Simulation Optimisation and Markov Models for Dynamic Ambulance Redeployment

Lei Zhang

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering Science

The University of Auckland

2012

Abstract

The study of dynamic ambulance redeployment, also known as move-up or system status management, is the main concern of this investigation. Move-up is a practice of dynamically deciding stand-by locations for free ambulances in attempt to achieve quick response times.

In the first part of the investigation, we study optimal move-up policies based on three small-scale Markov models to gain insights. The first Markov model considers one ambulance and aims to maximise the benefit of move-up for just the next call. The second Markov model still considers one ambulance, but aims to maximise the average benefit per unit time over an infinite horizon. The third Markov model extends the second Markov model by considering two ambulances. Numerical experiments are used to gain insights into optimal move-up policies based on the three models.

In the second part of the investigation, we present three move-up models for realistic-sized problems. The first two of these models extend existing work by proposing a new simulation-based optimisation algorithm. The third move-up model is a new integer programming model which incorporates some of the insights obtained from the small-scale Markov models. Simulation-based numerical optimisation is employed to tune the model parameters and consequently, the model can also be viewed as an approximate dynamic programming model.

Artificial call data generated for the city of Auckland, New Zealand, are used for computational experiments. We find that when move-up is performed appropriately, it can significantly improve the system performance. Moreover, the integer program proposed in this work gives the best performance.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my supervisors, Dr Andrew Mason and Professor Andy Philpott for their unmeasurable support in this research. Their wide knowledge and logical way of thinking have been of great value for me.

My warm thanks are due to The Optima Corporation for providing me with their simulation package from which we have benefited greatly. Especially, I wish to express my gratitude to Jim Waite, Paul Day, Peter Ebden and Oliver Weide at Optima Corporation for their help with coding in the simulation package.

I am very thankful to Shane Henderson, Matt Maxwell, and Huseyin Topaloglu at Cornell University for their advice and collaborations during my research.

I thank the staff of the Department of Engineering Science.

Thank you to all my friends Ed Bulog, David Dempsey, Graeme Mak, Athena Wu, Rupert Storey, Catherine Roberts, Antony Phillips, Jonathan Cheng, Anita Walbran, John O'Sullivan, Sepi Irani, Eylem Kaya, Emily Clearwater, Javad Khazaei and Siamak Mor, who have been an important part in my personal and professional growth over these years.

During this work, I visited Cornell University and DTU. I am grateful to their hospitalities.

I would also like to thank David Dempsey, Rupert Storey and John Pearce for proofreading my thesis. Many exotic snacks, e.g. dry bananas, offered by John Pearce also have made the writing of this thesis much sweeter.

Most importantly, I owe my loving thanks to my parents Zhigang Zhang and Jinfeng Bai. They have lost a lot due to my study abroad for the last ten years. Without their support and understanding, I would never be able to finish this work.

This research was funded by the New Zealand Institute of Mathematics and its Applications (NZIMA) and partially supported by OptALI and New Zealand Postgraduate Study Abroad Awards.

Declaration

This research utilises a simulation package provided by the Optima Corporation. Simulation results presented in this work are based on artificial data and do not reflect the performance that the local EMS provider, St John, provides in reality.

Contents

1

Introduction

1.1 Background and Motivation

Emergency medical services (EMS) are organisations that are dedicated to providing acute medical care on site and/or transport to hospitals for follow-up evaluation. The core of EMS operations is the response process for each emergency call. Undoubtedly, timely response to an emergency call is the essence of planning EMS operations. When a patient suffers a cardiac attack, the chance of survival decreases by 10% every minute until care is received [1]. Therefore, it is not surprising that almost all performance targets EMS providers are contracted to meet have *response time* taken into account; response time is the duration from when a call is received until ambulance crew reach the accident scene. For example, a common performance target in North America is to have response times below 9 minutes for at least 90% of calls related to life-threatening conditions [18].

Planning EMS operations involves many interrelated decisions. These are classified at three different levels: (1) strategic decisions involve location and construction of fixed facilities, the purchase of equipment, and the hiring and training of specialised staff; (2) tactical decisions involve staff scheduling, ambulance stand-by locations and dispatch policies; and (3) operational decisions involve procedures to be followed by ambulance crew depending on the nature of calls.

Improving the design and operations of EMS systems has attracted much attention from operations research practitioners for many years. The primary reason is the importance of high-quality EMS operations to society; in addition, EMS systems are typically complex and involve uncertainties with respect to many aspects, which present interesting and challenging mathematical problems.

The core of EMS operations is the response process for each emergency call; a typical one is described in Figure 1.1. When an emergency call is received, a dispatcher chooses an ambulance (waiting at/returning to a stand-by location) to respond to the call; typically, the ambulance closest to the accident scene is dispatched.

Once the ambulance crews reach the scene, they perform an initial at-scene treatment of the patient. If no more medical care is required then the ambulance becomes free at the scene, and travels to a stand-by location, e.g. an ambulance base. More typically, however, transportation is required to a hospital and the ambulance travels to a stand-by location after completing a patient 'hand-over'. The duration from when a call is received until the dispatched ambulance completes service/becomes free is referred to as *service time*. It is easy to see that response time is part of service time.

This response process can be complicated by dispatch delay, mobilisation delay, calls of different priorities, ambulances with different capabilities, the need to dispatch multiple ambulances to some high priority calls, the use of lights and sirens for some calls to reduce travel times, and the possible diversion of an ambulance from one call to a higher priority one, etc. Dispatch delay refers to time spent on the phone gathering information and time taken to notify the ambulance crew. Mobilisation delay refers to time taken for the crew to reach the ambulance and prepare for departure. Mobilisation delay is often associated with ambulances that are standing by at ambulance bases; ambulances may also stand by at street corners or drive on the road before getting dispatched in which case mobilisation delay is typically zero.

Clearly, stand-by locations of ambulances have a major impact on response times and the research on designing ambulance location policies in order to respond to calls in a timely manner has been active since at least the 1970's. An ambulance location policy determines the stand-by location of each ambulance. A common ambulance location policy is to have every ambulance return to its pre-determined 'home base' at the conclusion of every call; this is known as a static ambulance location policy. In the past, this approach has been popular and is still used by some today.

Two static models, which are considered as the cornerstones in this research area, are

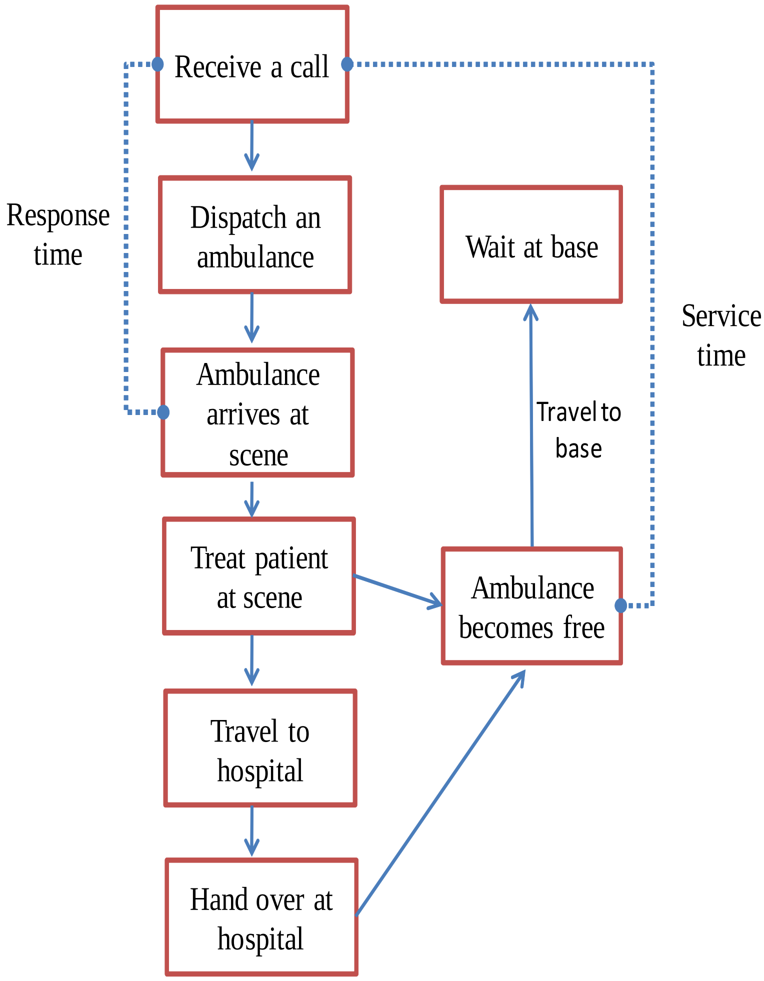


Figure 1.1: A typical response process for an emergency call

the location set covering model (LSCM) by Toregas et al. [43] and the maximal covering location problem (MCLP) by Church and Revelle [14]. Both models, which are formulated as integer programs, are viewed as deterministic models; they assume that each ambulance is always available at its home base when a call arrives. Since the development of LSCM and MCLP, there has been a surge of research aiming to develop more realistic models for optimsing static policies. For a good account of static models prior to 2003, we refer the reader to Swersey [42] and Brotcore et al. [10].

More recently, EMS providers have started to employ more dynamic ambulance location policies, known as move-up, system-status management, or ambulance redeployment.

Move-up is the practice of dynamically deciding the stand-by location of each free ambulance in an attempt to better position ambulances for future calls. A move-up policy uses real-time information, such as the location and status of each ambulance, to decide appropriate stand-by locations. Surveys conducted by Cady [12] in 2001 and Williams [46] in 2009 showed that the percentage of North American EMS providers using static policies has decreased from 41% to 30%, while those using move-up policies has increased from 23% to 37%; the proportion using a hybid strategy has changed from 36% to 33%.

The primary goal of implementing a move-up policy is to make better use of scarce ambulance resources in order to achieve better response times. Numerous operations research models have been developed to evaluate a fixed static policy or seek a high-quality static policy. In contrast to the rich literature on the static problem, the move-up problem has received less attention. To our knowledge, the first journal paper for designing a move-up policy for real-world ambulance operations was published in 2000 by Gendreau [20]. As stated by Gendreau [20], one primary reason behind the lack of motivation for exploring the move-up problem is that

> "... limitations of past technology did not allow for real-time solutions of dynamic large-scale problems. "

Entering the 21st century, advanced computing resources and technologies have enabled EMS providers to use software-based tools to manage their operations. For example, EMS providers can track the location of an ambulance through the Global Positioning System (GPS) and visualise their operations on a computerised map. New technologies make it possible to develop models that make use of a large amount of real-time data, which can be collected and processed with ease. Meanwhile, in the operations research community, the advent of new theories/methodologies and powerful solution techniques opens up many opportunities to solve problems of realistic dimensions in real time.

Although many EMS providers are implementing some form of move-up, most of the policies are constructed in ad-hoc ways, leading to limited performance gains and crew frustration at what are perceived as 'pointless moves'. For example, Alanis et al. [2] tested a set of move-up policies using realistic but artificial data from Edmonton, Canada. Unsurprisingly, performance of the policies varied significantly in their test; the percentage of calls that are reached within the associated response time target is 15.3% lower using the worst move-up policy than the best one.

The need to design high-performance move-up strategies is reinforced by the challenges that EMS providers face nowadays. On one hand, "with the population growth, changing social habits (including greater alcohol-related problems), an aging population and more patients having chronic illnesses" [40], demand for emergency services is growing at a rapid rate. For example, the city of Auckland, New Zealand, has seen a consistent increase in demand of 4% each year; this growth is expected to continue in the future [1]. On the other hand, EMS providers are facing problems such as rising equipment costs, underfunding, understaffing, and increasing congestion on urban roads. The consequence is that EMS providers have difficulty meeting their performance targets, e.g. the local ambulance service operator for Auckland failed to meet its contracted targets in 2011 [40].

From an ambulance-logistics software provider's perspective, the ability to offer high-performance move-up strategies can increase their chance of securing contracts from EMS providers. This research is supported by The Optima Corporation, specialising in the development of software for ambulance logistics; we give more details about their software in Chapter 6. A move-up model by Richards [39] has been embedded into Optima's software. However, Optima sees the value of move-up and is always interested in seeking and exploring alternative move-up approaches. In this thesis, we present three move-up models in response to their request.

Summarising, this research is motivated by the value of efficient EMS operations to society and the lack of systematic approaches to form high-quality move-up policies for large-scale operations. Our approach is outlined in the next section.

1.2 Outline of the Thesis

In the first chapter, we introduce the move-up problem that motivates this research, and summarise the main contributions in this thesis.

In Chapter 2, we give a review on the existing move-up models and provide the reader with a road map of subsequent chapters.

Chapters 3, 4 and 5 are focused on the formulation of three dynamic programming move-up models. These models quickly become intractable for realistic-sized problems so the main purpose of developing these models is to gain insights from (near) optimal move-up policies involving one and two ambulances in small-scale settings.

Chapter 6 is a transition chapter, taking us to practical move-up models. All of the large-scale move-up models studied in this research involve simulation-based optimisation. This chapter introduces Optima's simulation package used for these models. We present the simulation environment which we use to test and compare different ambulance location policies. We then familiarise the reader with the simulation package through the demonstration of three 'optimised' static policies under three simplified scenarios. The static policies are used to benchmark the move-up policies in the next two chapters.

In Chapter 7, we present two move-up models, which we refer to as the ranked-base free-ambulance move-up model and the ranked-base all-ambulance move-up model. As the names suggest, a key input parameter for both models is a set of rankings of stand-by locations (ambulance bases). A simulation-based local search algorithm is proposed to optimise the rankings for use under each of the two move-up models. Empirical comparisons between the optimised move-up policies obtained from the local search algorithm and the benchmark static policies are conducted from various aspects.

Chapter 8 proposes an integer programming (IP) model to make move-up decisions. Some of the move-up insights obtained from the small-scale DP models are employed in the IP model. In addition, a numerical optimisation scheme is suggested to tune some of the model parameters in the hope that the resulting move-up policy is of high performance. Similar empirical comparisons, as in Chapter 7, are conducted with the addition of the optimised IP move-up policies.

