Erasmus University Rotterdam
Erasmus School of Economics

Dynamic Ambulance Redeployment and Ambulance Dispatching

Jessica van der Zee
Student ID: 383052

A thesis submitted for the degree of
Bachelor of Science in Econometrics
and Operations Research

Supervisor: R.B.O. Kerkkamp
Second assessor: W. van den Heuvel

July 4, 2016

ABSTRACT

In this thesis, we address the problems of ambulance dispatching and ambulance redeployment. That is, deciding which *ambulance* to send to an accident and choosing a *base* for the ambulance to return to after it finished service. The goal in these problems is to minimize the fraction of late arrivals. As an alternative to the well known closest-idle policy, we propose a dispatching policy that makes a weighted choice between distance to the accident and the coverage an idle ambulance currently provides. For the ambulance redeployment, we alter an existing dynamic solution that was already shown to improve performance compared to a static benchmark. We alter it in such a way that every time an ambulance becomes idle, a base is chosen by a trade-off between the coverage an extra ambulance at that base will provide and the distance from the idle ambulance to that base. We evaluate our performances by a simulation of a realistic case study in which we measure the fraction of late arrivals. We compare the performances to a benchmark that uses a static redeployment policy and the closest-idle policy. We show that our dispatching policy has a relative improvement of 9.4% compared to the benchmark. Furthermore the original redeployment policy has a relative improvement of 12.3% compared to the benchmark static solution. Our alteration to this policy does not improve the fraction of late arrivals, but lowers the fraction of time an ambulance spends on the road compared to the unaltered policy. Together the policies result in a relative improvement of 17.8% which can be achieved without costs for extra crew shifts or extra vehicles.

CONTENTS

1 INTRODUCTION AND RELATED WORK

There is a constant urge to improve the efficiency of the emergency medical services (EMS). One way of improving efficiency is ambulance redeployment: the action of relocating ambulances during the day. Another is dispatching: deciding which ambulance to send to an accident[1]. When an accident comes in, two decisions have to made. The first decision is *which* ambulance should be sent to the emergency site. In the unlikely event that there is no ambulance available, the accident is put in a first-come-first-served queue. When the ambulance crew arrived on scene and treated and transported the patient, one has to decide *where* to send the idle ambulance. If there are still unanswered accidents in the queue, the ambulance is sent to the first arrived accident in the queue. If the queue is empty, a designated waiting location (or base) for the ambulance to return to has to be chosen.

Over the past 45 years, ambulance deployment and redeployment models have developed. While during the early years static policies were most common, there has been a great development in the last twenty years in which dynamic models emerged.

Static models assign each ambulance to a home base, to which it is sent when it becomes idle. Early research focused on deterministic location problems, later the static policies were altered by focusing on stochastic location problems. One of these static policies is the maximum expected covering location problem (MEXCLP) formulation by Daskin in 1983 [2]. In this model it is assumed that each ambulance has the same busy fraction q, the probability of being unable to respond to an emergency call. The downside of these static models is that they do not take real-time information into account.

Dynamic models are more recent. They consider the possibility to relocate the ambulances during the day. One of these models is presented by Jagtenberg et al. in 2015 [1]. Their relocation policy allows for ambulance redeployment during the day with the restriction that an ambulance can only be relocated the moment it becomes idle (which is at the accident scene or at the hospital). In this way the amount of relocations stays similar compared to static models. Jagtenberg presents an algorithm to decide to which base the ambulance is sent. This algorithm determines the base which has the highest benefit in terms of the marginal coverage according to the MEXCLP model.

The most commonly used dispatching policy is the closest-idle policy; simply sending the closest idle ambulance to an accident. This was already shown to be suboptimal by Carter et al. in 1972 [3] in terms of the fraction of late arrivals. While there has been research in allowing increased response times to non-urgent accidents (see for example [4]), this is not relevant for this thesis since we only focus on high-priority calls and make no distinction in urgency.

In these ambulance dispatching, location and relocation problems, the most commonly used objective is maximizing the fraction of accidents with a response time below a certain threshold, where response time is defined as the time from the emergency call until an ambulance arrives on scene. Often governments pose certain restrictions on this fraction and threshold. For ex-

1 In this thesis we refer to an accident as the demand for an ambulance.

ample in the Netherlands, where the response time for 95% of the accidents must be within fifteen minutes. Therefore the fraction of late arrivals seems like a reasonable performance indicator. On the other hand, this indicator does not take into account the actual response time of the ambulance when it was on time or too late. Obviously an arrival which is for example half an hour too late is less desired than an arrival which is only one second too late.

In this thesis we propose an alteration to the dynamic redeployment algorithm from Jagtenberg et al. and develop a dispatching policy different from the closest-idle policy. Therefore we first formulate the relocation problem and the dispatching problem in Section 2. Next, in Section 3 we propose our solution methods to these problems. The performances are evaluated by a simulation of a realistic case study which we compare to a benchmark in Section 4. We conclude our findings in Section 5.

2 PROBLEM FORMULATION

In this section, we describe the real-time ambulance relocation problem as well as the dispatching problem. First we introduce some notation for these problems. Let V be the discrete set of demand points at which an accident can occur. Accidents occur at these demand points according to a Poisson process with rate $\lambda > 0$. Let each demand point in V have demand fraction $d_i \in [0, 1]$, then accidents occur at demand point i with rate $d_i\lambda$ with $i \in V$. We define the set of bases as the set $W \subseteq V$ and the set of hospitals as $H \subseteq V$. The set of ambulances is defined as A.

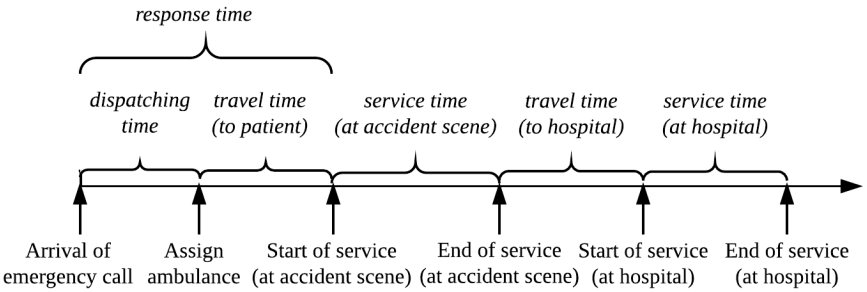

Figure 1: Graphical representation of dispatching and relocation process.

