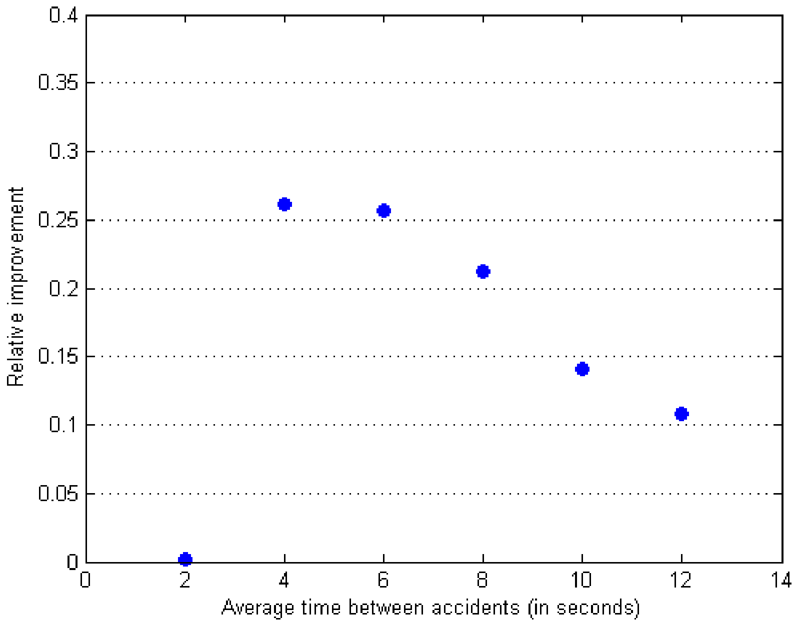

Figure 4: The relative improvement in performance when applying the dynamic policy over the static policy for $q = 0.3$, 1000 simulation hours, 10 runs and different accident rates.

of q, as there is small difference in the performance of our dynamic policy for $q = 0.2$ and $q = 0.4$.

Performance measures	$q = 0.2$		$q = 0.4$		$q = 0.6$		$q = 0.8$	
	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
Fraction of late arrivals	0.0843	0.0021	0.0855	0.0026	0.0991	0.0039	0.2303	0.0054
Av. response time (in seconds)	574.42	2.5873	579.29	1.8615	588.41	2.9846	708.15	3.5626

Table 2: Obtained performance values for different values of q, $\lambda = \frac{9.5}{3600}$ after 10 runs for 1000 simulation hours each.

When the busy fraction exceeds 0.5, it becomes increasingly harder to find available ambulances. The increase in both the expected fraction of late arrivals and the expected average response time are still relatively small for $q = 0.6$, with a relative increase of 15.9% and 1.6% respectively, compared to the performances of $q = 0.4$. However, for $q = 0.8$, the relative increases are 132.4% and 20.3% respectively, compared to the performances of $q = 0.6$.

In other words, the performance of the dynamic policy increasingly deteriorates for a higher busy fraction. This makes sense, as the probability that an ambulance is available decreases and it becomes more difficult to serve accidents within the time threshold.

4.3 Reallocation of base locations

We solve the MEXCLP again, but without a specific set of base locations known beforehand. We first assume the set of possible base locations to be the set of demand locations. After solving the MEXCLP, our solution shows at which locations ambulances are stationed. Now we let the set of base locations be the set of all locations that has at least one ambulance stationed. Fifteen locations in total have at least one ambulance stationed. Appendix D shows the locations and the corresponding number of ambulances. In addition, our previous base locations and the corresponding number of ambulances can be seen for comparison.

Using the new set of base locations, we perform the simulation for both the static and dynamic policy 10 times for 500 simulation hours each using $\lambda = \frac{9.5}{3600}$ and $q = 0.3$. Table 3 shows the performance measures of the static and dynamic policy. As can be seen, when comparing to Table 1, huge improvement in the expected fraction of late arrivals for both the static and dynamic policy can be achieved. The expected fraction of late arrivals of the static policy shows a relative improvement of 37.3% and the expected fraction of late arrivals of the dynamic policy a relative improvement of 51.3%. The expected average response time also shows a significant improvement over the earlier results. The results make sense, as, with our new base locations, we have optimized the total coverage over the whole RAV of Utrecht. The total coverage has improved relatively with 5.5% over the old situation.