1.3 Contributions

The main contributions of this thesis are:

- **Establishment of a set of theoretical results for the single-ambulance next-call move-up model**. The model is formulated as a discrete-time dynamic program to approximate a continuous optimisation problem. Besides the mathematical properties for optimal move-up policies, an error upper-bound for the move-up performance based on our discrete model with respect to that based on the continuous model is also presented.

- **Development of a label-setting algorithm to solve the single-ambulance next-call move-up model**. The algorithm is more effective than a standard solution technique – value iteration.

- **Formulation of infinite-horizon move-up models for one and two ambulances under the DP framework**. The models are an extension of a DP model in the literature, which provide more realistic settings to gain move-up insights.

- **Reduction in state space size for the single-ambulance infinite-horizon move-up model.** By exploiting the structure of the model, an alternative formulation is proposed with a much reduced state space dimension.

- **Development of a simulation package to analyse ambulance behaviors based on the two-ambulance infinite-horizon move-up model**. The package (which is is written in C#) helps us to gain new insights into move-up strategies.

- **Development of a simulation-based local search algorithm for the two move-up models presented in Chapter 7**. For the ranked-base free-ambulance move-up model, computational experiments suggest that our algorithm gives solutions that are at least as good as those generated by an algorithm proposed in the literature. For the ranked-base all-ambulance move-up model, our simulation-based algorithm has the advantage of more accurate performance estimation compared to the mathematical approximations used in the literature.

- **Formulation of an IP move-up model**. Empirical results suggest that the optimised move-up policies based on this IP model and optimised ranked-base all-ambulance move-up policies give similar performance in terms of maximising the percentage of calls reached within a specified target time, both of which dominate the (benchmark) static policies and the optimised ranked-base free-ambulance move-up policies.

2

Literature Review

In this chapter, we give a review on the existing move-up models and provide the reader with a broad summary of the motivations for the subsequent chapters.

Berman [5] made the first attempt to solve the move-up problem using dynamic programming (DP). The model aimed to find the optimal move-up policy for two *mobile servers* in order to minimise the expected travel time between two consecutive events; a event can be a dispatch to a call or a completion of service. Note that we use the term mobile servers as the model was developed for more generic move-up problems involving severs that travel to demand points. The second and third models proposed by Berman [6, 7] are a natural extension of his first model. The models consider move-up for multiple mobile severs under the same framework. The main difference between these two models is that where a server becomes available. In [6], a server becomes free at a pre-determined home base while in [7], a server becomes free at the nearest vacant location from its associated demand point. A vacant location refers to one that is not occupied or going to be occupied by any server according to previous move-up decisions.

One advantage of DP is that it models the stochastic nature of the EMS operations directly; however, this approach quickly becomes infeasible for any realistic-sized problems. Therefore the main use of the DP approach is to provide insights into the structure of optimal move-up policies for very small instances (e.g. one or two ambulances). Then one attempts to use these insights in approximate models for practical move-up problems. For example, Delasay et al. [16] developed a dynamic programming model to study the optimal dispatching of two trucks to shovels in surface mines; then they used the insights from the small model in the development of a heuristic model for multiple trucks.

Berman's models were developed in a more generic setting in which mobile servers travel to 'customers'. Consequently, some important characteristics unique to EMS operations were not incorporated. There is a need to develop new models that are specifically designed to approximate EMS operations. We then can gain insights of (near) optimal move-up policies in more realistic scenarios. We take a 'start-from-scratch' approach by starting with the study of optimal move-up policies for only one ambulance. Chapters 3 and 4 propose two dynamic programming move-up models, each of which only involves one ambulance in service. The first move-up model aims to maximise the expected reward for the next call while the second model aims to maximise the expected reward in an infinite horizon. In Chapter 5, we consider optimal move-up policies for two ambulances in an infinite horizon, which is a natural extension of the move-up model for a single ambulance in an infinite horizon presented in Chapter 4. We contrast our model assumptions to those used by Berman in [5].

The first move-up model, the dynamic double standard (DDSM) model, for realistic-sized EMS operations was proposed by Gendreau et al. [20]. The model is formulated as an integer program, which is an extension of their previous work [19] for generating static ambulance location policies. It takes some current ambulance configuration (i.e. a set of ambulance stand-by locations) and produces a set of moves for the ambulances. The objective is to maximise a score function measuring the benefit of a final ambulance configuration minus a cost function of the travel required to achieve that configuration. The call demand is aggregated into a set of demand points on the network. A demand point is considered covered if it can be reached by at least one ambulance within a specified response time target. The benefit of an ambulance configuration in this model is the sum of demand-weighted *backup coverage*. The backup coverage refers to the set of demand points that can be covered by at least two ambulances. Recent ambulance move-up history is used to impose practical constraints such as avoiding round trips, repeated movements of the same ambulance, etc.

A move-up decision is recommended whenever the number of free ambulances changes. A parallel tabu search algorithm was developed to speed up the solution process. This tabu search continuously pre-computes the solution for every possible future scenario in which one of the free ambulances is dispatched to the next call; however, there is no attempt to pre-compute solutions for scenarios in which busy ambulances become available. The computational results for the DDSM model provided in [20] were more focused on the ability to generate real-time solutions with the tabu search and efficient implementation with parallel computing technology. No results on the performance improvements generated by this model were provided, i.e. no comparison with (optimised) static policies.

Richards [39] revisited the DDSM model and proposed a different score function measuring the benefit of an ambulance configuration. Recall that in the DDSM model, the benefit collected from a demand point is either zero or the demand at this point. In [39], for each demand point, each additional ambulance that can cover the point, up to a target number, contributes to a concave increasing reward function. The score function is the sum of the reward functions over all demand points. In addition, under the assumption of perfect information, busy ambulances, which are likely to be available at some 'look-ahead' time, also contribute rewards around the location at which they will become free. This model has been embedded into Optima's software and experiments using simplified historical data from a client (whose identity cannot be disclosed for confidentiality reasons) of Optima were performed. The results showed that in comparison with a static policy used in practice, a move-up policy based on this move-up model can improve the percentage of calls reached within a target time of 8 minutes by 4.1% and a target time of 13 minutes by 3.6%.

Andersson et al. [3] proposed an integer program for move-up. They introduced a score function, namely preparedness, which is used in non-linear constraints; the constraints are designed to ensure that for each demand point, some minimum preparedness level is achieved. The target configuration is the one that minimises the maximum travel time of the ambulances being moved. When measuring preparedness at each demand point for a given ambulance configuration, ambulances are ranked according to increasing travel time to each demand point; the closer an ambulance, the more contribution. The model is solved whenever the preparedness for at least one demand point falls below a specified minimum level; to obtain a move-up decision in a short computation time, a heuristic was developed.

A simulation package was developed to evaluate the performance of their move-up model. Artificial call data were generaged based on the city of Stockholm, Sweden; calls were prioritised from 1 to 3 where priority-1 calls are the most urgent calls with life-threatening conditions. Varying arrival rates and minimum preparedness levels were used for comparisons. The mean time to solve the move-up model was 2.24 seconds and the maximum solve time was 5.59 seconds. For each tested instance, the performance of a static policy was used as the benchmark; the authors did not state whether the static policy was optimised or not. Results for three response time targets (which were 10 minutes, 15 minutes, and 20 minutes) with respect to priority-1 calls were reported. To summarise, the benefit of move-up increased as the minimum preparedness level increased; the trade-off to maintain a high minimum preparedness level was a large number of relocations, where a relocation refers to changing the stand-by location of a free ambulance.

The integer programs discussed above consider a set of feasible solutions (typically more than one feasible solution) and find the 'optimal' solution. The location of each ambulance plays a key role to decide the optimal solution because there is a cost associated with moving from one location to another; in other words, a move-up policy based on these models considers all free ambulances for move-up and chooses an appropriate ambulance configuration from a set of candidate configurations in real time. We refer to such a policy as a dynamic move-up policy to distinguish it from other forms of move-up policies introduced below.

The second type of move-up policy studied in the literature is referred to as *compliance-table move-up policy*. A compliance-table move-up policy defines, for each number n of free ambulances, a pre-determined ambulance configuration $C(n)$, which is listed in a so-called compliance table C. Whenever the number of free ambulances changes, the dispatcher decides a set of moves to reach the corresponding configuration, which is usually determined by solving an assignment problem to minimise total travel times. The main difference between a compliance-table move-up policy and a dynamic move-up policy is that the former forces n ambulances into a unique configuration while the latter does not. The similarity is that all free ambulances are considered for move-up.

Gendreau et al. [21] proposed an integer program to generate compliance-table move-up policies. Their objective is to maximise a score function, computed by multiplying the demand-weighted coverage with n ambulances by the probability $p(n)$ of having n free ambulances. The coverage refers to the set of demand points that can be covered by at least one ambulance assuming that n ambulances are at the stand-by locations defined by $C(n)$. The value of $p(n)$ is obtained by means of a binomial distribution. When formulating the optimisation model, they restrict the number of new stand-by locations that can be used when going from configuration $C(n)$ with n free ambulances to an 'adjacent' configuration $C(n+1)$. This gives configurations that are similar meaning adjacent configurations share common standby locations.

The performance of this model was tested on data from Montreal and Laval with a varying total number, N, of ambulances on duty ($N = 3, 4, 5, 6$) and a varying number of different standby locations allowed between adjacent configurations. Each optimised compliance-table move-up policy was compared with a static policy. The static policy was constructed by assigning an ambulance to each of the stand-by locations defined in configuration $C(N)$. Simulation experiments showed that their model can improve on the static policies with respect to the expected percentage of calls reached within a specified response time target; the improvements ranged from about 2% to about 13%. However, these static policies were not optimised; each static policy was 'simply' defined based on configuration $C(N)$. Therefore the 'best' performances the static strategy can produce were not used for comparison, which means the relative improvement may not be as high as the results suggested.

In the same vein, Alanis et al. [2] proposed a Markov chain model to quickly approximate the performance of a given compliance-table move-up policy for large-scale EMS operations. In their model, the state space contains two states for each number, n, of free ambulances: an 'in-compliance' state and an 'out-of-compliance' state. An 'in-compliance' state means that all free ambulances are at standby locations defined by configuration $C(n)$; an 'out-of-compliance' state is an aggregation state representing configurations in which one of the free ambulances is not at the stand-by location in $C(n)$. These configurations are as follows. (1) when an ambulance has just became free, the $n-1$ other ambulances are at the stand-by locations in $C(n-1)$ and the additional ambulance is free either at one of the demand points or at a hospital, (2) when an ambulance has just become busy, the n free ambulances are at stand-by locations defined by removing, in turn, each one of the stand-by locations in $C(n+1)$. This model was tested using data from the city of Edmonton, Canada. Several metrics, including the response time distribution, were compared between using this model and simulation. The results suggested that it is able to provide close-to-simulation estimations. An important use of this model, envisioned by the authors of [2], is as a surrogate for simulation to screen a large number of compliance tables and then carefully evaluate only the 'good' ones via simulation.

Note that in practice, a compliance table can contain more than one configuration for each number of free ambulances. For example, the city of Edmonton, Canada, is employing a compliance-table move-up policy for its EMS operations [2]. The table contains as many as 144 configurations when there are 8 ambulances available. This means that the dispatcher needs to choose a target configuration and decide ambulance moves to reach the configuration. The mechanism of constructing the table and choosing the appropriate configuration is not publicly available. In this thesis, we consider that it is more appropriate to refer to such a compliance-table move-up policy as a dynamic move-up policy.

The third type of move-up policy, which we refer to as *newly-freed-ambulance move-up policy*, was first explored by Restrepo [30] using an approximate dynamic programming(ADP) model. Under such a policy, a stand-by location for the newly-freed ambulance is determined and the ambulance then follows the fastest path to this location.

In the language of (exact) dynamic programming, a value function defined for all possible states is needed for the decision-making process. The value function can be interpreted as a look-up table containing values for all possible states. This look-up table quickly becomes intractable for many real-world stochastic optimisation problems, which is one of the main reasons limiting the use of dynamic programming. Under the ADP framework, one seeks to estimate the true value function using some form of approximate architecture. Restrepo [30] used a linear approximation architecture for his move-up problem. The general approximation form can be summarised as below

$$V(S) = \sum_{f \in F} \theta_f \phi_f(S),$$

where S represents a state variable, V is the approximate value function, F is called a set of features that aim to capture the important characteristics of the state variable, each $\phi_f(S)$ is called a basis function to quantity the value of the corresponding feature at state S and each θ_f is a tunable parameter. The best approximation is found by tuning parameters $\theta_f, f \in F$ via simulation-based regression; for a general explanation of ADP theories and solution techniques, see Powell [38]. Similar ideas have been developed in control theory [8], the computer science and artificial intelligence communities [41]. The linear approximation architecture eliminates the need to explicitly store a function value for every state and can thus model much more complex EMS operations such as an ambulance going to the scene, treating at the scene, and transporting to a hospital.

This ADP model was tested using artificial but realistic data from the cities of Edmonton in Canada and Melbourne in Australia. However, the simulation environment contained significant errors [34], so we do not report on the computational results.

Maxwell [34] extended this ADP model in various aspects. Firstly, Maxwell removed the errors in the simulation environment and updated the computational results. The static strategy was used to benchmark the performance of the Restrepo ADP model. For each case study, the static policy was found by screening a large number of solutions and selecting the one that gave the best performance, i.e. the expected percentage of calls reached within a specified response time target. In the case of Edmonton, the performances using the static policy and Restrepo's ADP move-up policy were 71.6% and 74.5%, respectively; the move-up policy led to an extra 2.9% of calls reached within the response time target. Maxwell notes that "A city-wide increase of just 1% is quite a significant improvement in the context of ambulance redeployment. A city the size of Edmonton would have to purchase, maintain, and staff an additional ambulance at a cost of approximately 1 million dollars a year to sustain such an improvement". In the case of the Melbourne case, they were 73.3% and

73.1%, respectively, meaning that the Restrepo ADP move-up policy did not provide an improvement on the static policy.

