EFFICIENCY AND FAIRNESS
IN AMBULANCE PLANNING

Caroline Jagtenberg
An efficient heuristic for real-time ambulance redeployment

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

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

This chapter is based on:

4.1 Introduction

This chapter considers the problem of dynamic ambulance repositioning, also known as redeployment or move-up: proactively relocating idle vehicles in order to reduce response times. The general idea is that the idle vehicles should be relocated to compensate for other ambulances that are busy and hence temporarily unavailable to respond to incidents. Decisions on how to redeploy the vehicles are to be made in real time, and may take into account the status of all other vehicles and incidents.
A variety of techniques has been used to tackle this problem, a summary of which can be found in Section 1.2.2 and in [11]. The randomness in the EMS system combined with a large state space make this problem difficult: while exact models can be solved for small problem instances, realistically-sized EMS regions require an approximation. Such approximations typically include simplifying assumptions. For example, some of the models in literature assume that an incident is served late if there are no idle vehicles present at the nearest base (e.g., [78]). This particular assumption would make a model unsuitable for the EMS region that we have in mind: it includes demand points that can be reached within the time threshold from as many as eight different bases. Despite using simplifying assumptions, some of the existing approximations are in fact not all that simple: they are still computationally heavy and require an expert to implement them.

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

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

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

## 4.2 Problem formulation

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>The set of ambulances.</td>
</tr>
<tr>
<td>$V$</td>
<td>The set of demand locations.</td>
</tr>
<tr>
<td>$H$</td>
<td>The set of hospital locations, $H \subseteq V$.</td>
</tr>
<tr>
<td>$W$</td>
<td>The set of base locations, $W \subseteq V$.</td>
</tr>
<tr>
<td>$T$</td>
<td>The time threshold.</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Incident rate.</td>
</tr>
<tr>
<td>$d_i$</td>
<td>The fraction of demand in $i$, $i \in V$.</td>
</tr>
<tr>
<td>$\tau_{ij}$</td>
<td>The driving time between $i$ and $j$ with siren turned on, $i, j \in V$.</td>
</tr>
<tr>
<td>$n_i$</td>
<td>The number of idle ambulances that have destination $i$, $i \in W$.</td>
</tr>
</tbody>
</table>

Table 4.1  Notation.

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

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

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

\(^1\)Our model uses two different travel speeds. If the ambulance is traveling without siren, its travel speed is 0.9 times the travel speed when it is traveling towards an incident scene.
4.2.1 State space and policy definition

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

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

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

4.2.2 Objective

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

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

(4.1)

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

4.3 Algorithm

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

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

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

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

**Myopic solution**

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

Dynamic MEXCLP solution

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

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

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

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

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

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

4.3.1 Limitations

As described in Section 4.2.1, our state space definition prohibits the ambulance relocation problem from being solved to optimality. But even within our state space, the Dynamic MEXLP model need not lead to optimal decisions. The definition of (marginal) coverage as given by the MEXCLP model has some well-known imperfections. For example, vehicles are assumed to operate independently, and the busy fraction is assumed to be the same for all vehicles. These limitations also transfer to the dynamic usage of (MEXCLP) coverage. Therefore, our proposed solution must be a heuristic one, and we do not claim to have solved the problem in an exact manner. However, heuristic policies are common
**Data:** The demand \( d_i \) per node \( i \in V \), base locations \( W \subseteq V \), busy fraction \( q \in [0,1] \), current destinations \( \text{dest}(a) \) for all \( a \in \text{IdleAmbulances} \subseteq A \), travel times \( \tau_{ij} \) between any \( i, j \in V \), time threshold \( T \) to reach an emergency call.