Performance measures	Static policy		Dynamic policy	
	Mean	St. Dev.	Mean	St. Dev.
Fraction of late arrivals	0.0697	0.0047	0.0414	0.0037
Av. response time (in seconds)	596.86	4.5689	570.04	3.3794

Table 3: Obtained performance values for the new set of base locations, for $q = 0.3$, $\lambda = \frac{9.5}{3600}$ after 10 runs for 500 simulation hours each.

We conclude that the MEXCLP is a capable model to optimize base locations, since great improvements in the performance of our policies can be seen. The question remains whether it would be possible to set a base location at any location the model indicates under real circumstances. Nevertheless, it gives a good and fast indication for research purposes. If certain circumstances changes, like an additional ambulance becomes available in our model, we can simply use the MEXCLP to determine the best base locations.

4.4 Different response time standards

We investigate the performance of the dynamic policy when we change the time threshold T. The acceptable time standard for ambulances to arrive at the accident location varies between countries. As we already mentioned, the time standard for the Netherlands is 15 minutes. Some other countries apply a different time standard, which are usually much stricter, like 8 or 10 minutes. We are interested in the performance of the Dutch system when other time standards are applied.

Table 4 shows the performance of the dynamic policy when applying different time standards. For this outcome, we have used our fixed set of base locations. The corresponding initialization of the system state is found in Appendix E in Table 12. As expected, the performances are very poor when the Dutch medical system applies these different time standards. For a time threshold of 12 minutes, only $\frac{4}{5}$ of the accidents are reached in time. For a time threshold of 8 minutes, less than half of the accidents are reached in time.

Performance measures	$T=8$		$T=10$		$T=12$		$T=15$	
	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
Fraction of late arrivals	0.5850	0.0042	0.3838	0.0057	0.2276	0.0029	0.0848	0.0026
Av. response time (in seconds)	582.62	3.0220	576.24	3.3630	574.36	2.0583	576.44	2.1176

Table 4: Obtained performance values for different values of T, using $q = 0.3$, $\lambda = \frac{9.5}{3600}$, 1000 simulation hours and 10 runs.

We see that the current system of the Netherlands is not adaptable to the time standards of other countries and is mostly optimized with respect to their own time standard. Therefore, we try to make the system adaptable to other time standards by choosing new base locations using the method described in Section 4.3 and changing the number of ambulances to acquire a new set of base locations and an optimal allocation of ambulances. We consider a policy acceptable if the expected fraction of late arrivals is at most 0.1. To determine the number of ambulances that achieves this performance, we manually change the number of ambulances and check its performance, until we reach our desired performance. In other words, we perform a linear search.

Table 5 shows the performance of the dynamic policy for different T values using the new optimized base locations and the new optimized allocations of ambulances. These performances are acceptable, as the expected fraction of late arrivals fall just below 0.1. We see that, in order to have an acceptable performance for $T = 8$, we need nearly four times the number of bases of $T = 15$ and more than double the number of ambulances.

Performance measures	$T = 8$		$T = 10$		$T = 12$	
	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
Number of bases	42	-	23	-	19	-
Number of ambulances	48	-	28	-	21	-
Fraction of late arrivals	0.0973	0.0017	0.0946	0.0027	0.0914	0.0048
Av. response time (in seconds)	371.38	0.8313	457.92	1.7968	528.55	2.9724

Table 5: The performance of the dynamic policy for different T values, after optimizing the base locations and the number of ambulances, using $q = 0.3$, $\lambda = \frac{9.5}{3600}$, 1000 simulation hours and 10 runs.

In short, lots of rearrangement and reallocation of base locations and ambulances are required, should the Dutch time standard change, to maintain an acceptable performance. Tables 13-15 in Appendix E show the new base locations and allocation of ambulances for different values of T.

4.5 Non-constant accident rate

As our last research, we consider the fact that the accident rate changes multiple times during the day. This is a reasonable assumption, since it is likely that accidents occur more frequently at certain hours than at other hours. We assume that accidents follow a piecewise constant non-homogeneous Poisson process. This means that the accident rate changes during the day, but for a certain period the accident rate remains constant.

From the reference framework [1], we find that the number of ambulances varies depending on the part of the day and also differs between weekdays, Saturdays and Sundays. The parts of the day considered are 12 pm to 8 am, 8 am to 4 pm and 4 pm to 12 pm. Since the number of ambulances used may be an indication of the bustle on a particular part of the day, we use the different numbers of ambulances for each part to derive the different accident rates during the day. For simplicity, we only consider the rates of the weekdays.