Maxwell [34] then proposed a different set of basis functions under the linear approximation architecture. For the computational experiments, the same data from Edmonton and Melbourne were used to test the performance of this new ADP model. The performance of Maxwell's ADP move-up policies (newly-freed-ambulance move-up policies) obtained from implementing the regression algorithm used by Restrepo [30] and two direct search algorithms, which are discussed in more detail in Chapter 7. For both cases, Maxwell's ADP move-up policies based on the regression approach were unable to outperform the benchmark static policies; Maxwell's ADP move-up policies obtained from the direct search approach performed better than both the static policies and Restrepo's ADP move-up policies based on the the approximation architecture in [30].

In addition, Maxwell [34] also made the first attempt to bound the performance of any ambulance location policy, i.e. an upper bound for the expected percentage of calls reached within a response time target under any ambulance location policy.

Recently, Schmid [44] proposed an ADP model for the ambulance move-up and dispatch problem. Recall that both Restrepo and Maxwell used the linear approximation architecture to tackle the challenge of an intractable state space dimension; the present author took another approach – aggregation to reduce the size of the state space. Aggregation does not simplify the state of the system (the system in the ADP model 'moves forward' as in the real-world system); rather it simplifies how the value function is approximated. A set of aggregation states is identified and one seeks to approximate the value function over these states. Assume that state S is the real state of the system: according to some aggregation rule, it is mapped to an aggregate state S^a and $V(S)$ is approximated to be equal to $V(S^a)$.

The Schmid ADP model assumes that the operations on distinct days are independent. The entire service area is partitioned into a set of grid cells. The length of the planning horizon (24 hours) is split into several intervals. An aggregation state consists of two pieces of information: the number of *at-base* ambulances and the number of *pending calls* in each cell and each time interval. So a real state S comprised of many parameters is mapped to an appropriate aggregation state S^a according to these two pieces of information. Here an at-base ambulance refers to an ambulance that is standing by at a base; a pending call refers to a call that has arrived but not yet received a response ambulance.

The performance of the Schmid ADP model was tested using 42 days of historical data from an EMS provider in Austria. Training datasets, which were derived from the real data, were used to approximate values for the aggregation states. Results for two scenarios were presented in [44]: for both scenarios, the objective was to minimise the average response time. For one scenario, a fixed dispatch policy (dispatching the closest at-base ambulance) was used while for the other scenario, the dispatch policy itself was part of the optimisation problem where any one of the at-base ambulances can be dispatched. For both scenarios, a target base was required whenever there was a newly-freed ambulance. So, considering the move-up problem alone, the model forms newly-freed ambulance move-up policies.

For the scenario where the dispatch policy was fixed, the optimised move-up policy was able to reduce the average response time compared to an optimised static policy. When the dispatch policy was not fixed, the optimised move-up-and-dispatch policy was able to further improve performance. However, we do not think this model appropriately describes real-world EMS operations: the model assumes that an ambulance can be dispatched to a call only if it is standing by at a base or has just became free; a newly-freed ambulance is either dispatched to a pending call immediately or moves to a target base. If an ambulance is driving towards a base on the road, it cannot be dispatched to any call. This is a very unreasonable assumption to make, so we question the validity of these results. In addition, the author of [44] commented that the ADP model of Maxwell [34] considers moving ambulances between ambulance bases. This comment is incorrect, as Maxwell's model generates newly-freed-ambulance move-up policies; there is no base-to-base ambulance moves.

In this thesis, Chapters 7 and 8 present three move-up models aimed for real-world EMS systems. All three move-up models are largely motivated by Maxwell's ADP model. Maxwell made several observations on newly-freed-ambulance move-up policies derived from his ADP model. A key observation is that a newly-freed-ambulance move-up policy determined by his ADP model can be stated in an alternative way using a so-called priority list.

Briefly, a priority list ranks stand-by locations. In Chapter 7, we describe in detail how to construct a newly-freed-ambulance policy using a priority list, which we refer to as a ranked-base free-ambulance move-up policy. Our contribution is to propose a simulation-based local search algorithm which sorts the items in the priority list in order to maximise the performance of the resulting move-up policy. Computational experiments in Chapter 7 suggest that this sorting algorithm is slightly more effective than the numerical optimisation approach taken by Maxwell under the ADP framework.

The concept of making move-up decisions using a priority list also motivates us in the development of compliance-table move-up policies. In Chapter 7, we show how to construct a compliance-table move-up policy using a priority list, which we refer to as a ranked-base all-ambulance move-up policy. Naturally, the simulation-based local search algorithm proposed to find a good ranked-base free-ambulance move-up policy can be used to seek a high-quality ranked-base all-ambulance move-up policy. In other words, the same sorting method is applied in order to find the locally optimal priority list for use under a ranked-base free-ambulance move-up policy or a ranked-base all-ambulance move-up policy.

In Chapter 8, we explore dynamic move-up policies based on an integer programming (IP) model. For our IP model, the score function to measure the benefit of an ambulance configuration is different from those presented in [20], [39], and [3], which are discussed in Chapter 8. Briefly, our score function is constructed based on the approximation architecture of Maxwell's ADP model and the insights gained from our DP models for small problems. In addition, we employ a simulation-based numerical optimisation algorithm to tune some of the model parameters associated with the score function and the cost function in order to find the best possible move-up policy. A key input to initialise the parameters associated with the score function is the optimised priority list for a ranked-base all-ambulance policy. Therefore, the IP model can be viewed as an extension of this base-ranking move-up model.

3

Optimal Move-up of Single-Ambulance Next-Call Model

3.1 Overview

The first move-up problem we study in this thesis is called single-ambulance next-call move-up model: assuming there is only one ambulance in service, we aim to find an optimal move-up policy in order to maximise the expected reward for the next call.

The most relevant work to the models we study in this chapter and the next two chapters is the dynamic programming (DP) model proposed by Berman [5]. The Berman model considers the move-up problem for two ambulances as discussed in Chapter 2. Because we only consider one ambulance at this stage, we feel that the best place to contrast our DP approach to Berman's is in Chapter 5 where we present a new DP move-up model for two ambulances.

It may seem strange that we start by exploring the (short-term) benefit of move-up for only the next call as it makes more sense to optimise move-up that focuses on the benefit for an infinite number of calls. Intuitively, we think, under some circumstances, a short-term optimal move-up policy might be very similar to a long-term optimal move-up policy. For example, in a system with a low arrival rate, an ambulance typically has time to reach any location before becoming busy again, so focusing on just the next call may give good results. However, we expect that an optimised static policy would also perform well for the same reason. In other words, there is little performance difference between an optimised short-term move-up policy, an optimised long-term move-up policy, and an optimised static policy. It becomes hard to predict the performance of different location strategies when the ambulance is more likely to become busy while travelling to a stand-by location due to a higher arrival rate. Therefore, we are motivated to formally develop the next-call move-up model in this chapter and then investigate the performance of this model, the static model and the infinite-horizon move-up model, which is presented in Chapter 4.

The remaining sections of this chapter are organised as follows. In Section 3.2, we present the assumptions and formulation of a DP model. In practice, this move-up problem is a continuous optimisation problem: an ambulance travels on a continuous transportation network. We model this problem in a discrete-time manner, i.e. a discrete network consisting of nodes and arcs is used as an approximation. We provide a bound on the difference between the performances of an optimal move-up policy based our discrete model and that of the true but unknown optimal move-up policy. In Section 3.3, we use simple examples to show insights into the structure and characteristics of optimal move-up policies based on our DP model. We then establish a set of theoretical results regarding an optimal move-up policy and the associated value function. In Section 3.4, we describe a label-setting solution technique as an alternative to the standard value iteration. We end this chapter with a summary in Section 3.5.

3.2 Single-Ambulance Next-Call Model Assumptions and Formulation

As discussed in Chapter 1, real-world ambulance operations are very complex due to many factors such as varying call priorities, changing travel times, the possible diversion of an ambulance from one call to a higher priority one, etc. In this thesis, we study move-up in simplified systems.

Consider a road network G on which call arrivals follow a Poisson process with a constant arrival rate λ. All calls are assumed to be of the same type with a common response time target W. The next call to arrive is located at position $u \in G$, which has a density function $p(u)$; in other words, a spatial distribution p of call demand is defined on G. Here for convenience u is assumed to be in $\mathbb{R}$, but our model can be extended to accommodate call locations in two-dimensional domains. Let $\tau(u,v)$ denote the travel time along the shortest path from position u to position v which we assume is symmetric. This implies that $\tau(u,v)$ is a metric on G. With respect to this metric we assume the existence of a Lipschitz function r on G, where $r(x)$ describes the expected reward earned for a call when the ambulance is dispatched from position x. This is calculated based on a set of "reward" rules and $p(u)$.

In our model, we define $r(x)$ to be the probability of reaching a call on time given the ambulance is dispatched from position $x \in G$. A call is considered reached on time if its response time is within a specified target W. We assume that there is no queueing, so call-arrivals when the ambulance is busy are lost to the system. This means when the ambulance has just become free, it will serve the next call-arrival. In addition, we ignore dispatch delay and mobilisation delay for simplicity[1]. Therefore, the response time is the duration of the trip from the dispatch location to the next call's location along an shortest path. We further assume deterministic travel speeds at each position for all possible driving directions. Let

$$\mathcal{N}(x,W) = \{u \mid \tau(x,u) \le W\}.$$

Then we have

$$r(x) = \int_{\mathcal{N}(x,W)} p(u)du.$$

It is easy to show that $r(x), x \in G$ is a Lipschitz function as long as $p(u)$ is bounded. This means there is some constant K such that

$$|r(x) - r(y)| \le K\tau(x,y) \quad x,y \in G \tag{3.1}$$

We are interested in finding a policy that defines, for any starting position $x \in G$, where the ambulance should stand by and which path it should travel along to the stand-

[1] We give more discussions about the possible impact of including these two delays in move-up models in Chapter 6.

by location in order to maximise the expected reward for the next call. It is easy to see that in practice we have a continuous optimisation problem to solve. In this model, we aim to find an optimal move-up policy defined on a discretised network as an approximation to this continuous optimisation problem.

We discretise the network G into a set N of nodes and let L denote the set of arcs, where L consists of $(i, i), \forall i \in N$, representing a directed arc to node i itself and (i, j) representing a unique directed arc from node i to j, $i, j \in N$, $i \neq j$. Let $N_i = \{k \in N : (i, k) \in L\}$ denote the set of possible successor nodes of node $i \in N$. Without loss of generality, we will use the term 'policy' to mean a 'next node' policy π defined on the discrete network (N, L) that specifies a successor node $\pi(i)$ of $i, \forall i \in N$. We approximate the performance of a policy π using a 'wait-then-jump' scheme on the discrete network. More specifically, given the ambulance is at node i and the successor node is $\pi(i)$, we assume the ambulance waits for a time interval $\Delta t_{(i,\pi(i))}$ which equals the travel time $\tau(i, \pi(i))$ if $\pi(i) \neq i$ or a strictly positive constant Δt otherwise. During this time interval, if the next call occurs, the ambulance is dispatched from node i with a reward $r(i)$. Otherwise the ambulance jumps instantaneously to node $\pi(i)$ at the end of this time interval.

Note that the time duration Δt associated with $\pi(i) = i$ is just a symbolic term, so that the move-then-jump scheme can be used without loss of generality; if $\pi(i) = i$, the ambulance just stays at node i until it gets dispatched with reward $r(i)$.

In practice, the ambulance may not always be at a node defined in our model, and moreover, it moves along a continuous path. We interpret our discrete policy π for the continuous problem as follows. If the ambulance's initial location is at node k, the ambulance just follows the continuous path that visits nodes defined by $\pi()$ starting from k. If the ambulance's initial location is on an arc between node k and k', there are three possible scenarios. The first scenario is $\pi(k) = k'$ in which case the ambulance moves to node k' and then follows the the continuous path that visits nodes defined by $\pi()$ starting from k'. The second scenario is $\pi(k') = k$ in which case the ambulance moves to node k and then follows the continuous path that visits nodes defined by $\pi()$ starting from k. The third scenario is $\pi(k) \neq k'$ and $\pi(k') \neq k$ in which case the ambulance moves to the closer node and then follows the continuous path that visits nodes defined by $\pi()$ starting from the closer node.

Implicit in our definition of π is that there is a discrete path consisting of a series of wait-then-jump moves starting from any node $k \in N$. Let d_k^π denote the discrete path starting from node k under the control of policy π, where $d_k^\pi(0) = k, d_k^\pi(i) = \pi(d_k^\pi(i-1)), i = 1, 2, ...$ The 'wait' time interval for $(d_k^\pi(i), d_k^\pi(i+1))$ is denoted by $\Delta t_k^\pi(i) = \Delta t_{(d_k^\pi(i), d_k^\pi(i+1))}$, and the time elapsed starting from node k after i steps under policy π is denoted by $t_k^\pi(i)$ which is defined as below.

$$t_k^\pi(i) = \begin{cases} 0 & i = 0 \\ \sum_{j=0}^{i-1} \Delta t_k^\pi(j) & i = 1, 2, ... \end{cases}$$

We aim to determine a policy π^* such that the expected reward for the next call starting from every node is maximised. Let $J^\pi(k)$ denote the expected reward given the ambulance is starting from node k under the control of policy π, then we have

$$J^\pi(k) = \sum_{i=0}^{i=\infty} e^{-\lambda t_k^\pi(i)} (1 - e^{-\lambda \Delta t_k^\pi(i)}) r(d_k^\pi(i)) \tag{3.2}$$

where $e^{-\lambda t_k^\pi(i)}$ represents the probability that the next call has not occurred before the ambulance jumps to node $d_k^\pi(i)$. The immediate reward is $(1 - e^{-\lambda \Delta t_k^\pi(i)}) r(d_k^\pi(i))$ which is the probability that the next call occurs during the 'wait' interval $\Delta t_k^\pi(i)$ at node $d_k^\pi(i)$ times the expected reward from node $d_k^\pi(i)$.