When an emergency call comes is, a chain of events is set in motion. This is visualized in Figure 1. When a call arrives, the dispatcher decides which idle ambulance to assign according to the dispatching policy. Next, the assigned ambulance travels to the accident scene. We assume the travel time with siren on $\tau_{ij} \geq 0$ between locations i and j to be deterministic ($i, j \in V$). Two different travel times are used: when the siren is turned on, the travel time is 0.9 times the travel time when the siren is turned off. The response time is compared with a threshold $T \geq 0$. When the ambulance has arrived, it takes an amount of time $\tau_{onscene} \geq 0$ to treat the patient which we assume to be deterministic. When the service on scene is finished, the decision is made whether the patient needs to go to hospital. If so, the ambulance transports the patient to the closest hospital. Upon arrival at the hospital, a certain amount of time $\tau_{hospital} \geq 0$, which we also assume to be deterministic, is needed after which the service at hospital is finished and the ambulance becomes idle. If there are no accidents in the queue, the ambulance becomes idle and is sent to a base according to the redeployment policy. When there are accidents in the queue the moment an ambulance finishes service, the ambulance is sent to the first accident in queue.

Note that we only allow for an ambulance to relocate the moment it becomes idle. This might seem restrictive, but especially in urban areas, ambulances become idle quite often and this allows for enough freedom in relocating. Second, no extra trips are made by imposing this restriction. This means that the workload and travel costs stay more or less the same compared to a static policy.

2.1 Ambulance redeployment

We redeploy ambulances when they become idle. Static models let ambulances always return to their home base while dynamic models use real-time information to decide to which base the ambulance is sent. Most dynamic models tend to use a rather elaborate state description, while the algorithm

of Jagtenberg et al. [1] is unique in the way that it only uses a very simple state space: the state space is defined as the destinations of all idle ambulances. When an idle ambulance is waiting at a base to be dispatched, its destination is simply its current location. This results in a relatively small state space. Since we consider all ambulances identical, it is sufficient to model the state as the number of idle vehicles n_j heading to each base for $j \in W$. We define the state space $\mathcal{S}$ as the set of states $s = \{n_1, \ldots, n_{|W|}\}$ where $n_j \in \mathbb{N}$ for $j = 1, \ldots, |W|$. We also define an action space $\mathcal{A}$ where the action is sending the newly available ambulance to a specific base. We define the policy $\pi \in \Pi$ as a mapping $\mathcal{S} \to \mathcal{A}$. So a policy defines a base for every possible state. That is, for every possible distribution of idle ambulances.

We choose a policy that minimizes the fraction of response times above the threshold T. Since we can order the accidents by their arrival times, the objective is

$$\arg\min_{\pi \in \Pi} \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{(R_i^\pi > T)},$$

where $\mathbb{I}$ is the indicator function and R_i^π is the response time for accident i under policy π. In Section 3 we introduce our solution procedure in order to minimize this objective.

2.2 Ambulance dispatching

When an emergency call comes in, one has to decide which ambulance to send to the accident. We use the same state space $\mathcal{S}$ as for ambulance redeployment, since both the dispatching and the redeployment are done by the same person and it makes sense to use the same information for both decisions. We define the action space $\mathcal{C}$, where the action is dispatching a specific ambulance to the accident. Then we define a policy $\gamma \in \Gamma$ as a mapping $\mathcal{S} \to \mathcal{C}$.

Next, we choose a policy to minimize the fraction of late arrivals. As we discuss later in this report, the fraction of late arrivals is not the only interesting criterion to evaluate. The objective is

$$\arg\min_{\gamma \in \Gamma} \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{(R_i^\gamma > T)}.$$

In the next section, we propose a solution method to this problem.

3 SOLUTION PROCEDURE

In this section we first discuss the static MEXCLP solution for ambulance redeployment as well as the algorithm of Jagtenberg et al. [1]. After that, we propose a modification to the method of Jagtenberg et al. Finally, a new dispatching method is presented.

3.1 Ambulance redeployment policy

We search for an ambulance redeployment policy that minimizes the fraction of late arrivals. It is intuitively clear that the fraction of late arrivals depends on the coverage the idle ambulances provide; the higher the coverage, the lower the fraction of late arrivals. Therefore it is sufficient to develop a policy that maximizes the coverage. There are many different definitions of coverage, we use the definition of the static MEXCLP model.

3.1.1 *Static MEXCLP model*

The maximum expected covering location problem (MEXCLP) model from Daskin [2] searches for the best static location policy using linear integer programming. There is a limited number of ambulances $p \geq 0$ and for each ambulance a home base is determined. A busy fraction $q \in (0,1)$ is assumed, which is equal for all ambulances. This fraction is determined by dividing the expected load of the system by the number of ambulances. We can express the expected covered demand at point $i \in V$ as $E_k = d_i(1-q^k)$ when point i is in range of k idle ambulances. This results in the marginal coverage of the k^{th} ambulance of $E_k - E_{k-1} = d_i(1-q)q^{k-1}$.

We introduce the binary variable y_{ik} which is equal to one if and only if demand point $i \in V$ is in range of at least k ambulances with $k = 1, \ldots, p$. Next the decision variables x_j represents the number of ambulances assigned to each base $j \in W$. Finally, let W_i be the set of bases that are within range of demand point i, so $W_i = \{j \in W : \tau_{ij} \leq T\}$. Recall that the goal of this policy is to maximize the coverage. The formulation of the MEXCLP model is

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in V} \sum_{k=1}^{p} d_i(1-q)q^{k-1} y_{ik} \\
\text{subject to} \quad & \sum_{j \in W_i} x_j \geq \sum_{k=1}^{p} y_{ik} && i \in V \\
& \sum_{j \in W} x_j \leq p \\
& x_j \in \mathbb{N} && j \in W \\
& y_{ik} \in \mathbb{B} && i \in V,\ k = 1, \ldots, p.
\end{aligned}
$$

In our dynamic redeployment policy we use the specification of the MEXCLP model for marginal coverage ($E_k - E_{k-1} = d_i(1-q)q^{k-1}$).