The number of ambulances for all emergencies are as follows: between 12 pm and 8 am 15 ambulances are deployed, between 8 am and 4 pm 35 ambulances are deployed and between 4 pm and 12 pm 22 ambulances are deployed. Even though these numbers are not just for A1 emergencies, we can still approximate the proper numbers of ambulances. We assume that $\lambda = \frac{9.5}{3600}$ is the highest rate. We use this rate and a total of 19 ambulances between 8 am and 4 pm, as within this time interval the number of ambulances for all emergencies is the highest. We further assume that the proportion of the number of ambulances used for A1 emergencies to the number of ambulances used for all emergencies is the same at any time. Using this approximation, we find that 9 ambulances are deployed between 12 pm and 8 am, 19 ambulances between 8 am and 4 pm and 12 ambulances between 4 pm and 12 pm. In a similar way, we derive the corresponding accident rates for each part of the day, which are $\frac{1}{800}$, $\frac{9.5}{3600}$ and $\frac{1}{600}$ respectively.

Some updates are made in the simulation algorithms. First of all, a new event is introduced, namely the change in the part of the day. Furthermore, we have to make sure that both the number of ambulances $|A|$ and the accident rate λ change when the simulation is at the next part of the day. We use the new notations $|A_{part}|$ and λ_{part} to distinguish between the different values. When the simulation time is updated, we check whether the current time is at the next part of the day. If this is the case, our new event is invoked. We also reverse the simulation clock back to the time at which the change in the part occurs.

Furthermore, we solve the MEXCLP using our fixed set of base locations with the numbers of ambulances for each part of the day and store the solutions. We assume that our simulation excluding the warm up period starts at 12 pm. Since we have a warm up period of 5 hours, our simulation including the warm up period actually starts at 7 pm, which means we initialize the system state with the MEXCLP solution for 12 ambulances.

In our new event routine, we update the state space by adding or removing ambulances from base locations. This is done by determining the number of ambulances that are actually at each base location and by calculating the differences of the solutions from the MEXCLP for different $|A_{part}|$ values and adding or removing these differences to each base location. To illustrate this, at the base location located at postal code 3436, we start with a total of 3 ambulances in the simulation. At the next part of the day, the MEXCLP solution indicates that we should start with 2 ambulances at that location. So, when we arrive at the next part of the day, after having determined the number of ambulances that are actually at the base location, we subtract 1 ambulance from the base location. This way, we do not remove an ambulance that is still busy or on the way to a base location. Because the number of ambulances at a base may be negative after subtracting, any excess ambulance is automatically removed when it returns to the base location with a negative number of ambulances.

Since the accident rates change during the day, we also make sure that the next accident is generated one at a time using the proper rate. A number is generated between 0 and 1. We add up the demand fraction of each demand location to the total until the sum exceeds our generated number. The demand location of which the fraction is added last is our next accident location. Afterwards, the next accident time is generated. This process is done at the end of the updated accident event routine or the change in the part of the day event routine. Appendix F shows the updated algorithms for our model.

In Table 6, the results of the new simulation model for both the static and dynamic policy can be seen, using $q = 0.3$ for 500 simulation hours and 10 runs. From the table, we see worse performances in both the expected fraction of late arrivals and the expected average response time compared to the performances from Table 1. This is due to our chosen accident rates and number of ambulances. When we perform the simulation with only $\lambda = 800$ and 9 ambulances using $q = 0.3$, 500 simulation hours and 10 runs, the expected fraction of late arrivals for the static and dynamic policy are 0.2535 and 0.2353 respectively. These performances are absolutely bad and it is very likely that we have not used the appropriate arrival rates. However, as we do not have any data on the different accident rates, our only option is to use these approximations.

Performance measures	Static policy		Dynamic policy	
	Mean	St. Dev.	Mean	St. Dev.
No. of accidents occurring	3331.9	40.9863	3331.5	44.4953
No. of accidents handled on time	2785.8	17.6623	2907.6	36.4789
No. of accidents served	3332.9	39.8119	3332.5	43.6100
No. of times transported	2310.2	36.9197	2308.1	36.6134
No. of accidents in queue	5.8	3.7947	4.7	4.3474
Fraction of late arrivals	0.1638	0.0079	0.1272	0.0052
Average response time (in seconds)	654.46	4.7722	618.94	5.9019

Table 6: Performance of the static policy and the dynamic policy using different accident rates and $q = 0.3$ for 500 simulation hours and 10 runs.