Equation (3.2) can be written in the recursive form as

$$\begin{aligned} J^\pi(k) &= (1 - e^{-\lambda \Delta t_k^\pi(0)}) r(k) + e^{-\lambda \Delta t_{(k,\pi(k))}} \sum_{i=0}^{i=\infty} e^{-\lambda t_{\pi(k)}^\pi(i)} (1 - e^{-\lambda \Delta t_{\pi(k)}^\pi(i)}) r(d_{\pi(k)}^\pi(i)) \\ &= (1 - e^{-\lambda \Delta t_k^\pi(0)}) r(k) + e^{-\lambda \Delta t_{(k,\pi(k))}} J^\pi(\pi(k)). \end{aligned}$$

Let $V(k)$ be the maximum expected reward at node $k \in N$ under policy π^*, i.e. $V(k) = \max_\pi J^\pi(k)$. We now have a DP model, and the Bellman optimality equation [8] for the maximum expected reward $V(k)$ at node $k \in N$ is

$$V(k) = \max_{k' \in N_k} ((1 - e^{-\lambda \Delta t_{(k,k')}})r(k) + e^{-\lambda \Delta t_{(k,k')}} V(k')). \tag{3.3}$$

The state space for this DP model is the set N of nodes on the network. At each node (state) $k \in N$, a move-up decision $k' \in N_k$ is made. The ambulance then waits for time $\Delta t_{(k,k')}$ at its current location k. If the next call occurs during this interval, the ambulance is dispatched from node k giving a reward $r(k)$. If no call arrives, the ambulance jumps to node k'. Note that to ensure a policy is unique, we break ties in (3.3) by assuming the ambulance only makes a move if this gives a strict improvement in the objective.

It is worth noting that this problem can be viewed as a stochastic shortest path problem [9] if we reformulate the problem with the objective of minimising the probability of not getting to the next call within the target response time. We construct a network of nodes N and arcs L as we defined for our model plus a termination node T and an arc from each node $i \in N$ to T. The length associated with an arc $(i, j) \in L$ is zero. The length associated with each arc$(i, T), \forall i \in N$, is $1 - r(i)$. At each node $i \in N$, the ambulance can stay at node i or move to any node $j \in N_i$; the probability of reaching node j as a result of this decision is $e^{-\lambda \Delta t_{(i,j)}}$ and the probability of reaching node T instead is $1 - e^{-\lambda \Delta t_{(i,j)}}$. We seek the shortest expected path from each node $i \in N$ to T, which gives an optimal move-up policy π.

For each node $i \in N$, it is natural to consider the difference between following the optimal continuous path which is unknown and the optimal discrete path found by our model. If we let c^* be the optimal continuous path starting from node i and d^* be the optimal discrete path starting from node i, we establish the following theorem.

Theorem 1.

$$|R(c^*) - R(d^*)| \leq K \Delta \tau$$

where $R(c^*)$ is the expected reward when following the optimal continuous path starting from node i, $R(d^*)$ is the expected reward when following the optimal discrete path d^*, $\Delta \tau$ is the maximum travel time along any arc in the discrete network and K is as given in 3.1

The proof of Theorem 1 is given as follows. Let d denote a discrete path along which the ambulance performs a series of 'wait-and-jump' steps starting from node k at time $t = 0$. Let c denote a continuous path starting from node k that visits the same sequence of nodes, and denote by c_t the location of the ambulance at time $t, t \geq 0$ along path c. Let D define a transformation that maps any continuous path c to a unique discrete path $D(c)$. The path $D(c)$ describes a series of 'wait-and-jump' moves following the nodes that path c visits. Here we denote by $D(c)_t$ the last node visited by path c visits at time t.

The mapping D is not 1-1: any discrete path d can be the image under D of many continuous paths. Each of these continuous paths visits the nodes in the order defined by path d and the last node it visits is the last node that path d visits. Let $\mathcal{C}(d)$ denote the inverse image of d under D, i.e. $\mathcal{C}(d) = \{c : D(c) = d\}$.

The probability of a call occurring in the interval $[0, t]$ is $\int_0^t q(y)dy$, where $q(t)$ is the probability density of arrival time. Given $q(t)$, we can compute the expected reward if the ambulance follows a continuous path c as

$$R(c) = \int_0^\infty r(c_t)q(t)dt.$$

Similarly $R(d)$ denotes the expected reward along a discrete path d. Finally, let c^* be the optimal continuous path starting from node k and d^* be the optimal discrete path starting from node k. To simplify our notation, we let ϵ denote the term $K\Delta\tau$. We have the following results.

Lemma 2.

$$|r(c_t) - r(D(c)_t)| \leq \epsilon, \tag{3.4}$$

Proof. Clearly, the maximum travel time between position c_t and $D(c)_t$ is at most $\Delta\tau$ whence the result follows by (3.1). □

Corollary 3.

$$|r(d_t) - r(c_t)| \leq \epsilon, \forall c \in \mathcal{C}(d) \tag{3.5}$$

Proof. Similar to the proof for Lemma 2, the travel time between position d_t and c_t is at most $\Delta\tau$. Therefore the difference of the reward for the next call is bounded by ϵ. □

Lemma 4. *The difference between the expected reward when following path c and path $D(c)$ is bounded by ϵ, i.e.*

$$|R(c) - R(D(c))| \leq \epsilon \tag{3.6}$$

Proof.

$$\begin{aligned} |R(c) - R(D(c))| &= |\int_0^\infty q(t)r(c_t)dt - \int_0^\infty q(t)r(D(c)_t)dt)| \\ &\leq \int_0^\infty q(t)|(r(c_t) - r(D(c)_t))|dt. \end{aligned}$$

Using lemma 2, we have

$$\int_0^\infty q(t)|(r(c_t) - r(D(c)_t)|dt \leq \epsilon \int_0^\infty q(t)dt = \epsilon$$

□

Corollary 5. *The difference between the expected reward when following path d and any associated continuous path $c \in \mathcal{C}(d)$ is bounded by ϵ, i.e.*

$$|R(d) - R(c)| \leq \epsilon, \forall c \in \mathcal{C}(d) \tag{3.7}$$

Proof.

$$\begin{aligned} |R(d) - R(c)| &= |\int_0^\infty q(t)r(d_t)dt - \int_0^\infty q(t)r(c_t)| \\ &\leq \int_0^\infty q(t)|(r(d_t) - r(c_t))|dt \end{aligned}$$

Using Corollary 3, we have

$$\int_0^\infty q(t)|(r(d_t) - r(c_t)|dt \leq \epsilon \int_0^\infty q(t)dt = \epsilon$$

□

We recall theorem 1 is stated as below.

$$|R(c^*) - R(d^*)| \leq K\Delta\tau$$

Proof. Using Lemma 4 and Corollary 5, we have

$$\begin{aligned} |R(c^*) - R(D(c^*))| &\leq \epsilon && (3.8) \\ |R(d^*) - R(c)| &\leq \epsilon, \forall c \in \mathcal{C}(d^*) && (3.9) \end{aligned}$$

Based on the condition for optimality, we also have

$$\begin{aligned} R(D(c^*)) &\leq R(d^*) && (3.10) \\ R(c) &\leq R(c^*), \forall c \in \mathcal{C}(d^*) && (3.11) \end{aligned}$$

First suppose that $R(c^*) - R(d^*) \geq 0$. Then

$$R(c^*) \geq R(d^*) \geq R(D(c^*))$$

Therefore, (3.8) can be written as

$$0 \leq R(c^*) - R(D(c^*)) \leq \epsilon$$

$$0 \leq R(c^*) - R(d^*) + R(d^*) - R(D(c^*)) \leq \epsilon$$

Since $R(d^*) - R(D(c^*)) \geq 0$ from(3.10), we must have

$$0 \leq R(c^*) - R(d^*) \leq \epsilon \qquad (3.12)$$

Now suppose that $R(c^*) - R(d^*) < 0$. Then from (3.11) we have

$$R(c) \leq R(c^*) < R(d^*), \forall c \in \mathcal{C}(d^*)$$

Therefore, (3.9) can be written as

$$0 < R(d^*) - R(c) \leq \epsilon, \forall c \in \mathcal{C}(d^*)$$

$$0 < R(d^*) - R(c^*) + R(c^*) - R(c) \leq \epsilon, \forall c \in \mathcal{C}(d^*)$$

Since $R(c^*) - R(c) \geq 0, \forall c \in \mathcal{C}(d^*)$ (3.11), we must have

$$0 < R(d^*) - R(c^*) \leq \epsilon \tag{3.13}$$

Therefore, from (3.12) and (3.13) we have $|R(c^*) - R(d^*)| \leq \epsilon$ □

A corollary of Theorem 1 is that the maximum expected reward at any node $i \in N$ when following the optimal discrete path converges to the maximum expected reward when following the optimal continuous path as $\Delta\tau \to 0$.

3.3 Examples and Insights

We now consider two simple examples to gain insights into the properties of the optimal policy and the value function in this model. The first example considers move-up on a single road. The second example considers a small network.

Example 1: Single-Ambulance Next-Call Move-up Model on a Line

The horizontal axis in Figure 3.1 represents a network consisting of 30 nodes located along a line with one minute spacing. We assume Δt, the time step when the ambulance stays at its current node, also equals to 1 minute. Using a target response time of $W = 4$ minutes, each node k has a reward of $r(k)$ as shown on the vertical axis. Solving (3.3) with $\lambda = 0.5$ calls/hour gives an optimal policy in which, given the ambulance's current location (node

1,2,...,20), the free ambulance always moves to the adjacent node that is closer to node 20 and eventually stops at node 20. The maximum expected reward $V(k)$ under this policy is shown on the plot.

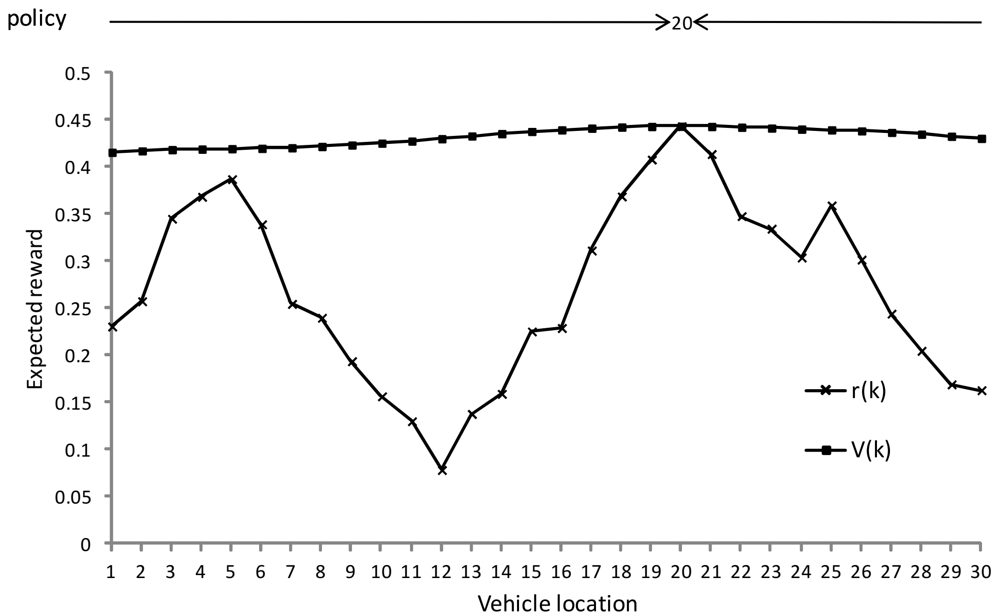


Figure 3.1: Plot of reward function $r(k)$, value function $V(k)$ and the optimal move-up policy for Example 1 with $\lambda = 0.5$ calls/hour. There is one optimal stand-by location, node 20. Note that $r(k)$ gives the expected reward if the ambulance stays at its initial node k, while $V(k)$ gives the maximum expected reward under the optimal move-up policy.

Figure 3.2 shows the optimal policy and its associated value function when the call-arrival rate increases to $\lambda = 3$ calls/hour. We can see that a new move-up policy is formed in which there are two stand-by locations, nodes 5 and 20. If the ambulance is initially located between node 1 and node 10, then it keeps moving to the adjacent node that is closer to node 5 and eventually stops at node 5; otherwise the ambulance moves to stand-by node 20 in a similar manner.

We can see that different optimal move-up policies arise from the two call-arrival rates. With $\lambda = 3$ calls/hour, we note that driving to node 20 is no longer optimal for an initial ambulance location between node 1 and 10, but instead the closer stand-by location (node 5) is optimal. This occurs because, for the higher call-arrival rate, there is too great a

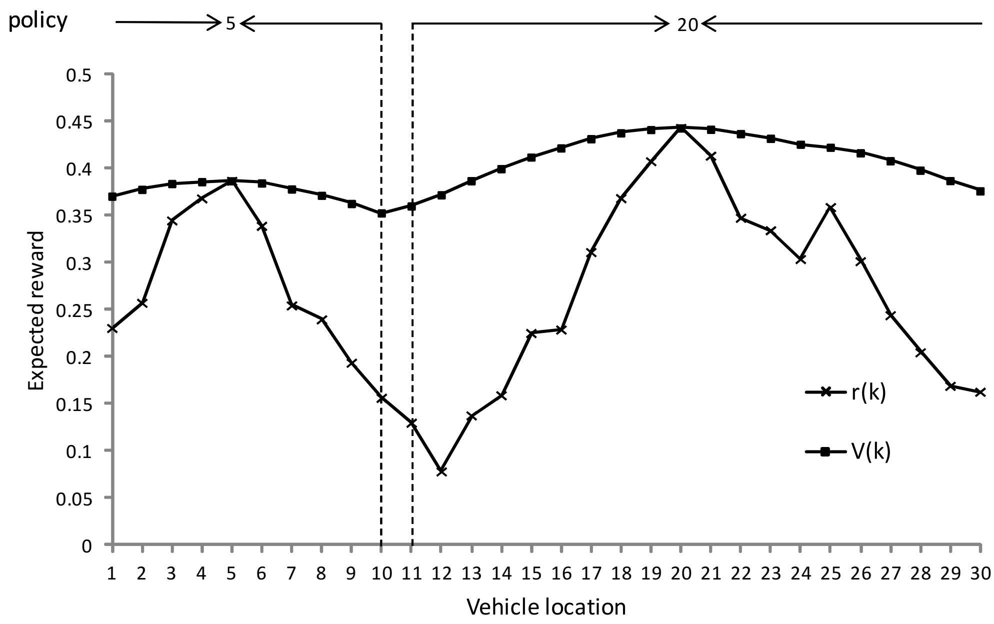


Figure 3.2: Plot of reward function $r(k)$, value function $V(k)$ and the optimal move-up policy for Example 1 with $\lambda = 3$ calls/hour. There are two optimal stand-by locations, nodes 5 and 20.

chance of the next call occurring during the move-up while the ambulance is at a location with a low expected reward for the next call.