**Result:** A new destination for the ambulance that is about to become idle

BestImprovement = 0
BestLocation = NULL

**Algorithm 1:** Dynamic MEXCLP

```
foreach \( j \) in \( W \) do
    CoverageImprovement = 0
    foreach \( i \) in \( V \) do
        \( k = 0 \)
        if \( \tau_{ji} \leq T \) then
            \( k++ \)
            foreach \( a \) in \( \text{IdleAmbulances} \) do
                if \( \tau_{\text{dest}(a)i} \leq T \) then
                    \( k++ \)
                end
            end
        end
        CoverageImprovement += \( d_i (1 - q) q^{k-1} \)
    end
    if \( \text{CoverageImprovement} > \text{BestImprovement} \) then
        BestLocation = \( j \)
        BestImprovement = \( \text{CoverageImprovement} \)
    end
end
return BestLocation
```
Chapter 4. Real-time ambulance redeployment

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

4.3.2 Computation time

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

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

4.4 Computational results

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

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

In our simulation, \( \tau_{\text{onscene}} \) is exponentially distributed with an expectation of 12 minutes. \( \tau_{\text{hospital}} \) is drawn from a Weibull distribution with an expectation of 15 minutes. More specifically, it has shape parameter 1.5 and scale parameter 18 (in minutes). We state these distributions for completeness, however, numerical experiments (done by the authors in ongoing work) indicate that the performance does not depend much on the chosen distribution for \( \tau_{\text{onscene}} \) or \( \tau_{\text{hospital}} \). In our simulations, patients need hospital treatment with probability 0.8. This value
was estimated from Dutch data [112]. Similar numbers (78% nation-wide) can be deduced from [89].

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

### 4.4.1 A small region

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

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

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

---

2Recall that the ambulance might not arrive at this base location, because it may be dispatched before reaching its destination.
Chapter 4. Real-time ambulance redeployment

<table>
<thead>
<tr>
<th>i</th>
<th>( d_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>City 1</td>
<td>0.2</td>
</tr>
<tr>
<td>City 2</td>
<td>0.4</td>
</tr>
<tr>
<td>City 3</td>
<td>0.2</td>
</tr>
<tr>
<td>Town 1</td>
<td>0.07</td>
</tr>
<tr>
<td>Town 2</td>
<td>0.07</td>
</tr>
<tr>
<td>Town 3</td>
<td>0.06</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.2 Distribution of demand in a small region.

serve the incidents in this region.

Static policies
Let us consider static policies first. We have nine nodes and give vehicles available. If vehicles were distinguishable, this would mean there are \( 9^5 = 59,049 \) different static policies. Instead, we assume vehicles are indistinguishable, which makes the set of truly different policies smaller. If we number the nodes 1 up until 9, we can describe a policy by a five tuple of non-decreasing integers, representing the home locations of the five vehicles. For example, \((2,2,5,8,9)\) denotes a policy, but \((5,6,3,1,9)\) does not. Using this definition, we can iterate over all static policies. This allows us to take a closer look at the static solution space.

Finding the optimal solution for a discrete event dynamic system (DEDS) is in general difficult due to the large search space and the simulation-based performance evaluation. Inspired by Ordinal Optimization (see, for example, [70] or [104]), which has become an important tool for optimizing DEDSs, we create an Ordered Performance Curve (OPC) as follows. For each policy, we simulate the EMS region for an amount of time, and use the measured fraction of late arrivals as an estimate for the true performance of the policy. Then, we sort the policies by their estimated performance, giving us the desired OPC. At first, we look into the case where there are relatively few incidents, i.e., \( \lambda = 1/45 \) (per minute). In this case, we evaluate each policy with 10 simulated days. For the corresponding OPC, see Figure 4.2a. According to the theory of Ordinal Optimization, the shape of this OPC indicates that there are many good solutions (policies) for this problem [70].

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

\[\text{We start with an empty system, i.e., no incidents have occurred. Therefore, we need to allow the system some time to evolve towards a more natural and representative state. We disregard the first five simulated hours in each run, and only consider the performance of the remaining time.}\]
the shape of the OPC. For $\lambda = 1/13$, the OPC is shown in Figure 4.2b. For this case, we evaluate each policy with 2.9 simulated days, which boils down to the same expected number of incidents per evaluation as in the $\lambda = 1/45$ case. First of all, note that the best static solution for this problem seems to have a performance of 17% (compared to 1% in Figure 4.2a). An increase was to be expected, because the same number of vehicles needs to serve a higher number of incidents. Perhaps more surprising is that also the shape of the OPC has changed. For Figure 4.2b, the OPC indicates that there exist only a few good static policies for this problem.