Another limitation to our approximation is that we base our accident rates on the number of ambulances that are used per part of the day. Since the number of ambulances only changes three times per day, we can only acquire three different accident rates that change constantly after eight hours. However, in real time the accident rate is way more volatile and may change in a matter of seconds. Moreover, since we assume a piecewise constant behaviour for accident rates, the rates abruptly change after eight hours. We could account for this by assuming that the accident rates follow a piecewise linear non-homogeneous Poisson process. Again, since we lack the data for it, we unfortunately cannot implement it.

5 Conclusion

Using the methods from the research of Jagtenberg et al. [2], we have solved the dynamic ambulance redeployment problem. In particular, the dynamic variant of the MEXCLP searches for the next best location for an ambulance that has just become idle, which is implemented in the simulation model. After performing several simulations, we acquire and analyze our results. Furthermore, the methods from the research of Jagtenberg are very easy to implement and to adjust for further analysis.

As we have already seen in Section 4, the dynamic policy yields generally better performances than the static policy, with a relative improvement of 23.5% in the expected fraction of late arrivals when we replicate the research of Jagtenberg. Nevertheless, the static policy is still useful as a benchmark. Further analysis shows that the dynamic policy performs better than the static policy when we use different accident rates, although for extreme values of the accident rates the differences in performance between the static and dynamic policy are minimal. When we change the busy fraction q, our performances are not much affected for lower values of q. However, when we use higher values of q, the performances start to worsen exponentially.

Our research shows that by including all demand locations in our set of base locations and solving the MEXCLP we can obtain a new set of base locations. Since this set of base locations optimizes the total coverage, we obtain much better performances using this set than using our fixed set of base locations, with relative improvements of 37.3% and 51.3% for the static and dynamic policy respectively. The MEXCLP can also be used to determine which base locations are needed to acquire a decent performance when different time standards T are used. In order to derive the required number of ambulances for each T, we perform a linear search. We see that a lot more base locations and ambulances are needed in order to obtain an acceptable performance for lower T values.

By assuming that accident rates are not constant over the day, we modified our simulation model to include the different accident rates and numbers of ambulances per part of the day. Since we do not have any data on the different accident rates, we have to use approximations. These approximations lead to worse performances for both the static and dynamic policy than before.

All in all, the MEXCLP and the simulation model are very simple and flexible methods, and especially useful to analyze the dynamic ambulance redeployment problem. Should more data on the accident rates be collected, further research could include piecewise linear accident rates. For now, our methods are good enough to warrant a good performance.

A Locations overview

1391	3439	3515	3562	3626	3723	3791	3902	3984
1393	3441	3521	3563	3628	3731	3811	3903	3985
1396	3442	3522	3564	3631	3732	3812	3904	3989
1426	3443	3523	3565	3632	3734	3813	3905	3991
1427	3444	3524	3566	3633	3735	3814	3906	3992
3401	3445	3525	3571	3634	3737	3815	3907	3993
3402	3446	3526	3572	3641	3738	3816	3911	3994
3403	3447	3527	3573	3642	3739	3817	3912	3995
3404	3448	3528	3581	3643	3741	3818	3921	3997
3405	3449	3531	3582	3645	3742	3819	3922	3998
3411	3451	3532	3583	3646	3743	3821	3927	3999
3412	3452	3533	3584	3648	3744	3822	3931	4121
3413	3453	3534	3585	3701	3749	3823	3941	4122
3415	3454	3541	3601	3702	3751	3824	3945	4124
3417	3455	3542	3602	3703	3752	3825	3947	4131
3421	3461	3543	3603	3704	3754	3826	3951	4132
3425	3464	3544	3604	3705	3755	3828	3953	4133
3431	3467	3545	3605	3706	3761	3829	3956	
3432	3471	3546	3606	3707	3762	3831	3958	
3433	3474	3551	3607	3708	3763	3832	3959	
3434	3481	3552	3608	3709	3764	3833	3961	
3435	3511	3553	3611	3711	3765	3834	3962	
3436	3512	3554	3612	3712	3766	3835	3971	
3437	3513	3555	3615	3721	3768	3836	3972	
3438	3514	3561	3621	3722	3769	3901	3981	

Table 7: All four digits demand locations in the RAV of Utrecht.