Figures 3.1 and 3.2 suggest the following propositions:

Proposition 1. *For an optimal policy* π^*,

(i) $\pi^*(k) = k \Rightarrow V(k) = r(k) \quad \forall\, k \in N.$

(ii) $\pi^*(k) \neq k \Rightarrow V(\pi^*(k)) > V(k) > r(k) \quad \forall\, k \in N.$

Proof. From (3.3) we have the optimality equation

$$V(k) = (1 - e^{-\lambda \Delta t_{(k,\pi^*(k))}})r(k) + e^{-\lambda \Delta t_{(k,\pi^*(k))}} V(\pi^*(k)). \tag{3.14}$$

The proof of (i) follows by putting $\pi^*(k) = k$ in (3.14) and using $e^{-\lambda \Delta t_{(k,k)}} = e^{-\lambda \Delta t} > 0$. To prove the strict inequality $V(k) > r(k)$ in (ii) given $\pi^*(k) \neq k$ is straightforward given our assumption of moving only if this gives a strict improvement, i.e. the expected reward generated by moving at node k as given by (3.14) must be greater than the reward generated by staying at node k, which by (i) is $r(k)$. Furthermore, the fact that $V(k) > r(k)$ naturally leads to $V(\pi^*(k)) > V(k)$ in (ii) as follows. From (3.14), we have $V(\pi^*(k)) - r(k) = \frac{1}{q}(V(k) - r(k))$ where $q = e^{-\lambda \Delta t_{(k,\pi^*(k))}}$. Because $\lambda > 0$ and $\Delta t_{(k,\pi^*(k))} > 0$, we have the strict inequality $0 < q < 1$. Furthermore, $V(k) > r(k)$ and $1/q > 1$ imply $V(\pi^*(k)) - r(k) > V(k) - r(k)$, proving $V(\pi^*(k)) > V(k)$. □

Our definition of policy π does not prevent the ambulance from driving forever around a cycle on the network. However we now show we do not have to consider policies of this form.

Proposition 2. *For any starting node, the ambulance, under the optimal policy, always travels along a non-cyclic path to a node where it then stops.*

Proof. We prove this by contradiction. Assume for some starting node under the optimal policy, the ambulance travels around a subset Z of nodes in a cycle. Let k^* be the node with the maximum expected reward in this cycle, i.e., $k^* = \arg\max_{k \in Z}(r(k))$. It is easy to see that $V(k^*)$ is a convex combination of the rewards in this cycle because the next call must occur when the ambulance is at some node $k \in Z$. Then we must have $V(k^*) \leq r(k^*)$ which violates our assumption of moving only if this gives a strict improvement. □

Proposition 3. *Under an optimal policy, the reward $r(k)$ at a stand-by location $k : k = \pi^*(k)$ is a local maximum, i.e. $r(k) \geq r(k')$, $\forall\ k' \in N_k$.*

Proof. We prove this by contradiction. Assume $r(k)$ is not a local maximum, and so there is at least one adjacent node $k' \in N_k$ with $r(k') > r(k)$. The policy of staying at location k gives an expected reward of $V(k) = r(k)$. But, moving to k' and waiting there gives an expected reward of $(1 - e^{-\lambda \Delta t_{(k,k')}})r(k) + e^{-\lambda \Delta t_{(k,k')}} r(k') > r(k)$. Hence we have a contradiction. □

Proposition 4. *Suppose an ambulance starting at node u travels along some path and stops n nodes later at node v where it waits for the next call. Let the nodes be re-numbered as*

$0, 1, ..., n-1, n$ *along this move-up path, and let* $J(c,n),\ c = 0, 1, ..., n,$ *denote the expected reward when an ambulance following this move-up path to node* n *is at node* c*. The following conditions are necessary for this move-up path to occur in an optimal policy* π^**.*

(i) $J(c+1, n) > r(c),\ \forall\ c = 0, 1, ..., n-1$

(ii) $r(n) > r(c),\ \forall\ c = 0, 1, ..., n-1$

Proof. If this path describes an optimal policy, then we must have $J(c,n) = V(c),\ c = 0, 1, ..., n$ and $\pi^*(c) = c+1,\ c = 0, 1, 2, ..., n-1$. Condition (i) above then follows from Proposition 1 (ii). Proposition 1 (i) and (ii) give $r(n) = J(n,n) > J(n-1,n) > \cdots > J(c+1,n) > J(c,n) > r(c)$ for $c = 0, 1, 2, ..., n-1$, from which (ii) follows immediately. □

Proposition 5. *For any move-up path satisfying the necessary conditions in Proposition 4,* $J(c,n),\ c = 0, 1, ..., n-1$ *is a strictly decreasing function of* λ*.*

Proof. Note that, by definition, we have

$$J(c,n) = \begin{cases} r(n), & c = n \\ (1 - e^{-\lambda \Delta t_{(c,c+1)}}) r(c) + e^{-\lambda \Delta t_{(c,c+1)}} J(c+1,n) & c = 0, 1, ..., n-1 \end{cases}$$

so

$$\begin{aligned} \frac{dJ(c,n)}{d\lambda} &= \Delta t_{(c,c+1)} e^{-\lambda \Delta t_{(c,c+1)}} (r(c) - J(c+1,n)) \\ &\quad + e^{-\lambda \Delta t_{(c,c+1)}} \frac{dJ(c+1,n)}{d\lambda}, \quad c = 0, 1, ..., n-1. \end{aligned}$$

When $c = n-1$, the term $J(c+1,n) = r(n)$ does not depend on λ, and so

$$\frac{dJ(n-1,n)}{d\lambda} = \Delta t_{(n-1,n)} e^{-\lambda \Delta t_{(n-1,n)}} (r(n-1) - r(n)) < 0$$

by Proposition 4 (ii).

Now suppose for some $c < n-1$, we have $\frac{dJ(c+1,n)}{d\lambda} < 0$, and so

$$\frac{dJ(c,n)}{d\lambda} < \Delta t_{(c,c+1)} e^{-\lambda \Delta t_{(c,c+1)}} (r(c) - J(c+1,n)) < 0$$

by Proposition 4 (i). The result for every $c = 0, 1, ..., n-1$ then follows by induction. $\square$

Proposition 6. *$V(k)$ is a non-increasing function of λ for all nodes $k \in N$.*

Proof. Given a call-arrival rate λ, let $D(k, \lambda)$ be the set of destination nodes for all paths $C(k, \lambda)$ starting from node k that satisfy the necessary conditions in Proposition 4. Modifying our notation to explicitly show the dependence on λ, we note that Proposition 5 shows $\lambda_1 < \lambda_2 \Rightarrow J(k, n, \lambda_1) > J(k, n, \lambda_2)$ for all destination nodes $n \in D(k, \lambda_1)$. We will shortly show that as λ increases, the set $D(k, \lambda)$ reduces in the sense that $\lambda_1 < \lambda_2 \Rightarrow D(k, \lambda_2) \subseteq D(k, \lambda_1)$. Thus, for $\lambda_1 < \lambda_2$ we have $\max_{n \in D(k,\lambda_1)} J(k, n, \lambda_1) > \max_{n \in D(k,\lambda_2)} J(k, n, \lambda_2)$. By definition, $V(k, \lambda) = \max(\max_{n \in D(k,\lambda)} J(k, n, \lambda), r_k)$, and so our result follows.

To show that $\lambda_1 < \lambda_2 \Rightarrow D(k, \lambda_2) \subseteq D(k, \lambda_1)$, we recall that $D(k, \lambda)$ is the set of destination nodes of paths satisfying the conditions in Proposition 4. Only the first condition, $J(c+1, n) > r(c)$, depends on λ. Proposition 5 shows that $J(c+1, n)$ is strictly decreasing in λ. Given that $r(c)$ is constant, the result follows immediately. $\square$

Proposition 7. *Let $\pi^*_\lambda(k)$ be an optimal move-up policy for a call-arrival rate λ. An optimal stand-by location $k : k = \pi^*_{\lambda_1}(k)$ for arrival rate λ_1 is also an optimal stand-by location for a higher call-arrival rate $\lambda_2 > \lambda_1$, i.e. $\pi^*_{\lambda_1}(k) = k, \lambda_2 > \lambda_1 \Rightarrow \pi^*_{\lambda_2}(k) = k$.*

Proof. If $\pi^*_\lambda(k) = k$, then we must have $V(k) = r(k)$, and $V(k') \leq V(k),\ \forall\ k' \in N_k$. As we increase λ, the policy of not moving will continue to give a reward of $r(k)$, while the rewards associated with neighbouring nodes $V(k'),\ k' \in N_k$, will be non-increasing (Proposition 6). Thus the optimal decision will not change, and so the result follows. $\square$

Example 2: Single-Ambulance Next-Call Move-up Model on a Network

We now apply the model to an undirected network of 35 nodes with a call-arrival rate $\lambda = 6$ calls/hour and one minute travel time for each arc. The target response time is assumed to be 2 minutes. The expected reward $r(k)$ at each node k for the next call is shown Table 3.1.

Figure 3.3 illustrates the optimal move-up policy. The two solid circles at nodes 24 and 33 represent two optimal stand-by locations. Under this optimal policy, the successor node

$r(1)$	0.0646	$r(10)$	0.0669	$r(19)$	0.0644	$r(28)$	0.2227
$r(2)$	0.1077	$r(11)$	0.0673	$r(20)$	0.0458	$r(29)$	0.1226
$r(3)$	0.1398	$r(12)$	0.0635	$r(21)$	0.0404	$r(30)$	0.1942
$r(4)$	0.2142	$r(13)$	0.0055	$r(22)$	0.0211	$r(31)$	0.3088
$r(5)$	0.2865	$r(14)$	0.0056	$r(23)$	0.0164	$r(32)$	0.3090
$r(6)$	0.1677	$r(15)$	0.0535	$r(24)$	0.3698	$r(33)$	0.3131
$r(7)$	0.1422	$r(16)$	0.0774	$r(25)$	0.3696	$r(34)$	0.2497
$r(8)$	0.0662	$r(17)$	0.0588	$r(26)$	0.3615	$r(35)$	0.1643
$r(9)$	0.0554	$r(18)$	0.0700	$r(27)$	0.3184		

Table 3.1: The expected reward $r(k), k = 1, 2..., 35$ for Example 2.

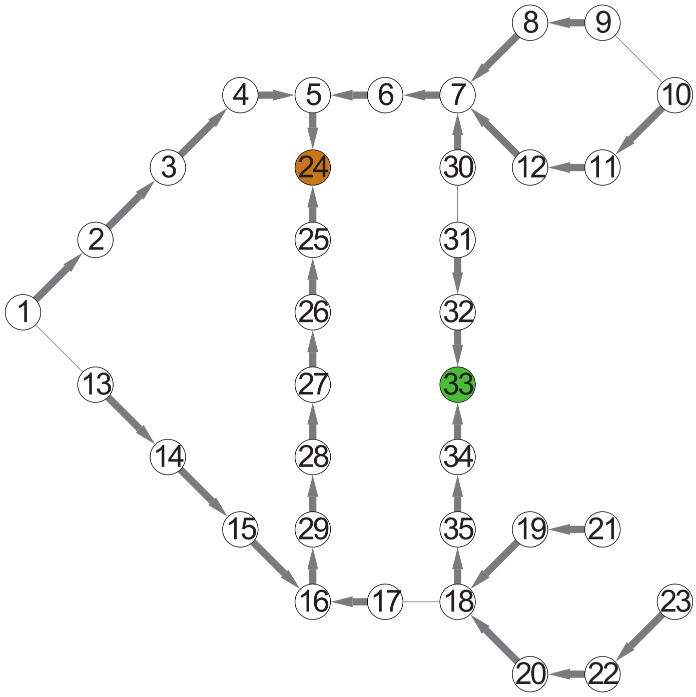


Figure 3.3: The optimal move-up policy indicated by arrows on a discrete network of 35 nodes. There are two optimal stand-by nodes, nodes 24 and 33.

given the current node is indicated by the arrow. We make the following two observations.

Remark 1 The move-up policy divides the network into separate trees with each optimal stand-by node forming the root of a tree.

Remark 2 An optimal move-up path may not be a shortest path. For example, for an ambulance at node 13, the shortest path in Figure 3.3 to node 24 is 13-1-2-3-4-5-24, but the optimal move-up path is 13-14-15-16-29-28-27-26-25-24. In our model, an ambulance can respond to the next call during move-up, and so the reward values $r(k)$ at both the destination and along the move-up path are important. From Table 3.1, we can see that the optimal move-up path is longer, but it includes more nodes (e.g. nodes 29, 28, 27, 26, 25, 24) with good expected reward values $r(k)$.

3.4 Solution Techniques

Algorithm 1 A label-setting algorithm for solving the single-ambulance next-call move-up problem

1 Assign an initial policy of staying put at every node k, i.e. put $V(k) = r(k), \ \forall \ k \in N$.

2 Define T to be the set of nodes with temporary labels. Initialise $T = V$.

3 Repeat

3.1 Set current node $u = \arg\max_{k \in T}(V(k))$. Designate the label on node u as permanent and remove u from set T.

3.2 For current node u, consider each temporary labeled adjacent node $h \in T \cap N_u \backslash u$ and update $V(h)$:

$$V(h) \leftarrow \max(V(h), (1 - e^{-\lambda \Delta t_{(h,u)}})r(h) + e^{-\lambda \Delta t_{(h,u)}} V(u))$$

Until every node is permanently labeled

To find an optimal policy π^*, we can use value iteration [22] to solve our DP model given by Equation (3.3). A sequence of value functions V^n is produced by starting from an arbitrary V^0, and defining

$$V^{i+1}(k) = \max_{k' \in N_k}((1 - e^{-\lambda \Delta t_{(k,k')}})r(k) + e^{-\lambda \Delta t_{(k,k')}} V^i(k')), \forall k \in N.$$

The sequence of functions V^i converges to V in the limit. A sensible set of starting values are $V^0(k) = r(k), \forall k \in N$. In this case, the complexity of value iteration is $|N|^2$. However, the fact that this problem is a stochastic shortest path problem [9] leads us to develop the label-setting procedure given in Algorithm 1 which can be viewed as a modification of Dijkstra's algorithm. At each iteration, the node with the maximum value is given a permanent label and used to update the values of its neighboring nodes. In this way, the trees rooted at each optimal stand-by locations are calculated node by node. Step 3 of the algorithm requires sorting and searching elements in a heap data structure. This leads to the complexity of $O(log|N|)$ for each iteration. This algorithm finds the objective value node by node in descending order of $V(k)$ with an overall complexity of $|N|log|N|$. The validity of this solution technique is proven in Appendix 1.