3.1.2 *Dynamic MEXCLP solution*

A redeployment policy π that maximizes the coverage is wanted, while only making use of the state space as described in Section 2. So when an ambulance becomes idle, the only information available is the set of destinations of all idle ambulances. The idea behind the algorithm of Jagtenberg et al. is as follows: whenever an ambulance becomes idle, we send the idle ambulance to the base at which the ambulance provides the best marginal coverage for future accidents. The base is chosen by calculating the marginal

coverage one extra ambulance would give for each base and choosing the base with the highest marginal coverage. When calculating the marginal coverage we only look at the idle ambulances and the coverage they provide. Furthermore, since we only know the destinations of the idle ambulances, we make no distinction whether or not they have arrived at their destinations when calculating the coverage. The algorithm can then be formulated as in Algorithm 1. In the following subsection we alter this policy.

Algorithm 1: Dynamic ambulance redeployment [1]

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 $a \in IdleAmbulances \subseteq A$,
ambulance $a^* \notin IdleAmbulances$ that will become available,
travel times τ_{ij} between any $i, j \in V$,
time threshold T to reach an emergency call.
Result: Destination for the ambulance that becomes idle

```
BestImprovement = 0;
BestLocation = NULL;
foreach j ∈ W do
    CoverageImprovement = 0;
    foreach i ∈ V do
        k = 0;
        if τ_ji ≤ T then
            k++;
            foreach a ∈ IdleAmbulances do
                if τ_dest(a)i ≤ 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
```

3.1.3 *Dynamic MEXCLP with travel times*

A possible flaw of the current algorithm is that the travel time of the trip from the current location of the ambulance to the base it is sent to is not taken into account. This might not be desired since an ambulance can be sent to the other side of the region which means higher transport cost and longer driving times. Also when the travel time between ambulance and base is high, the ambulance might not arrive at the base before being dispatched again. Our state space does not take into account that an ambulance may not have arrived yet, so when an ambulance has to travel far, the state does not represent reality as good as when travel times are shorter. So instead of simply choosing the base with the highest marginal coverage, we also consider the travel time to each base. Note that we do not consider the actual distance between two location points, but the travel time is approximately proportional with the distance.

To illustrate our method we first consider the previous situation where we choose the base with the highest marginal coverage [1]. The decision problem for ambulance $a \in A$ is

$$\arg\max_{j \in W} \{cov_j\},$$

where cov_j is the marginal coverage improvement of base j as calculated in Algorithm 1.

To take the travel time into account as well as the coverage, we first scale both expressions so they are both in the range between zero and one. The minimum marginal coverage improvement is equal to zero so we divide the coverage by the maximum marginal coverage. The new expression is $cov_j^* = \frac{cov_j}{u_{cov}}$ where u_{cov} is the upperbound of the marginal coverage. This upperbound can be mathematically determined: recall that the marginal coverage for an ambulance at a base $j \in W$ is

$$cov_j = \sum_{i \in V_j} d_i (1-q) q^{k_i - 1},$$

where V_j is the set of demand points in range of location j, so $V_j = \{i \in V : \tau_{ji} \leq T\}$, and k_i is the amount of idle ambulances in range of demand point i. The marginal coverage is maximal, when k_i is minimal. So to determine the maximum we take $k_i = 1$ for all $i \in V_j$. Then the marginal coverage is maximal for the base j for which $\sum_{i \in V_j} d_i$ is maximal. With an upperbound of $u_{cov} = (1-q) \max_{j \in W} \sum_{i \in V_j} d_i$, the scaled coverage is

$$cov_j^* = \frac{cov_j}{(1-q) \max_{j \in W} \sum_{i \in V_j} d_i}. \tag{1}$$

Next, we scale the travel times. For the travel times we make the modification

$$\tau^*_{loc(a)j} = 1 - \frac{\tau_{loc(a)j}}{\max_{i \in V, j \in W} \tau_{ij}},$$

where $loc(a)$ is the location of ambulance a. Note that we divide by the maximum travel time between a demand point and a base to scale the expression. Next, we subtract it from one since a higher travel time is less desired than a low travel time and we are maximizing the objective. As both travel time and coverage are now on the same range, we solve the following problem for ambulance $a \in A$ and $\theta \in [0, 1]$:

$$\arg\max_{j \in W} \{\theta \; cov_j^* + (1-\theta) \; \tau^*_{loc(a)j}\}.$$

The parameter θ is determined by a parameter search. Setting $\theta = 1$, means the travel time is not taken into account and the problem remains the same as before. We continue by proposing an alternative dispatching policy.

3.2 Ambulance dispatching policy

The most commonly used dispatching policy is the closest-idle policy. With this policy the closest idle ambulance is sent to an accident. You can imagine that, although sending the closest ambulance is best for the emergency the ambulance is sent to, sending another ambulance might result in a better coverage for future emergencies. So instead of sending the closest ambulance, one could also send for example the ambulance which provides the

least marginal coverage. Of course we do not want an ambulance to arrive late at the accident scene just because it has a smaller marginal coverage while another ambulance could have been on time. Therefore we restrict our decision to all ambulances which would arrive at the accident scene on time. When there are no ambulances that would arrive on time, we send the closest ambulance.