3436	3447	3561	3582	3608	3645	3707	3811	3823
3911	3941							

Table 8: All four digits base locations in the RAV of Utrecht.

3435	3447	3543	3582	3707	3743	3813	3831

Table 9: All four digits hospital locations in the RAV of Utrecht.

B Notation overview

Notation	Description
A	The set of ambulances in the RAV of Utrecht.
V	The set of demand locations in the RAV of Utrecht.
W	The set of base locations in the RAV of Utrecht, $W \subseteq V$.
H	The set of hospital locations in the RAV of Utrecht, $H \subseteq V$.
T	The time threshold.
λ	The accident rate in the RAV of Utrecht.
d_i	The demand at location $i \in V$.
g_i	The demand fraction at location $i \in V$.
τ_{ij}	The driving time between i and j when driving to an accident location, $i, j \in V$.
τ_{answer}	The time before an ambulance is dispatched to an accident location.
$\tau_{onscene}$	The time the ambulance is at the accident location.
$\tau_{hospital}$	The time the ambulance is at the hospital.
λ_s	The rate to generate $\tau_{onscene}$.
λ_h, α	The shape and scale parameters to generate $\tau_{hospital}$.
p	The probability an accident needs a transport.
q	The busy fraction.
v	The fraction of the speed an idle ambulance travels to a base location.
$\mathcal{S}$	The state space of the simulation.
s	The states $\{n_1, \ldots, n_{\|W\|}\}$ $\mathcal{S}$ consists of.
n_j	The number of ambulances that has destination $j \in W$.
W_i	The set of base locations within range of demand location $i \in V$, $W_i \subseteq W$.
k	The number of ambulances that covers demand location $i \in V$.
x_j	The number of ambulances at location $j \in W$.
y_{ik}	Binary value equal to 1 if demand location $i \in V$ is covered by k ambulances and 0 otherwise.
E_k	The expected coverage when there are k ambulances within range of demand location $i \in V$.
t	The current time.
h	The warm up period for the simulation.

C Simulation algorithms

Algorithm 1 Dynamic MEXCLP

Require: Demand d_i of each demand location $i \in V$,
base locations $W \subseteq V$,
busy fraction $q \in (0, 1)$,
current destinations $dest(a)$ for all $a \in IdleAmbulances \subseteq A$,
travel times τ_{ij} between any $i, j \in V$,
time threshold T to reach an emergency call.

Output: new destination for the ambulance that is about to become idle. This ambulance should not be counted as an idle ambulance yet.

$BestImprovement = 0.$
$BestLocation = NULL.$
for each $j \in W$ **do**
    $CoverageImprovement = 0.$
    for each $i \in V$ **do**
        $k = 0.$
        if $\tau_{ji} \leq T$ **then**
            k++.
            for each $a \in IdleAmbulances$ **do**
                if $\tau_{dest(a)i} \leq T$ **then**
                    k++.
                end if
            end for
            $CoverageImprovement+ = d_i(1 - q)q^{k-1}.$
        end if
    end for
    if $CoverageImprovement > BestImprovement$ **then**
        $BestLocation = j.$
        $BestImprovement = CoverageImprovement.$
    end if
end for
Return $BestLocation.$

Algorithm 2 Initialization routine

Require: Demand locations $i \in V$ and its demand d_i, demand fraction g_i,
coordinates and whether also $i \in W$ or $i \in H$,
matrix of travel times between locations τ_{ij}, $i, j \in V$,
system state with number of ambulances with destination $j, \forall j \in W$,
number of vehicles $|A|$,
busy fraction $q \in (0, 1)$,
accident rate λ,
driving time threshold T,
boolean value that determines static or dynamic policy,
total simulation hours u,
total simulation runs.

Initialize variables and statistical counters f.
for each simulation run **do**
    Invoke Algorithm 3.
end for
Print statiscal counters f.

Algorithm 3 Main event loop

Require: System state with number of ambulances with destination $j, \forall j \in W$.

Initialize current time t.
Initialize list of next accidents, finish times, system state and queue.
Initialize statistical counters used within a run.
Generate next accident time of each location $i \in V$.
while $t < u$ **do**
    Invoke Algorithm 4.
    if $t > u$ **then**
        Do not count the next event.
    end if
    if next event is accident **then**
        Invoke Algorithm 5.
    else
        Invoke Algorithm 6.
    end if
end while
Update statistical counters f.