3.5 Summary

This chapter has been devoted to the study of a DP move-up model, which involves only one ambulance and aims to maximise the expected reward for the next call. Some insights were showed and a set of theoretical results were established. A label-setting algorithm was developed as an alternative to the standard DP solution technique – value iteration.

Maximising the benefit for just the next call may or may not be the 'right' thing to do. In the next chapter, we continue to follow the DP approach to study maximising the long-term performance when there is still only one ambulance in service.

4

Optimal Move-up of Single-Ambulance Infinite-Horizon Model

4.1 Overview

In the previous chapter, we presented a DP model for the single-ambulance next-call move-up problem. In this chapter, we consider a single-ambulance move-up problem which aims to maximise the average reward per unit time over an infinite horizon.

The problem is treated as a Semi-Markov dynamic programming model. In Section 4.2, we present the model assumptions and briefly discuss the difference between Markov Decision Processes(MDPs) and Semi-Markov Decision Processes(SMDPs). In Section 4.3, we describe the states in the model and how the system 'moves forward' over an infinite horizon. In Section 4.4, we formulate the optimality equations for the states in this model. In Section 4.5, we provide an alternative model with a reduced state space dimension. In Section 4.6, we compare results for optimal static policies, single-ambulance next-call move-up policies and single-ambulance infinite-horizon move-up policies under simplified scenarios. This chapter ends with a summary in Section 4.7.

4.2 Model Assumptions

We adopt the notations used for the single-ambulance next-call move-up model. In the next-call model, we restrict the ambulance's location to be a node on a discretised network. For the infinite-horizon move-up problem, we need to model the full response process including an ambulance driving to a call location and to a hospital. Thus we further assume that calls can only originate from the nodes on the network and hospitals must be located at nodes on the network. We redefine $p(k)$ to be the function that describes the spatial distribution of call demand at node $k \in N$ with $\sum_{k \in N} p(k) = 1$.

The single-ambulance operating system evolves in a discrete-time manner with a time-step size of one unit time. We assume that the travel time on each arc is an integral multiple of the unit time and the size of one time-step is small enough that the probability of more than one event occurring in this interval is insignificant. Here an event refers to a call arrival, a completion of on-site treatment or at-hospital hand-over.

As before, we use move-then-jump to model an ambulance traversing an arc $(i, j) \in L$: the ambulance waits at node i for a duration of $\Delta t_{(i,j)}$, then it jumps to node j; for a self-directed arc (i, i), we assume that the travel time is equal to one unit time.

We view the model, detailed shortly, as a Semi-Markov Decision Process based on the terminology used by White [45]. In [45], White discussed Markov Decision Processes (MDPs) and Semi-Markov Decision Processes (SMDPs). Briefly, a MDP refers to the case in which an action is taken at some time-step t, then at time-step $t + 1$, the system 'immediately' moves to a state according to the associated transition probability distribution. On the contrary, the system in a SMDP may not immediately move to a state after one time-step. Instead, it can take multiple time-steps, which may not be deterministic, to make a transition. Informally speaking, one can think that a MDP has no 'holding times' once an action is taken, while a SMDP is associated with possible holding times.

As stated above, the travel time for each arc $(i, j) \in L$ is an integral multiple of one time-step. This means that if the ambulance moves from node i to j, it may take one time-step or multiple time-steps to make a transition of reaching node j. Therefore, it is suitable to describe our model as a SMDP.

Regarding the at-scene service time, it is assumed to follow an exponential distribution with rate μ. At the conclusion of the at-scene treatment, we assume that the ambulance may, with probability $p_{\text{transport}}$, transport the patient to the closest hospital, or it may become free at the scene. Any service time required at hospital $h, h \in M$, where M denotes a set of nodes to represent hospitals, is also assumed to follow an exponential distribution with rate μ_h.

We assume that there is no queueing, so calls that arrive while the ambulance is busy are lost to the system. Finally, for simplicity, we assume that the travel time for each arc, arrival rate, response time target, on-site service rate, and at-hospital service rate are scaled according to one unit time.

4.3 State Space and Control

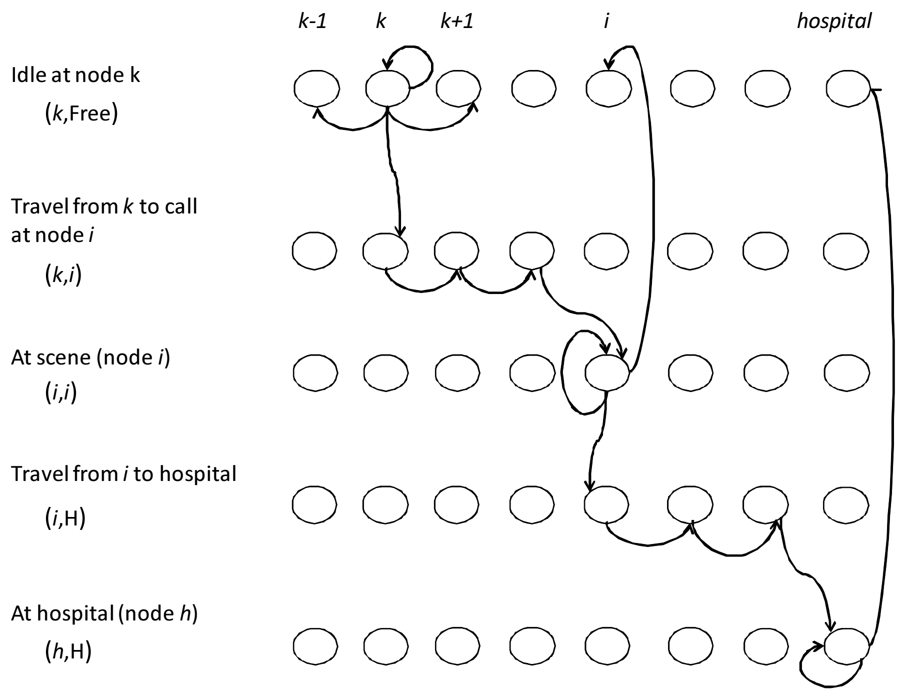


Figure 4.1: Example of the state space for the single-ambulance infinite-horizon move-up model.

Consider now the states required in our model. These states, as illustrated in Figure 4.1, track the steps in the typical response process described earlier in a discrete-time manner. State $(k, \text{Free}), k \in N$ indicates that the ambulance is free at node k. In such a state, we must determine the successor node $k' \in N_k$ which the ambulance moves to next. For each action, if there is no call arriving during the wait interval $\Delta t_{(k,k')}$, the ambulance jumps to node k' after $\Delta t_{(k,k')}$ time-steps. Otherwise, after γ time-steps where $\gamma = 1, ..., \Delta t_{(k,k')}$, the ambulance is dispatched to the random call location i with probability $e^{(\gamma-1)\lambda}(1-e^{\lambda})p(i)$ and reward $r(k)$. The system then enters state (k, i), meaning that the ambulance is at node k travelling on a shortest path to a call at location i. Arrival at the scene i is denoted by the state (i, i). This state indicates that treatment is being undertaken at the scene. Under the assumption of an exponential at-scene service time, after one time-step, the system will still be in state (i, i) with probability $e^{-\mu}$, or the treatment will have been completed. If the treatment completes, the system enters either state (i, Free) with probability $1 - p_{\mathrm{transport}}$, indicating the ambulance is now free at the scene, or state (i, H), indicating that the ambulance is at node i transporting a patient to the closest hospital along the shortest path. Assuming the closest hospital is at node $h = h(i)$, arrival at the hospital leads the system into state (h, H), indicating that the ambulance is at node h handing over the patient. After one time-step, the system may still be in state (h, H) with probability $e^{-\mu_h}$, or enter the free state (h, Free). Keep in mind that we model the ambulance movement using wait-then-jump not just for move-up but also for travelling towards a call location or a hospital.

4.4 Objective and Optimality Equations

We choose to maximise the average reward per time step over an infinite horizon as our objective. Therefore, we have an undiscounted DP model. For our problem, the state space forms a single recurrent class in which case the average reward is independent of the starting state [31]. Given a set of states S, let g denote the maximum average reward per time step, and $r(s, a)$ be the immediate reward given the system is at state $s \in S$ and action a from some action set $A(s)$ is taken. Using the theory of SMDP's [45], the optimality equation for each state $s \in S$ can be written as follows:

$$g = \max_{a \in A(s)} \Big[\frac{r(s,a) + \sum_{w \in S, 1 \le \gamma \le L} P(s,a,w,\gamma)V(w) - V(s)}{\sum_{w \in S, 1 \le \gamma \le L} P(s,a,w,\gamma)\gamma} \Big]. \tag{4.1}$$

The term $r(s,a)$ is the immediate reward given the system is at state s and action a from action set $A(s)$ is taken. The term γ denotes the time-steps taken to the next state, where $\gamma = 1, ..., L$ and L is the maximum possible number of time-steps required to make a transition. The term $P(s,a,w,\gamma)$ denotes the probability of making a transition to state w after γ time-steps for action a at state s. The (non-zero) denominator in (4.1) is the expected number of time-steps taken to the next state.

The value function $V(s)$ is often referred to as the *relative value* function. The difference $V(s_1) - V(s_2)$ represents the extra gain (loss) in the long term by starting in state s_1 as opposed to state s_2. Note that the difference $V(s_1) - V(s_2)$ is independent of any absolute level. For a good interpretation of relative values, we refer the reader to Howard [22].

The second form is a rearrangement of the first form as follows:

$$\begin{aligned} V(s) &= \max_{a \in A(s)} \Big[r(s,a) + \sum_{w \in S, 1 \leq \gamma \leq L} P(s,a,w,\gamma) V(w) \\ &\quad -g \sum_{w \in S, 1 \leq \gamma \leq L} P(s,a,w,\gamma)\gamma) \Big]. \end{aligned} \tag{4.2}$$

We now provide the optimality equation (4.2) for every state in our model. First we consider each 'Free' state $(k, \text{Free}), \forall k \in N$ – the ambulance is available at node k. The ambulance can move to any node $k' \in N_k$ in which case the non-zero transition probabilities associated with such a state are:

$$P\{(k', \text{Free})|(k, \text{Free})\} = e^{-\lambda \Delta t_{(k,k')}},$$
$$P\{(k,i)|(k, \text{Free})\} = (1 - e^{-\lambda \Delta t_{(k,k')}})p(i) \quad \forall i \in N,$$

where $e^{-\lambda \Delta t_{(k,k')}}$ is the probability of no call arriving during the time interval $\Delta t_{(k,k')}$ while the ambulance is still at node k, and $(1 - e^{-\lambda \Delta t_{(k,k')}})p(i)$ is the probability that a call arrives at node i during the time interval $\Delta t_{(k,k')}$. The immediate reward when in state (k, Free) is

$$(1 - e^{-\lambda \Delta t_{(k,k')}})r(k).$$

The expected number $E(k, k')$ of time-steps to either get dispatched to a call from node k or reach node k' is

$$e^{-\lambda \Delta t_{(k,k')}} \Delta t_{(k,k')} + \sum_{\gamma=1}^{\Delta t_{(k,k')}} e^{-\lambda(\gamma-1)}(1 - e^{-\lambda})\gamma,$$

where the term $e^{-\lambda(\gamma-1)}(1-e^{-\lambda})$ represents the probability that the ambulance is dispatched at time-step γ. Therefore (4.2) gives:

$$\begin{aligned} V(k, \text{Free}) = \max_{k' \in N_k} \Big[&(1 - e^{-\lambda \Delta t_{(k,k')}}) r(k) + (1 - e^{-\lambda \Delta t_{(k,k')}}) \sum_{i \in N} p(i) V(k, i) \\ &+ e^{-\lambda \Delta t_{(k,k')}} V(k', \text{Free}) - gE(k, k') \Big], \quad \forall k \in N. \end{aligned} \tag{4.3}$$

Consider next states $(k, i), \forall k, i \in N, k \neq i$, and states $(k, \text{H}), \forall k \in N, k \neq h(k)$, in which the ambulance is travelling from node k to (but has not yet reached) i to serve a call, or travelling to (but has not yet reached) the closest hospital at node h. There is only one transition state from each of these states, being to move to the next node along the shortest path. There is no immediate reward for either of these states, Thus (4.2) gives

$$V(k, i) = V(\text{next}(k, i), i) - g\Delta t_{(k,\text{next}(k,i))}, \forall k \in N, i \in N, k \neq i. \tag{4.4}$$

$$V(k, \text{H}) = V(\text{next}(k, h(k)), \text{H}) - g\Delta t_{(k,\text{next}(k,h(k)))}, \forall k \in N, k \neq h(k) \tag{4.5}$$

where $\text{next}(k, j)$ represents the successor of node k along the shortest path from k to j. For states $(i, i), \forall i \in N$, in which the ambulance is treating at the scene, the non-zero transition probabilities are:

$$P\{(i, i)|(i, i)\} = e^{-\mu}$$

$$P\{(i, \text{Free})|(i, i)\} = (1 - e^{-\mu})(1 - p_{\text{transport}})$$

$$P\{(i, \text{H})|(i, i)\} = (1 - e^{-\mu}) p_{\text{transport}}.$$

There is no immediate reward for such states and so (4.2) gives

$$\begin{aligned} V(i,i) &= e^{-\mu}V(i,i) + (1-e^{-\mu})((1-p_{\text{transport}})V(i,\text{Free}) \\ &\quad +p_{\text{transport}}V(i,\text{H})) - g, \quad \forall i \in N. \end{aligned} \tag{4.6}$$