For each incoming accident we have to make a trade-off between future coverage and the response time for the incoming accident. It is obvious that for an ambulance, which is not the closest and the marginal coverage is only slightly lower, but the response time is a lot higher, the closest ambulance is still preferred. So we propose a method quite similar to what we have seen in the previous section. First, we formulate the closest-idle policy. Let $A^* \subseteq A$ be the set of idle ambulances. Then the formulation for the accident at location $l \in V$ is

$$\arg\min_{a \in A^*} \{\tau_{loc(a)l}\}. \tag{2}$$

We alter the objective function so that the marginal coverage of ambulance a is taken into account. To compare travel time and coverage we first scale both expressions so they are in the same range from zero to one. Note that the lower the marginal coverage, the more likely the ambulance is chosen. The scaled travel time is

$$\tau^{**}_{loc(a)l} = \frac{\tau_{loc(a)l}}{\max\limits_{i \in V, j \in W} \tau_{ij}} \left(= 1 - \tau^{*}_{loc(a)l}\right).$$

Again the marginal coverage $cov_{loc(a)}$ of an idle ambulance a can be scaled as in (1). We define the set A^*_l to be the set of idle ambulances that is within range of the accident at location l, so $A^*_l = \{a \in A^* : \tau_{loc(a)l} \leq T\}$. Then the formulation of the new MEXCLP dispatching policy for the accident at location $l \in V$ and $\eta \in [0, 1]$ is

$$\arg\min_{a \in A^*_l} \{\eta\ \tau^{**}_{loc(a)l} + (1 - \eta) cov^{*}_{loc(a)}\}. \tag{3}$$

Note that we only use this objective (3) when there is at least one idle ambulance in range of the accident ($A^*_l \neq \emptyset$). When there are no idle ambulances in range ($A^*_l = \emptyset$), we choose the closest idle ambulance as in (2). We do a parameter search for the best η. Setting $\eta = 1$, means this MEXCLP dispatching method is similar to the closest-idle dispatching method. In the next section we evaluate our methods.

4 RESULTS

In order to compare the results of our policies explained in Section 3, we measure their performance using simulation. In our simulation model there are three types of events: the arrival of an emergency call, the arrival of an ambulance on scene and an ambulance becoming idle.

When an *emergency call arrives*, an idle ambulance is chosen and sent to the accident. Next, the event of the arrival of the ambulance on scene is scheduled. When there is no idle ambulance available, the accident is put in a first-come first-serve queue. Finally, the next accident is scheduled.

When an *ambulance arrives on scene*, it is decided if the patient needs to be transported to hospital. If so, the operating ambulance is sent to a hospital after treating the patient on scene. For simplicity, the patient is always transported to the closest hospital. Next, the event of the ambulance becoming idle is scheduled. This event happens when the ambulances finishes service on scene, if the patient does not need to go to hospital. If the patient needs to go to hospital, this event is scheduled when the ambulances finishes service at the hospital.

When an *ambulance becomes idle*, we first check whether there are still accidents in queue. If so, the ambulance is sent to the first accident in line and the arrival of the ambulance on scene is scheduled. If there are no more accidents in queue, a base is chosen for the ambulance to return to. No new events are scheduled.

4.1 The case study of RAV Utrecht

The problem instance we use to evaluate our policies is a region in the Netherlands called Utrecht. The Netherlands is split up in twenty-four ambulance care regions (in Dutch: *Regionale Ambulance Voorzieningen* or *RAV*). One of the largest is RAV Utrecht which has an area of almost 1400 square kilometers. In RAV Utrecht there are nineteen ambulance bases and eight hospitals[2]. We only focus on high priority emergency calls. Since 40% of all calls are high priority calls, we assume 40% of the total fleet of RAV Utrecht that consists of 46 ambulances is realistic to cover this area [6], which is 18 ambulances.

In the Netherlands governments pose restrictions on the percentage accidents that are not served within fifteen minutes. Since answering the call and assigning a vehicle takes about three minutes, we use a threshold T of twelve minutes. That is, when an arrival of an emergency call occurs, the call first needs to be processed which takes about three minutes, then a vehicle can be dispatched and should arrive within twelve minutes. On average 9.5 accidents happen per hour. So accidents happen with an average interarrival time of $\frac{60}{9.5}$ minutes, therefore we use a rate of $\lambda = \frac{1}{6.32}$ for the Poisson process.

The Dutch National Institute for Public Health and the Environment (RIVM [5]) provided us with deterministic estimations of the driving times τ_{ij} with the siren turned on between any pair of four-digit postal codes in Utrecht. So we let V consist of all four-digit postal codes in Utrecht. We choose the fraction of population as demand d_i, since the demand is approximately

2 We make a distinction between community hospitals, academic hospitals and polyclinics, for this case study we consider all community hospitals and academic hospitals in RAV Utrecht.

Table 1: Characteristics of the problem instance RAV Utrecht.

Parameter	*Magnitude*	*Clarification*
λ	1/6.32 min	Realistic for high priority emergency calls
V	217	All four-digit postal codes in Utrecht
W	19	Bases as in 2016
H	8	Hospitals as in 2016
A	18	Realistic amount of ambulances to cover high-priority demand
d_i		Fraction of inhabitants as in 2008
τ_{ij}		Deterministic driving times with siren turned on as in 2008 [5]
$\tau_{\text{on scene}}$	12 min	Realistic estimate of the service time on scene
τ_{hospital}	15 min	Realistic estimate of the service time in the hospital
h	0.735	Fraction of accidents that went to hospital in 2014 [6]
T	12 min	Realistic value for threshold [6]

proportional with the population. Although this might not be the actual distribution of demand (for example industrial areas have a higher demand during working hours), the population is known with great accuracy and gives a realistic setting. To determine whether or not a patient needs further service in hospital, we assume the probability $h \in [0, 1]$ that a patient needs to go to a hospital is equal for each patient. For a summary of characteristics of RAV Utrecht, see Table 1.

We simulate and evaluate five models of which an overview can be found in Table 2. For all models we use the static MEXCLP solution to initialize the location of the ambulances. To determine the location of an ambulance, we use linear interpolation between the origin and the destination and the driving time of the ambulance and determine the point in V closest to the interpolated location. In all simulation models, we use the random number generator by Mersenne Twister [7] to generate interarrival times for accidents. Furthermore, we use the same set of seeds for the random number generator to compare different models.