Algorithm 4 Timing routine

Store next accident time and finish times of all ambulances in a list.
if earliest time is accident **then**
  Next event is accident.
else
  Next event is finish.
end if
Update t.
for each idle ambulance still on the way a_{onway} **do**
  if t exceeds arrival time to base **then**
    Remove a_{onway}.
  end if
end for

Algorithm 5 Accident event routine

Generate next accident time for the current location $i \in V$.
if number of busy ambulances equals $|A|$ **then**
  Store accident time and location in queue.
else
  Initialize $ShortestTravelTime$ and $BestBase$.
  for each $j \in W$ **do**
    if $\tau_{ji} < ShortestTravelTime$ and j contains ambulances **then**
      $ShortestTravelTime = \tau_{ji}$ and $BestBase = j$.
    end if
  end for
  for each $a_{onway} \in IdleAmbulances \subseteq A$ **do**
    Determine current position.
    for each $k \in V$ **do**
      Calculate distance between k and current position.
      Store k if smallest distance.
    end for
    if $\tau_{ki} < ShortestTravelTime$ **then**
      $ShortestTravelTime = \tau_{ki}$ and $BestBase = k$.
    end if
  end for
  Update system state.
  Invoke Algorithm 7.
end if

Algorithm 6 Finish event routine

Determine which ambulance has finished service.
if Queue is empty **then**
  Make ambulance idle and store current location and time.
  if policy is static **then**
    Store original base location and update system state.
  else
    Invoke Algorithm 1.
    Store best location and update system state.
  end if
  Calculate and store the arrival time of the idle ambulance.
else
  Remove the first accident from the queue.
  Invoke Algorithm 7.
end if

Algorithm 7 Service method

Generate $\tau_{onscene}$.
Generate number o between 0 and 1.
if $o > p$ **then**
  Calculate finish time and store current location $i \in V$.
else
  Initialize $ShortestTravelTime$ and $BestHospital$.
  for each $k \in H$ **do**
    if $\tau_{ik} < ShortestTravelTime$ **then**
      $ShortestTravelTime = \tau_{ik}$ and $BestHospital = k$.
    end if
  end for
  Generate $\tau_{hospital}$.
  Calculate finish time and store current location $k \in H$.
end if
Store original base location.

D Optimized base locations

Base	Number of ambulances
3417	1
3439	1
3461	1
3471	1
3541	2
3573	2
3628	1
3645	1
3741	1
3818	1
3819	1
3825	1
3905	2
3945	2
3958	1

Table 10: The new set of base locations in the RAV of Utrecht obtained after solving the MEXCLP without a specific set of base locations and the corresponding number of ambulances.

Base	Number of ambulances
3436	2
3447	2
3561	1
3582	2
3608	2
3645	2
3707	0
3811	2
3823	2
3911	2
3941	2

Table 11: The fixed set of base locations in the RAV of Utrecht and the corresponding number of ambulances.

E Results for different time standards

Base	Number of ambulances			
	$T = 8$	$T = 10$	$T = 12$	$T = 15$
3436	2	2	2	2
3447	2	1	2	2
3561	2	2	1	1
3582	3	2	2	2
3608	1	2	2	2
3645	1	2	2	2
3707	2	2	2	0
3811	2	2	2	2
3823	2	2	1	2
3911	1	1	2	2
3941	1	1	1	2

Table 12: The number of ambulances at each base location, after solving the MEXCLP using the fixed set of base locations, for different time thresholds T.

Base	Number of ambulances	Base	Number of ambulances	Base	Number of ambulances
3401	2	3604	1	3825	1
3417	1	3632	1	3833	1
3425	1	3641	2	3835	1
3431	1	3701	1	3903	1
3434	1	3702	1	3904	2
3439	1	3712	1	3947	1
3442	1	3723	1	3951	1
3445	1	3741	1	3953	1
3453	2	3752	1	3961	1
3522	1	3755	1	3985	1
3531	1	3761	1	3995	2
3534	1	3765	1	4131	1
3552	1	3813	1		
3573	2	3817	1		
3603	1	3824	1		

Table 13: Base locations and initial allocation of ambulances for $T = 8$.