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

![Figure 4.2 OPC curves for static policies in the same region, for two different incident intensities.](image)

**DMEXCLP versus the best static policy**

We now compare the performance of dynamic MEXCLP (DMEXCLP) with the best static policy. We will test our method on multiple scenarios, to show that the method gives good results for more than just one specific problem instance. We create different problem instances by changing the value of $\lambda$. Since we keep the number of vehicles equal to five, by varying $\lambda$ we also vary the load of the system. In Figure 4.3, it shows that the DMEXCLP policy outperforms the best static policy for every choice of $\lambda$. When we let $\lambda$ take even more extreme values, we see that DMEXCLP has approximately the same performance as the best static solution. This occurs when $\lambda = 1/9$, in which case the expected fraction of late arrivals for both the best static and the DMEXCLP solution is around 67%. A fraction this high will never be acceptable in real life, and would indicate that more vehicles are needed. Therefore, we should not draw conclusions on the applicability based on this parameter choice. Note that, even if the performance of DMEXCLP is equal to the performance of the best static policy, DMEXCLP
Chapter 4. Real-time ambulance redeployment

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

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

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

4.4.2 A realistic case study

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

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

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

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

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

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

Results

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

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

![Figure 4.5](image)

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

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

### 4.4.3 Sensitivity to the busy fraction

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

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

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

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

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

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

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

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

Changes during the day

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

- Incident probabilities may shift, for example, an incident is more likely to occur in an industrial area during office hours.

- Travel times may be longer in rush hour, or may depend on the weather.

Changing parameters over time, such as the examples above, are often difficult to incorporate in a solution. However, in our case, there is no need to complicate the algorithm. At any decision epoch, a new set of parameters could be used. The question remains how to choose relevant parameters. One should keep in mind that there is only a limited number of decision epochs. Hence, a redeployment decision should not necessarily use the parameters of the system at the exact decision moment, but parameters that are relevant for the upcoming period. The choice of the period size may depend on the EMS region, but for example 30 minutes would be a good starting point.
Stochastic travel times
One straightforward way of dealing with stochastic travel times is to use the expectations $E[\tau_{ij}]$ in Algorithm 1. Alternatively, one could use for example the 0.8 quantile of the driving time distribution, i.e., the number $X_{ij}$ such that $P[\tau_{ij} \leq X_{ij}] = 0.8$. This showed to give a good performance in some additional numerical work that we performed. The performance will generally depend on the exact distribution function chosen, and we suggest some preliminary experiments to obtain a good strategy.

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

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

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

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

4.6 Implementation in practice

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

![Figure 4.8 A screenshot of the pilot in Flevoland.](image)

In collaboration with the EMS managers of GGD Flevoland [43], some practical adaptations were made to the DMEXCLP algorithm. Due to the low incident frequency in the region, we created extra decision moments: a relocation was allowed whenever a vehicle became idle or busy. Furthermore, some moves were rather long trips, taking half an hour or more. We decided to split those travels in two, if possible, by repositioning two vehicles that resulted in the same net move. Note that this is beneficial for the system because the desired configuration is reached quicker.
The dispatchers generally followed the relocation advice, unless they had good arguments not to. In the initial phase, we discussed all situations for which the dispatchers disagreed. This lead to interesting conversations and new insights on both ends. In some cases, the situation was simply too complex for a single person to oversee, and the system made better decisions than the dispatchers. In other cases, the dispatchers were right, and often this was because they were able to use more information than our software tool.

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

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

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

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

Currently, the software is developed further in order to add more practical features. For example, the software will include information about the shift start- and end times, such that it can steer ambulances towards their finish location when the end of their shift is approaching. Furthermore, ambulances should not be sent towards a base where a new shift will start shortly. These developments are being done under the name of Stokhos B.V. [108]: a spin-off company founded after the success of the Flevoland pilot.
Figure 4.9 Feedback during the pilot: EMS dispatcher Annemieke thinks the DMEXCLP algorithm is very pleasant to work with.