Table 2: Characteristics of the evaluated models.

Section	*Redeployment Policy*	*Dispatching Policy*
4.2	Static MEXCLP (benchmark)	Closest-idle (benchmark)
4.2	Dynamic MEXCLP	Closest-idle
4.3	Dynamic MEXCLP with travel times	Closest-idle
4.4	Static MEXCLP	MEXCLP dispatching
4.5	Dynamic MEXCLP	MEXCLP dispatching

4.2 Static and dynamic MEXCLP redeployment

A commonly used benchmark for dynamic redeployment models is the static MEXCLP model, which typically gives a good static policy. We compare the static MEXCLP model and the dynamic MEXCLP model from Jagtenberg et al. [1] in Table 3. The fraction of late arrivals decreases from 8.9% with the static model to 7.8% with the dynamic model. This is a significant relative decrease of 12.3% which can be achieved without extra costs for purchasing extra ambulances or extra crew shifts. The average response times are a little bit lower with the dynamic model compared to the static model. This is confirmed by the cumulative distribution function as in Figure 2. The figure shows that overall, the dynamic policy has lower response times.

Table 3: Comparison between the static MEXCLP model and the dynamic MEXCLP model, both policies are evaluated with 100 runs of 10.000 simulation hours and a value of $q = 0.3$ is used.

Performance	***Static***		***Dynamic***	
	Mean	*Std dev*	*Mean*	*Std dev*
Fraction of late arrivals	8.94%	0.001	7.84%	0.001
Fraction of time on the road	18.40%	0.001	26.49%	0.001
Average response time	10.1 min	0.019	9.9 min	0.016
Average on-time response time	9.3 min	0.011	9.2 min	0.011
Average late response time	18.6 min	0.051	18.4 min	0.042

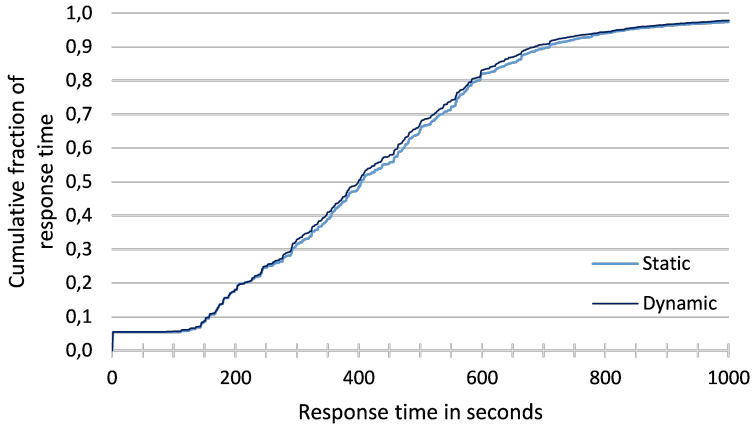

Figure 2: Response times for the static and dynamic MEXCLP redeployment policies, for both policies a value of $q = 0.3$ is used. Each policy is evaluated with 100 runs of 10.000 simulation hours.

As can be expected, the average fraction of time an ambulance spends on the road is a lot higher when using the dynamic MEXCLP policy; 18% and 26% for the static and dynamic policy respectively. This is also an incentive to the modification we presented previously in Section 3.1.3; taking the travel time between ambulance and base into account when redeploying.

We simulated the dynamic policy for different values of the busy fraction q. As can be seen in Table 4, we can conclude that for this problem instance the quality of the solution is insensitive to the value of q.

Table 4: Comparing the fraction of late arrivals of the dynamic MEXCLP model for different values of q, each policy is evaluated with 10 runs of 10.000 simulation hours.

q	0.1	0.2	0.3	0.4
Mean	8.09%	7.99%	7.85%	7.84%
Std dev	0.0006	0.0008	0.0010	0.0010

4.3 Dynamic MEXCLP redeployment with travel times

We evaluate the performance of our altered redeployment policy as described in Section 3.1.3, therefore we perform a parameter search for the best value of θ. For each value of θ, we check the fraction of late arrivals and the average fraction of time that an ambulance spends on the road. The results can be found in Figure 3. The figure shows that θ close to one results in the lowest fraction of late arrivals. As expected, when θ decreases, the average time on the road also decreases. If less time on the road is wanted, one could choose to use this policy with a value of θ a little below one. This way the fraction of late arrivals only increases negligibly while the time on the road decreases significantly. For example, setting $\theta = 0.95$ results in a significant relative improvement of 7% in time on the road, while the fraction of late arrivals increases with only 0.02 percentage point.

The performance of the policy was compared for different values of q, all values gave similar results as in Figure 3. That is, for all q setting $\theta = 1$ was best and the shape of the graphs was similar. We can conclude, that the solution of this policy is also insensitive to the value of q.

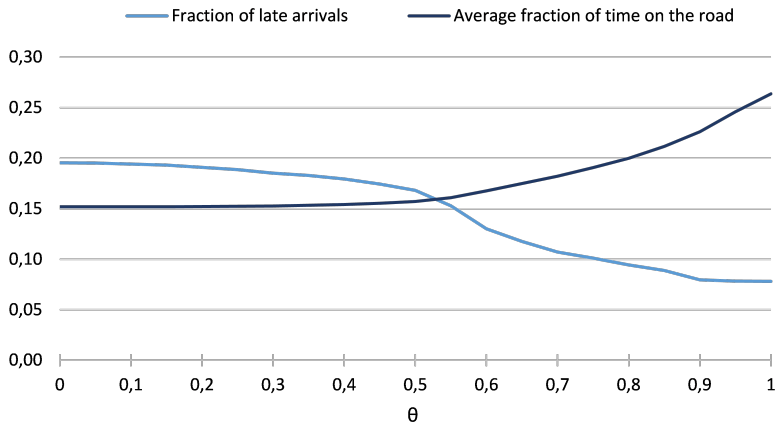

Figure 3: Comparison between fraction of late arrivals and percentage of time spent on the road. θ iterated from 0 to 1 with steps of 0.01, for each θ we ran the simulation for 10 runs of 1000 simulation hours and a value of $q = 0.3$ was used.