The last states $(h, \text{H}), \forall h \in M$, in which the ambulance is at hospital node h handing over the patient, have the following associated non-zero transition probabilities:

$$P\{(h,\text{H})|(h,\text{H})\} = e^{-u_h},$$
$$P\{(h,\text{Free})|(h,\text{H})\} = 1 - e^{-u_h}.$$

There is no immediate reward for such states and so (4.2) gives:

$$V(h,\text{H}) = e^{-u_h}V(h,\text{H}) + (1-e^{-u_h})V(h,\text{Free}) - g, \ \forall h \in M. \tag{4.7}$$

4.5 Optimality Equations with State Space Reduction

The state space for the DP model above includes a state (k, i) for every pair of nodes $k \in N$, $i \in N$, and so the state space size is of order $|N|^2$. We now show that this state space can be reduced to a size of order $|N|$, eliminating the need to store many intermediate states and therefore allowing problems with a large number of nodes to be solved more efficiently. By carefully observing Equations (4.3)-(4.7), we can see that if the ambulance gets dispatched from some node k to a call at node i, it follows a deterministic path to the call location and then becomes free again either at the call location or at a hospital after some service time. Therefore, using (4.4)-(4.7), we can rewrite $V(k, i)$ in terms of the average reward g, the value $V(i, \text{Free})$, and the value $V(h(i), \text{Free})$ corresponding to the closest hospital location

$h(i)$:

$$\begin{aligned} V(k,i) &= (1-p_{\text{transport}})V(i,\text{Free}) + p_{\text{transport}}V(h(i),\text{Free}) \\ &\quad -(d_{k,i}+\frac{1}{1-e^{-\mu}})g \\ &\quad -p_{\text{transport}}(d_{i,h(i)}+\frac{1}{1-e^{-\mu_{h(i)}}})g, \end{aligned} \tag{4.8}$$

where $d_{k,i}$ is the number of time steps needed to travel from node k to node i along a shortest path. This equation can be interpreted as follows. The first two terms on the right hand side show possible transitions from the point of being dispatched to becoming free upon the completion of a service. If no transport to hospital is required, the ambulance will become free at the scene (node i); otherwise the ambulance will become free at the closest hospital to the scene (node $h(i)$). The third and fourth terms give the loss of reward due to ambulance unavailability. The third term shows the loss arising from time spent travelling to the call location and completing the on-site treatment, while the fourth term gives the further losses that occur while travelling to the hospital and handing over the patient at the hospital. Note that in Equation (4.8), the term $\frac{1}{1-e^{-\mu}}$ is an approximation of the expected number of time steps $\frac{1}{\mu}$ required to complete the on-site treatment. This approximation arises from our earlier use of discrete time steps. As the service rate μ reduces (corresponding to a finer time discretisation), the approximate value gets closer to the exact value. Similarly, $\frac{1}{1-e^{-\mu_h}}$ is also an approximation of $\frac{1}{\mu_h}$.

We can substitute (4.8) into (4.3), to give

$$\begin{aligned} V(k,\text{Free}) = \max_{k'\in N_k} \Bigg[&(1-e^{-\lambda\Delta t_{(k,k')}})\Big[r(k) + (1-p_{\text{transport}})\sum_{i\in N} p(i)V(i,\text{Free}) \\ &+ p_{\text{transport}}\sum_{i\in N} p(i)V(h(i),\text{Free})) - \left(\sum_{i\in N} p(i)d_{k,i} + \frac{1}{1-e^{-\mu}}\right) g \\ &- p_{\text{transport}}\left(\sum_{i\in N} p(i)(d_{i,h(i)} + \frac{1}{1-e^{-\mu_{h(i)}}})\right) g\Big] \\ &+ e^{-\lambda\Delta t_{(k,k')}} V(k',\text{Free}) - gE(k,k')\Bigg], \quad \forall k\in N. \end{aligned} \tag{4.9}$$

This gives a set of optimality equations (4.9) that more concisely describe our infinite-horizon single-vehicle model. We now have a much reduced state space $S = \{(k, \text{Free}), k \in N\}$ with only $|N|$ states that is more computationally attractive to solve. Solutions generated using this model are presented next.

4.6 Computational Experiments and Insights

In this section, we perform experiments with a set of simplified scenarios. Using the (long-term) objective of the infinite-horizon move-up model as the performance measure, the results for the optimised static policies, next-call move-up policies and infinite-horizon move-up policies are compared, and empirical findings and insights are discussed. Similar analysis is performed in the next chapter involving two ambulances in service.

In total, 250 scenarios are constructed for the experiments. All scenarios use: a network consisting of 50 nodes on a single road G with 1 minute spacing and a hospital at node 25; a response time target W of 3 minutes; a transport probability $p_{\text{transport}}$ of 0.7; an on-site service rate μ of $\frac{1}{4}$ calls/min, and an at-hospital service rate μ_h of $\frac{1}{20}$ calls/min. The arrival rate λ is varied from $\frac{1}{600}$ calls/min to $\frac{1}{120}$ calls/min with a step size of $\frac{1}{600}$ calls/min, i.e. $\lambda = 0.1, ..., 0.5$ calls/hr. A set of 50 spatial distributions of call demand are generated randomly. Each combination of arrival rate λ and spatial distribution p stated above creates one scenario – therefore, we have 250 scenarios for experiments.

To create one spatial distribution p of call demand, we choose 6 nodes 1, 14, 24, 36, 47, 50 and randomly choose values for these nodes sampled from a continuous uniform distribution. We then construct a piece-wise linear interpolant using the function values at the 6 nodes. After normalising the interpolated values at the 50 nodes, the spatial distribution p of call demand is uniquely determined and the reward function r can also be computed.

For all the 250 scenarios, we assume that the size of one time-step is equal to 1 minute. Therefore, the infinite-horizon move-up model (SMDP) for each scenario becomes a simpler MDP. To solve optimality equations for a MDP/SMDP, one can use value iteration, policy iteration, or linear programming (LP) as discussed in White [45]. For our experiments, we choose to use LP as the solution technique. The LP formulation is as below.

$$
\begin{aligned}
&\text{maximise } g \\
\text{s.t. } V(k,\text{Free}) \quad &\geq \quad (1-e^{-\lambda\Delta t})\Big[r(k) + (1-p_{\text{transport}})\sum_{i\in N} p(i)V(i,\text{Free}) \\
&\quad +p_{\text{transport}}\sum_{i\in N} p(i)V(h(i),\text{Free})) - \left(\sum_{i\in N} p(i)d_{k,i} + \frac{1}{1-e^{-\mu}}\right) g \\
&\quad -p_{\text{transport}}\left(\sum_{i\in N} p(i)(d_{i,h(i)} + \frac{1}{1-e^{-\mu_{h(i)}}})\right) g\Big] \\
&\quad +e^{-\lambda\Delta t}V(k',\text{Free}) - gE(k,k'), \quad \forall k \in N,\ \forall k' \in N_k \\
V(k_0,\text{Free}) \quad &= \quad 0,
\end{aligned}
$$

where $(k_0, \text{Free}) \in S$ is a reference state.

The code for this LP problem is written in AMPL and solved using CPLEX 11.0.0. We compare the performance of the single-ambulance infinite-horizon move-up model, the simpler next-call move-up model and the static model. For each scenario, we modify the infinite-horizon model (code) to determine the best static policy by nominating each node in turn as the home base. Similarly, we modify the model (code) to evaluate the optimal next-call move-up policy (obtained from the model in the previous chapter) over an infinite horizon.

The utilisation varies between, approximately, 0.3 and 0.55 for these scenarios. To calculate the utilisation under a given move-up/static policy, we evaluate the average number of calls reached per time-step using the reward function $r(k) = 1, k = 1, ..., 50$. Let g^e be generic notation for the result of this evaluation process; as the modified reward function implies that every call is reached on time, g^e is equivalent to the average number of calls served per time-step. Therefore, the probability of a call being lost due to the ambulance being busy is $1 - \frac{g^e}{\lambda}$; in queueing theory, this probability is referred to as the Erlang loss probability. Since there is only one ambulance in service, it also represents the utilisation of the ambulance.

To measure the effectiveness of each next-call/infinite-horizon move-up policy, we compute the *relative improvement* which is defined as $\frac{g-g^b}{g^b}$ where g is the average number of calls reached on time per time-step under the move-up policy and g^b is the corresponding average under the static policy [1]. Table 4.1 summarises the statistics for relative improvement; the 250 scenarios are divided into 5 groups based on their respective arrival rate.

Obviously, with respect to the objective of the infinite-horizon move-up model, an optimal infinite-horizon move-up policy will always perform at least as well as a next-call move-up policy and a static policy; its relative improvement is always non-negative. From Table 4.1, we see that on average, the next-call move-up model performs worse than the static model in 4 out of 5 scenario groups; the infinite-horizon move-up model gives better performance than the static model in all 5 groups and the benefit increases as λ increases. The number of scenarios for which the associated next-call move-up policy gives the same performance as the static policy is 48, 43, 40, 40 and 38 out of 50, respectively, for each λ; the number of scenarios for which the infinite-horizon move-up policy and the static policy perform equally well is 48, 46, 46, 45 and 43 out of 50, respectively.

We find that if the optimal infinite-horizon move-up policy performs better than the corresponding static policy for a given call distribution $p()$, it is also true for the scenarios that have higher arrival rates while all other input parameters are the same. We do not see such a pattern for the next-call move-up model. Furthermore, for a scenario whose performance is improved using the infinite-horizon move-up model, the next-call move-up model may not improve the performance on the static model and may even worsen the performance.

We also observe that when the next-call/infinite-horizon move-up model performs better than a static model, the move-up policy involves multiple optimal stand-by nodes. The ambulance moves to one of the stand-by nodes and avoids travelling through low reward sections of the road. This behavior was observed in the previous chapter as well.

The results suggest that when the home base is chosen properly, the associated static policy is expected to perform as well as an optimised next-call move-up policy while an

[1] For simplicity, we overload the term g to represent the average reward over an infinite horizon under a next-call policy.

optimised infinite-horizon move-up policy is expected to be the most beneficial. Meanwhile, we see that the chance that the three types of location strategies give the same performance is relatively high when λ is low (which was expected when introducing the next-call move-up model) and decreases as λ increases. The observation of 'pointless moves' with respect to the next-call move-up model may seem discouraging, as our goal is to improve performance with move-up. However, it is important to know that a move-up strategy may actually worsen the performance, so alternatives such as the infinite-horizon move-up model deserve to be explored.

The small relative improvement (the average is less than 0.08% and the maximum is less than 1.1%) using the infinite-horizon move-up model is explained as follows. When the ambulance becomes free at the hospital (with probability 0.7), the optimal stand-by node starting from the hospital location is the same under the move-up policy and the static policy. In other words, the benefit derived from move-up is solely from the 30% of calls where the ambulance becomes free on site.

We follow the same argument to consider the case in which the probability $p_{\text{transport}}$ is 1.0: although an optimal infinite-horizon move-up policy specifies the optimal stand-by node for every node at which the ambulance becomes free, the ambulance can only become free at the hospital after a completion of service; then it drives from the hospital to the optimal stand-by node. This is exactly the same way an ambulance behaves under a static policy. Therefore, the move-up policy and the static policy must always lead the ambulance to the same optimal stand-by node, meaning that they perform equally well in the long term.

On the other hand, if we reduce $p_{\text{transport}}$, meaning that the ambulance is more likely to become free on site, then a more intelligent move-up policy is expected to provide an increased benefit. Table 4.2 shows the statistics of the relative improvement with $p_{\text{transport}} = 0$ in which case the ambulance always becomes free on site. All the infinite-horizon move-up policies, next-call move-up policies and static policies are re-optimised. We see that the average and maximum relative improvement figures for the infinite-horizon model are both increased compared to the results in Table 4.1.

λ (calls/min)	1/600		1/300		1/200		1/150		1/120	
Policy	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite
Min. RIM	-0.00295	0	-0.00130	0	-0.00723	0	-0.01218	0	-0.01632	0
Avg. RIM	-0.00003	0.00006	0.00006	0.00012	-0.00003	0.00023	-0.00009	0.00036	-0.00029	0.00080
Max. RIM	0.00169	0.00169	0.00377	0.00378	0.00581	0.00581	0.00778	0.00779	0.00971	0.01044
No. of RIM>0	1	2	3	4	4	4	5	5	6	7
No. of RIM<0	1	0	4	0	6	0	5	0	6	0
No. of RIM$=0$	48	48	43	46	40	46	40	45	38	43

Table 4.1: The statistics for relative improvement (RIM) for the optimal next-call and infinite-horizon move-up policies when compared with the optimal static policies based on the 250 scenarios. The scenarios are divided into 5 groups of 50 scenarios according to the arrival rate λ.

Note that the assumption of the ambulance always becoming free on site is not realistic, it is simply used to show that the infinite-horizon move-up model is expected to be more beneficial when there are more opportunities to behave differently with respect to the static model.

We also make a note regarding the performance of the re-optimised next-call move-up policies: on average, the next-call model performs better than the static model in all 5 scenario groups. Recall that it performed worse than the static model in 4 out of 5 groups before (Table 4.1). Therefore, it is hard to draw conclusions about the performance of this model. It is clear that focusing on the next call will never outperform an optimal infinite-horizon move-up policy. However, the next-call move-up model can be extended to multiple ambulances in an approximate manner for realistic-sized problems; the infinite-horizon move-up model is hard to extend for practical problems.

We envision one form of approximate next-call move-up for large-scale EMS operations as follows. For each number of free ambulances, a set of 'good' configurations (stand-by locations) are generated based on some heuristic method. The reward for the next call in each configuration is therefore treated as input. At each move-up time instant, we evaluate the expected reward for moving to each configuration associated with the current number of free ambulances in a similar manner to the one-ambulance case which uses Equation 3.2; the configuration giving the best reward is the 'optimal' decision. A mechanism of allocating ambulances into a configuration is needed for this evaluation process. A reasonable mechanism would be minimising the total travel times.