Base	Number of ambulances	Base	Number of ambulances	Base	Number of ambulances
3405	1	3645	1	3921	1
3436	2	3709	1	3953	1
3444	2	3712	1	3962	1
3531	1	3741	2	3992	1
3544	1	3768	1	3994	1
3573	2	3823	1		
3603	1	3835	2		
3606	1	3904	1		
3643	1	3905	1		

Table 14: Base locations and initial allocation of ambulances for $T = 10$.

Base	Number of ambulances	Base	Number of ambulances	Base	Number of ambulances
3405	1	3645	2	3997	1
3433	1	3709	1		
3438	1	3712	1		
3442	1	3741	1		
3461	1	3742	1		
3541	1	3821	1		
3553	1	3832	1		
3573	1	3906	2		
3606	1	3947	1		

Table 15: Base locations and initial allocation of ambulances for $T = 12$.

F Algorithms for non-constant accident rate

Algorithm 8 Initialization routine

Require: Demand locations $i \in V$ and its demand d_i, demand fraction g_i, coordinates and whether also $i \in W$ or $i \in H$,
matrix of travel times between locations τ_{ij}, $i, j \in V$,
system state with number of ambulances with destination $j, \forall j \in W$,
number of vehicles used at a part of the day $|A_{part}|$, $part \in PartOfDay$,
busy fraction $q \in [0, 1]$,
accident rates λ_{part}, $part \in PartOfDay$,
driving time threshold T,
boolean value that determines static or dynamic policy,
total simulation hours u,
total simulation runs.

Initialize variables and statistical counters f.
for each simulation run **do**
    Invoke Algorithm 9.
end for
Print statiscal counters f.

Algorithm 9 Main event loop

Initialize current time t.
Initialize *part*, list of finish times, system state and queue.
Initialize statistical counters used within a run.
Generate next accident time for a location in V.
while $t < u$ **do**
    Invoke Algorithm 10.
    if $t > u$ **then**
        Do not count the next event.
    end if
    if next event is change in part of the day **then**
        Invoke Algorithm 11.
    else if next event is accident **then**
        Invoke Algorithm 12.
    else
        Invoke Algorithm 6.
    end if
end while
Update statistical counters f.

Algorithm 10 Timing routine

```
Store next accident time and finish times of all ambulances in a list.
Determine earliest time and update t.
Update part if t is at the next part of the day.
if part changes then
    Next event is part of the day change.
    Update t to time at which part changes.
else if earliest time is accident then
    Next event is accident.
else
    Next event is finish.
end if
for each idle ambulance still on the way a_onway do
    if t exceeds arrival time to base then
        Remove a_onway.
    end if
end for
```

Algorithm 11 Part of the day change event routine

```
Change the state space by adding or removing ambulances from base
locations, excluding the ambulances that are still on the way to a base
location.
Generate next accident time for a location in V.
```

Algorithm 12 Accident event routine

if number of busy ambulances equals or exceeds $|A_{part}|$ **then**
  Store accident time and location in queue.
else
  Initialize $ShortestTravelTime$ and $BestBase$.
  for each $j \in W$ **do**
    if $\tau_{ji} < ShortestTravelTime$ and j contains ambulances **then**
      $ShortestTravelTime = \tau_{ji}$ and $BestBase = j$.
    end if
  end for
  for each $a_{onway} \in IdleAmbulances \subseteq A$ **do**
    Determine current position.
    for each $k \in V$ **do**
      Calculate distance between k and current position.
      Store k if smallest distance.
    end for
    if $\tau_{ki} < ShortestTravelTime$ **then**
      $ShortestTravelTime = \tau_{ki}$ and $BestBase = k$.
    end if
  end for
  Update system state.
  Invoke Algorithm 7.
end if
Generate next accident time for a location in V.

References

[1] G.J. Kommer, S.L.N. Zwakhals, (2008) *Modellen referentiekader ambulancezorg 2008 : Documentatie rijtijden- en capaciteitsmodel*, Rijksinstituut voor Volksgezondheid en Milieu (RIVM), Rapport 270412001.

[2] C.J. Jagtenberg, S. Bhulai, R.D. Van der Mei, (2015) *An efficient heuristic for real-time ambulance redeployment.* Operations Research for Health Care 4: 27-35.

[3] Mark S. Daskin, (1983) *A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution.* Transportation Science 17(1): 48-70.

[4] Ambulancezorg Nederland, (2010) *Ambulances in-zicht 2010.* Zwolle: AZN, September 2007.