4.4 MEXCLP dispatching

To evaluate our MEXCLP dispatching policy, we use the closest-idle policy as a benchmark. In both models we use the static MEXCLP policy for redeployment. To determine the best η, we perform a parameter search as in Figure 4. These results show that by implementing the MEXCLP dispatching policy one can obtain a lower fraction of late arrivals compared to the closest-idle policy. A more detailed analysis shows that $\eta = 0.32$ is best for this problem instance with 8.1% late arrivals. As could be expected: although the fraction of late arrivals lowers, the average response time is higher. A comparison of the properties of both policies can be found in Table 5. The MEXCLP dispatching policy leads to a decrease in late arrivals of 0.84 percentage point which is a significant relative decrease of 9.4%. The average response time for on-time accidents increases from 9.3 minutes to 9.6 minutes which is a relative increase of 3.2%. This is confirmed in Figure 5.

Table 5: Comparison between closest-idle dispatching policy and the MEXCLP dispatching policy with $\eta = 0.32$, both use the static redeployment policy and are evaluated with 100 runs of 10.000 simulation hours and use a value of $q = 0.3$.

Performance	*Closest-idle*		*MEXCLP*	
	Mean	*Std dev*	*Mean*	*Std dev*
Fraction of late arrivals	8.94%	0.001	8.10%	0.001
Fraction of time on the road	18.40%	0.001	19.20%	0.001
Average response time	10.1 min	0.019	10.3 min	0.018
Average on-time response time	9.3 min	0.011	9.6 min	0.011
Average late response time	18.6 min	0.051	18.7 min	0.050

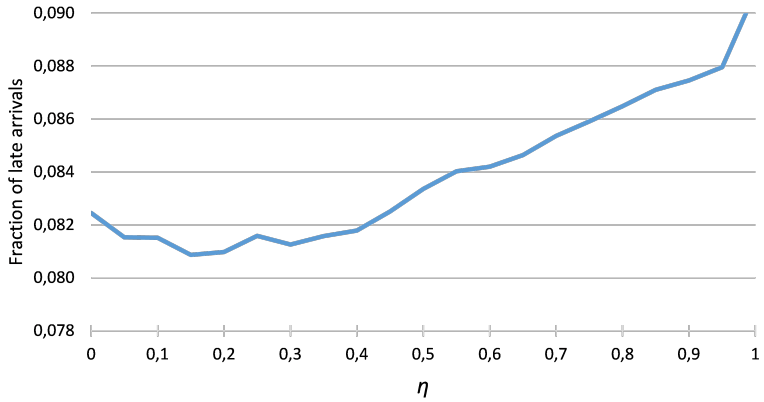

Figure 4: Fraction of late arrivals for different values of η which iterated from 0 to 1 with steps of 0.05, for each η we ran the simulation for 10 runs of 1000 simulation hours and a value of $q = 0.3$ was used.

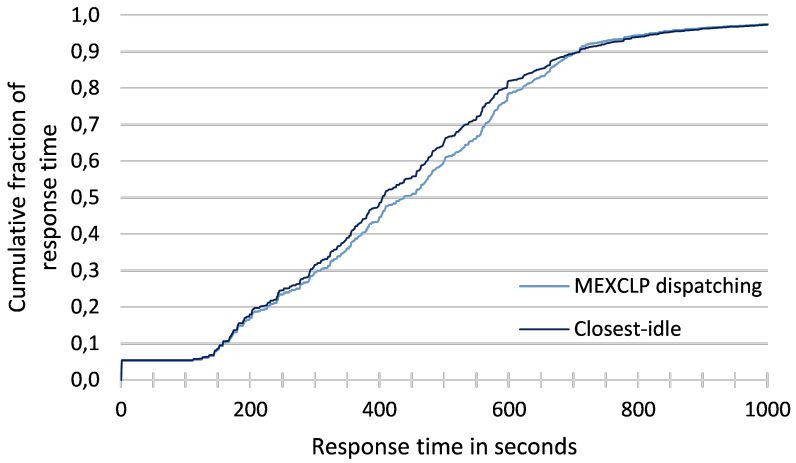

Figure 5: Response times for the closest-idle and MEXCLP dispatching policies, for both policies a value of $q = 0.3$ is used. Each policy was evaluated for 100 runs of 10.000 simulation hours.

We check if the quality of the solution is insensitive to the busy fraction q by simulating the model for different values of q as in Table 6. The results show that setting $q = 0.3$ is a good choice. Note that the values of q between 0.1 and 0.7 are quite extreme and small deviations of q result in negligible changes in the performance. Therefore we still conclude the quality of the solution is insensitive to q.

Table 6: Comparing the fraction of late arrivals of the MEXCLP dispatching model for different values of q, each policy is evaluated with 10 runs of 10.000 simulation hours.

q	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Mean	8.32%	8.15%	8.09%	8.14%	8.27%	8.51%	8.82%
Std dev	0.0013	0.0011	0.0013	0.0009	0.0009	0.0013	0.0011

4.5 MEXCLP redeployment and MEXCLP dispatching

Finally, we combine the redeployment policy as well as the MEXCLP dispatching policy. Since the altered MEXCLP redeployment policy did not improve the fraction of late arrivals, we use the (unaltered) MEXCLP redeployment policy. We compare this model to the benchmark that uses the static MEXCLP redeployment policy and the closest-idle dispatching policy. A detailed analysis showed that in this case setting $\eta = 0.27$ for the dispatching policy is best. Combined our policies reduce the fraction of late arrivals from 8.94% to 7.35% as can be seen in Table 7, which is a relative improvement of 17.8%.

Table 7: Comparison between the *benchmark* model that uses the static MEXCLP redeployment policy and the closest-idle dispatching policy and the *MEXCLP* model that uses the dynamic MEXCLP redeployment policy and the MEXCLP dispatching policy with $\eta = 0.27$, both are evaluated with 100 runs of 10.000 simulation hours and use a value of $q = 0.3$.

Performance	***Benchmark***		***MEXCLP***	
	Mean	*Std dev*	*Mean*	*Std dev*
Fraction of late arrivals	8.94%	0.001	7.35%	0.001
Fraction of time on the road	18.40%	0.001	27.28%	0.001
Average response time	10.1 min	0.019	10.3 min	0.018
Average on-time response time	9.3 min	0.011	9.6 min	0.013
Average late response time	18.6 min	0.051	18.5 min	0.039

5 CONCLUSION AND DISCUSSION

In this thesis, we developed a real-time ambulance redeployment policy and a dispatching policy, with the goal to minimize the fraction of late arrivals. The dynamic MEXCLP redeployment policy from Jagtenberg et al. [1] reduces the fraction of late arrivals by relatively 12.3% compared to a benchmark for static solutions, furthermore the overall response times are lowered. The quality of the solution is insensitive to the value of q. The redeployment algorithm does not take the travel time between the newly idle ambulance and base into account, but chooses a base only according to the coverage an extra ambulance at that base will provide. Our alteration to this policy makes a trade-off between coverage and travel time. It does not result in a lower fraction of late arrivals, but the time on the road can be significantly lowered with only a negligible increase in the fraction of late arrivals.

Typically the closest idle ambulance is dispatched to an accident. Our proposed MEXCLP dispatching policy makes a trade-off between travel time to the accident and the coverage an idle ambulance is providing. It reduces the fraction of late arrivals by relatively 9.4%, but the average response times are longer compared to the static benchmark that uses the closest-idle policy. The increase in response times is an important downside and a trade-off needs to be made between minimizing the objective and decreasing response times. The solution is also insensitive to the value of q.

Together the unaltered redeployment policy and the MEXCLP dispatching policy give a relative improvement of 17.8% compared to the static benchmark with the closest-idle policy. This result is achieved without extra costs for crew shifts or ambulances and without extensive state information.

The improvement of performance by the dynamic redeployment policy of Jagtenberg et al. can be obtained without any extra costs. Note that it might be hard for staff members to give up having a home base. EMS managers should make the trade-off between staff satisfaction and performance improvement. Also, the workload of staff increases since they will spend more time on the road when the dynamic policy is implemented. For this reason, we propose our altered dynamic MEXCLP policy which can lower the time on the road while still improving the fraction of late arrivals.

It is interesting to consider the applicability of our policies when we relax some of our assumptions. In practice, EMS systems may have different, more complicated, characteristics than assumed in this thesis. First of all, we assumed all parameters to be constant, while it might be more realistic to have changes during the day. For example driving times can be higher during rush hours or the demand can be higher in industrial areas during office hours. This is usually quite hard to incorporate in a solution, but in our algorithms you can simply change the relevant parameters over time. A second assumption we made is that the driving times and service times are deterministic, while it would be more realistic to consider them stochastic. Jagtenberg et al. proposed using the expected value of driving times, but it would require further research to check the validity of our methods.

Acknowledgements

I would like to thank my supervisor Rutger Kerkkamp for guiding me through the process of writing this thesis. I also thank the Dutch Public Ministry of Health (RIVM) for giving access to the travel times of ambulances in RAV Utrecht.

REFERENCES

[1] Jagtenberg, Bhulai, and Van der Mei (2015), 'An efficient heuristic for real-time ambulance redeployment', Operations Research for Health Care 4, 27-35.

[2] Daskin (1983), 'A maximum expected location model: Formulation, properties and heuristic solution', Transportation Science 7, 48–70.

[3] Carter, Chaiken and Ignall (1972), 'Response areas for two emergency units', Operations Research 20(3):571–594.

[4] McLay and Mayorga (2013), 'A dispatching model for server-to-customer systems that balances efficiency and equity', Manufacuring and Service Operations Management 15(2):205–220, 3.

[5] Kommer and Zwakhals (2011), 'Modellen referentiekader ambulancezorg 2008: Documentatie rijtijden- en capaciteitsmodel', Rapport 270412001/2011.

[6] Boers (2014), 'Ambulances in-zicht 2014', Ambulancezorg Nederland.

[7] Matsumoto and Nishimura (1998), 'Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator', ACM Trans. on Modeling and Computer Simulation Volume 8, Number 1, pp.3-30.

APPENDICES

LIST OF TABLES

LIST OF FIGURES

A PSEUDOCODE SIMULATION PROGRAM

Algorithm 2: Simulation

```
Input  : T and instance parameters
Output: Fraction of late arrivals
initialize numAccidents, numOnTimeAccidents and clock to zero
plan first accident arrival event and add it to the eventlist
while clock is below T do
    get next event E and remove it from the eventlist
    update clock to time of E
    if E is an accident arrival then
        execute AccidentArrivalEvent
    else if E is an ambulance arrival on scene then
        execute AmbulanceArrivalEvent
        numAccidents++
    else if E is an ambulance becoming idle then
        execute AmbulanceBecomingIdleEvent
    end
end
```

Algorithm 3: Accident arrival event

```
Input  : event, clock
determine location of event
get list L of idle ambulances at time clock
if L is empty then
    add event to queue
else
    find closest idle ambulance A at time clock
    let A go to the accident scene
    make A not idle
    schedule ambulance arrival event of A on scene
end
schedule next accident arrival
```

Algorithm 4: Ambulance arrival event

```
Input  : event E, clock, ambulance A
if A arrived on time at E then
    numOnTimeAccidents++
end
if patient needs to go to hospital then
    determine closest hospital H
    let A go to H after it finished service on scene
    schedule A becoming idle event at H for when A brought the
      patient to hospital and finished service at the hospital
else
    schedule A becoming idle event for when A finishes treating the
      patient on scene
end
```

Algorithm 5: Ambulance becoming idle event

Input : ambulance A, *clock*

if *queue is empty* **then**
    determine base B to send A to, by the redeployment policy
    let A go to B
    make A idle
else
    get first accident in line C and remove C from queue
    let A go to accident C
    schedule the ambulance arrival event of A at C
end

B ALTERED MEXCLP REDEPLOYMENT ALGORITHM

Algorithm 6: Altered dynamic MEXCLP redeployment [1]

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 $a \in IdleAmbulances \subseteq A$,
ambulance $a^* \notin IdleAmbulances$ that will become available at $location(a^*)$,
travel times τ_{ij} between any $i, j \in V$,
time threshold T to reach an emergency call,
parameter $\theta \in [0,1]$,
upperbound u_{cov} of CoverageImprovement,
upperbound u_{tt} of the travel time from base to demand point.
Result: Destination for the ambulance that becomes idle

```
BestImprovement = 0;
BestLocation = NULL;
foreach j ∈ W do
    CoverageImprovement = 0;
    foreach i ∈ V do
        k = 0;
        if τ_ji ≤ T then
            k++;
            foreach a ∈ IdleAmbulances do
                if τ_dest(a)i ≤ T then
                    k++;
                end
            end
            CoverageImprovement += d_i(1 − q)q^(k−1);
        end
    end
    Improvement = θ CoverageImprovement/u_cov + (1 − θ)(1 − τ_location(a*)j/u_tt);
    if Improvement > BestImprovement then
        BestLocation = j;
        BestImprovement = Improvement;
    end
end
```

C DATA FOR RAV UTRECHT

Table 8: Bases of RAV Utrecht with their four-digit postal codes.

Location of base	*Postal code*
Abcoude	1391
Amerongen	3958
Amersfoort Centrum	3811
Amersfoort Noord	3823
Baarn	3743
Doorn	3941
Houten	3991
Maarssen	3608
Montfoort	3417
Nieuwegein	3436
Rhenen	3911
Soesterberg	3769
Utrecht (Andreaelaan)	3582
Utrecht (Vader Rijndreef)	3561
Vinkeveen	3645
Wilnis	3648
Woerden	3447
Woudenberg	3931
Zeist	3707

Table 9: Hospitals in RAV Utrecht with their four-digit postal codes.

Hospital	*Postal code*
Diakonessenhuis Utrecht	3582
Diakonessenhuis Zeist	3707
Meander Medisch Centrum Amersfoort	3813
Meander Medisch Centrum Baarn	3743
St. Antonius Ziekenhuis Nieuwegein	3435
St. Antonius Ziekenhuis Utrecht	3543
Universitair Medisch Centrum Utrecht	3584
Zuwe Hofpoort Ziekenhuis	3447

Table 10: All four-digit postal codes (PC) in RAV Utrecht together with their population (Pop.) as in 2008.

PC	*Pop.*	*PC*	*Pop.*	*PC*	*Pop.*	*PC*	*Pop.*
1391	7435	3525	5700	3646	805	3826	3470
1393	1590	3526	9785	3648	6545	3828	10820
1396	1330	3527	11825	3701	5700	3829	2900
1426	905	3528	10	3702	3830	3831	12815
1427	965	3531	11350	3703	5890	3832	3800
3401	12555	3532	6915	3704	9110	3833	7350
3402	9240	3533	7365	3705	9345	3834	2050
3403	2285	3534	1280	3706	5200	3835	350
3404	10140	3541	15	3707	5815	3836	135
3405	3455	3542	205	3708	5910	3901	9070
3411	8090	3543	7105	3709	320	3902	10260
3412	1110	3544	16705	3711	1460	3903	7125
3413	320	3545	135	3712	2035	3904	13510
3415	1200	3546	215	3721	7620	3905	14315
3417	9705	3551	6940	3722	4465	3906	7595
3421	7875	3552	6220	3723	9935	3907	165
3425	850	3553	7180	3731	5955	3911	13760
3431	7240	3554	7355	3732	4510	3912	660
3432	5795	3555	7705	3734	3940	3921	4065
3433	3490	3561	8755	3735	1730	3922	365
3434	8520	3562	6670	3737	1890	3927	4600
3435	6095	3563	6230	3738	4975	3931	11895
3436	6530	3564	9530	3739	1445	3941	10050
3437	13665	3565	65	3741	8885	3945	3110
3438	9260	3566	335	3742	9660	3947	2060
3439	300	3571	9590	3743	4410	3951	4650
3441	1925	3572	11110	3744	1080	3953	1310
3442	3930	3573	3205	3749	285	3956	7555
3443	7035	3581	9630	3751	6095	3958	5475
3444	235	3582	8505	3752	13035	3959	1355
3445	3985	3583	5885	3754	865	3961	9640
3446	7725	3584	5440	3755	8835	3962	8350
3447	15	3585	180	3761	4905	3971	7915
3448	9565	3601	6980	3762	8405	3972	10500
3449	385	3602	3610	3763	2565	3981	6635
3451	9245	3603	2365	3764	5190	3984	5460
3452	9280	3604	1510	3765	5040	3985	2360
3453	9255	3605	5275	3766	7460	3989	10
3454	11680	3606	30	3768	5805	3991	17125
3455	420	3607	14125	3769	6300	3992	6665
3461	3785	3608	3370	3791	2435	3993	5390
3464	395	3611	545	3811	5920	3994	13620
3467	720	3612	1190	3812	10780	3995	1560
3471	3810	3615	1220	3813	15450	3997	630
3474	2345	3621	10610	3814	5830	3998	1935
3481	8370	3626	730	3815	10455	3999	690
3511	8535	3628	3265	3816	11185	4121	1240
3512	7820	3631	455	3817	11885	4122	810
3513	5315	3632	4200	3818	10120	4124	1480
3514	7360	3633	1745	3819	430	4131	4220
3515	4765	3634	515	3821	195	4132	4655
3521	5360	3641	11625	3822	8555	4133	7235
3522	8530	3642	3410	3823	11850		
3523	8770	3643	1665	3824	15275		
3524	11600	3645	8465	3825	9595		