4.7 Summary

This chapter has been devoted to the study of optimal move-up for one ambulance in order to maximise long-term performance. A DP model was developed for this infinite-horizon move-up problem. The structure of the states in the DP model was explored to formulate an alternative model which reduces the size of the state space from $|N|^2$ to $|N|$ where N is the set of nodes on the network. A modified value iteration algorithm was developed to

λ (calls/min)	1/600		1/300		1/200		1/150		1/120	
Policy	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite	Next-call	Infinite
Min. RIM	0	0	-0.00215	0	-0.00904	0	-0.01521	0	-0.02067	0
Avg. RIM	0.00011	0.00018	0.00024	0.00037.	0.00034	0.00071	0.00049	0.00114	0.00067	0.00202
Max. RIM	0.00554	0.00556	0.123	0.0123	0.01878	0.01879	0.02499	0.025	0.03093	0.03095

Table 4.2: The statistics for relative improvement (RIM) for the optimal next-call and infinite-horizon move-up policies when compared with the optimal static policies based on the 250 scenarios after reducing the value of $p_{\text{transport}}$ from 0.7 to 0. The scenarios are divided into 5 groups of 50 scenarios according to the arrival rate λ.

solve this alternative model.

Simplified scenarios were used to compare the performance of the optimal static policies, next-call move-up policies, and infinite-horizon move-up policies. The results suggested that the performance of the 'short-term' next-call move-up model was hard to appraise; on the other hand, the infinite-horizon move-up model could provide increasing benefit compared with the static model as the arrival rate increases. This encourages us to extend the infinite-horizon move-up model to two ambulances in the next chapter.

5

Optimal Move-up in a Two-Ambulance Infinite-Horizon Model

In the previous chapter, we presented the single-ambulance infinite-horizon move-up model. In this chapter, we consider the extension to a two-ambulance infinite-horizon move-up model under the DP framework. Because the system with two ambulances is more complicated than that with only one ambulance, the two-ambulance model is based on a simpler Markov Decision Process, i.e. the travel time on each arc is equal to one unit time. We use this model to gain some understanding of interactions/cooperations between ambulances, which cannot be investigated using the single-ambulance move-up models discussed in the previous two chapters. Note that, it is an easy modification to model the two-ambulance SMDP (in which the travel time on each arc is an integral multiple of one unit time). However, we expect to see insights that are similar to those gained from the simpler Markov dynamic programming model.

The most relevant work in the literature is the two-server move-up model by Berman [5]. That model was developed in a more generic setting involving mobile servers traveling to 'customers'. In other words, it was not specifically designed to approximate ambulance operations. Consequently, some important characteristics unique to ambulance operations were not incorporated. We summarise the model assumptions in [5] below and then contrast them to our model assumptions.

In Berman's work [5], each server has a pre-determined home base. After an exponentially distributed service time, it returns to its home base and only then is considered available. The service rate associated with each ambulance depends on its home base location. Note that the service time includes the duration of travelling back to a home base.

A move-up policy based on Berman's model can be summarised as below. When only one ambulance is free, it can move to any location on the network along a shortest path for stand-by; moreover, it must arrive at the target location before it is allowed to move to another location (assuming it is not dispatched while travelling to the target location). When both ambulances are free, they must stand by at their respective home bases.

In contrast to solving a move-up problem in a generic mobile-server setting, we construct our DP model to specifically approximate ambulance operations. Our model assumes that an ambulance becomes free either on site or at a hospital, which is a key characteristic of ambulance operations. There are no home bases. As in the single-ambulance models, a move-up decision for each ambulance is made node-by-node instead of deciding a target node and then following a shortest path to that node. In addition, we model the service process for a call more carefully, i.e. the location and status of the dispatched ambulance is tracked step-by-step. Instead of assuming an exponentially distributed service time as in [5], travel times to a call location, on-site treatment times, travel times to a hospital, and at-hospital hand-over times are modelled separately. Overall, we think the model we propose gives a more realistic environment for the study of optimal move-up policies.

This chapter is organised as follows. In Section 5.1, we describe the problem assumptions. In Section 5.2, we present the state space in this DP model. In Section 5.3, we use simple examples to discuss insights obtained from the optimal move-up policies, followed by a summary in Section 5.4.

5.1 Problem Assumptions

We follow the notation that we used for the single-ambulance infinite-horizon move-up model. Because we now have two ambulances in the system, we introduce a new reward function $r(k_1, k_2), \forall k_1, k_2 \in N$ which is the probability of reaching the next call on time given that the two ambulances are free at nodes k_1 and k_2 when a call arrives. We assume the two ambulances are identical and if both ambulances are free when a call arrives, the closest ambulance is dispatched. If there is a tie, each ambulance has an equal probability (0.5) of getting dispatched. As before, we assume no queueing, so calls that arrive while both ambulances are busy are lost to the system.

The reward function $r(k)$ used in the previous two chapters now is redefined as the probability of reaching the next call on time given that there is only one free ambulance and it is at node k.

5.2 State Space and Control

The state space S for two ambulances is a natural extension of the state space for the single-ambulance infinite-horizon move-up model presented in the previous chapter. A state is denoted by a vector of two elements, (a_1, a_2), where a_1 represents the state of the first ambulance and a_2 represents the state of the second ambulance. The state space for a single ambulance, as described in Section 4.3, tracks the steps in the typical response process. The state space S for two ambulances consists of all possible pairs of states for a single ambulance. The size of S is in the order of $|N|^4$ where N is the set of nodes on the network. The states that require a move-up decision are those in which both ambulances are free or just one ambulance is free. A free ambulance at some node k can move to any adjacent node $k' \in N_k$.

We leave the optimality equations for this DP model in Appendix 2, as it is a trivial extension from the single-ambulance infinite-horizon move-up model. The objective is the same as that for the single-ambulance infinite-horizon move-up model – to maximise the average (undiscounted) reward per time-step. Keep in mind that the optimality equations yield an optimal move-up policy, the corresponding objective value, and relative values representing the difference in the long-term total reward by starting from one state as opposed to another.

Theoretically, this DP model can be further extended for more ambulances. However, it is computationally infeasible even for two ambulances on realistic networks. The main use of this DP approach is to gain insights into the properties of optimal move-up policies and value functions on a small scale. We then use these insights to guide the development of large-scale approximate models with the hope that performance can be improved on the static ambulance location strategy.

5.3 Computational Experiments and Insights

In this section, we construct a set of scenarios for computational experiments. Firstly, we compare the performance of the optimised move-up policies and static policies. Secondly, we present some moves sampled from the optimised move-up policies, which we would expect to observe in practical problems and then we show some characteristics of the relative values.

In total, 120 scenarios are constructed for the experiments. All scenarios use: a network consisting of 15 nodes on a single road with 1 minute spacings; a spatial distribution $p(k), k = 1, ..., 15$, of call demand shown in Figure 5.1; a response time target W of 2 minutes; a transport probability $p_{\text{transport}}$ of 0.8; an on-site service rate μ of $\frac{7}{12}$ calls/min; a hospital with an at-hospital service rate μ_h of 0.5 calls/min. The difference between these scenarios is the arrival rate λ and/or the hospital location h. The arrival rate λ is varied from $\frac{1}{60}$ calls/min to 1 call/min with a step size of $\frac{1}{60}$ calls/min; the hospital location h is either node 4 or node 11. Each combination of λ and h creates one scenario.

We make a few notes regarding the response time target, on-site and at-hospital service rates. The response time target is relatively small compared to that in reality, so that we have distinguishable rewards for a call at different nodes on this small network.

The on-site and at-hospital service rates are considerably higher than the rates encountered in practice. We find that if the service rates are close to reality, when one ambulance gets dispatched and the other one remains free, the dispatched ambulance is typically busy for long enough that the free ambulance behaves as under an optimised single-ambulance infinite-horizon move-up policy. In other words, the busy ambulance has no impact on where the free ambulance should stand by. We do not think this is an interesting case to

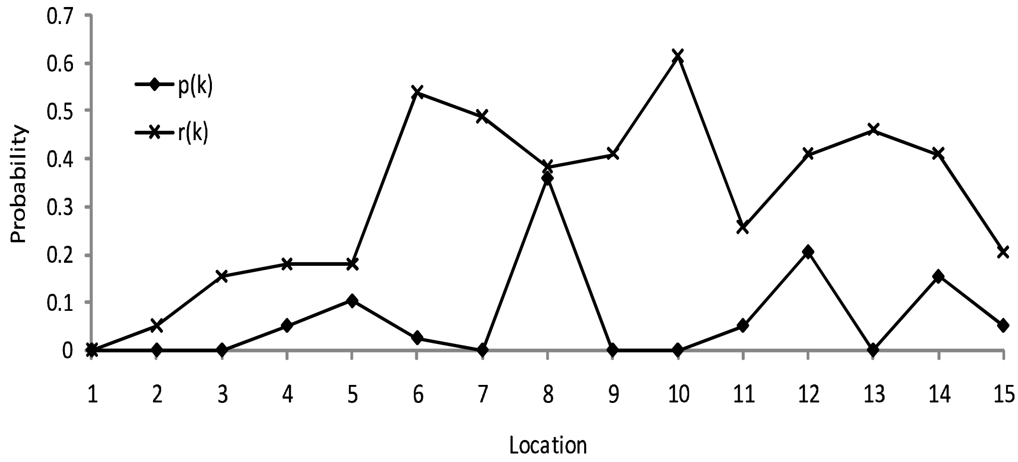


Figure 5.1: Spatial distribution $p(k)$ of call demand and the reward function $r(k), k = 1, ..., 15$.

study.

For typical large-scale EMS operations, we expect that (1) the chance of freeing up (at least) one ambulance in the near future is relatively high because of the ambulance fleet size and the workload, and (2) where a busy ambulance is expected to become free may play an important role in an optimal move-up policy. Therefore,we purposely choose these high rates to increase the chance of a busy ambulance becoming free in the near future. So in some sense, we are attempting to imitate the real-world ambulance operations by speeding up the service process in the small-scale setting.

To obtain an optimal move-up policy, we use the value iteration algorithm [45]. To obtain an optimal static policy, we modify the DP model to allow fixed base locations to be given, and the policy iterations are performed to evaluate the system performance. Every pair of nodes $k_1 = 1, ..., 15$ and $k_2 = k_1, ..., 15$ is tested as the home bases for the two ambulances. The pair that gives the best performance defines the optimal static policy.

To analyse our move-up policies in detail, we have developed a simulation tool in C#. The tool is also used to estimate the utilisation associated with each static/move-up policy, using a set of artificial calls generated based the assumptions above for each scenario. The utilisations range from about 0.03 to about 0.85 for our experiments.

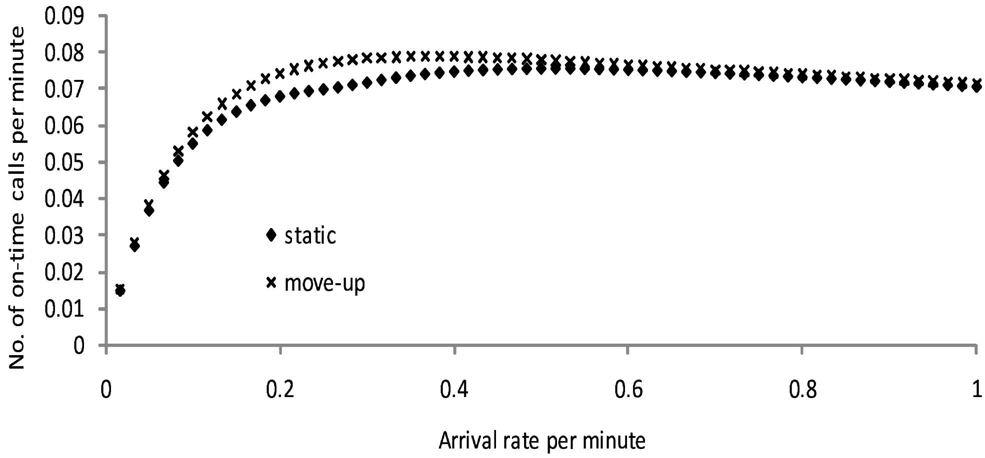
5.3.1 Performance Comparisons of the Optimised Move-up and Static Policies

We plot the objective values associated with the optimal move-up policies and the optimal static policies in Figure 5.2 (a)-(b). Figure 5.2 (a) corresponds to a group of 60 scenarios, which have the hospital at node 11 but varying arrival rates; Figure 5.2 (b) corresponds to the other 60 scenarios, which have the hospital at node 4. The x-axis shows the arrival rate for each scenario; the y-axis shows the objective value. Recall that the objective value represents the average number of calls reached per unit time (minute).

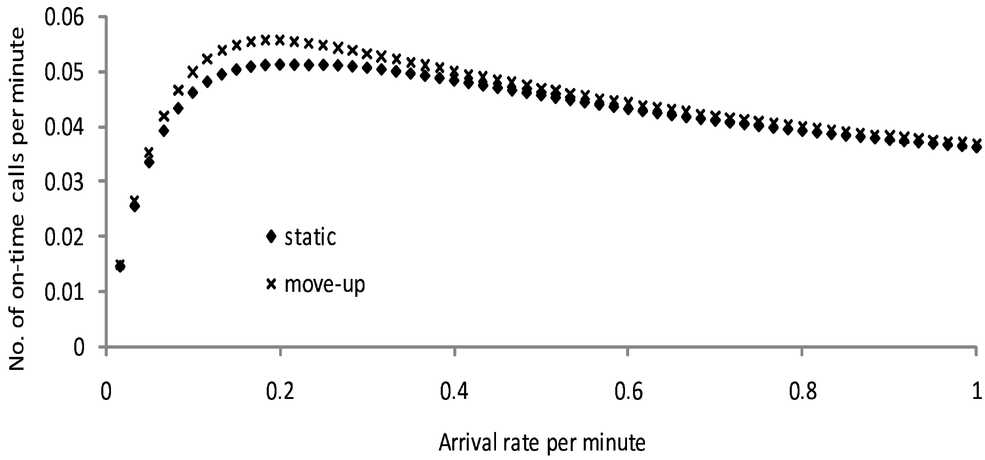
In Figure 5.2 (a), we see that both of the objective values, under the move-up and static policies, increase first, and then decrease. However, although it is not shown here, the ratio $g(\lambda)/\lambda$ where $g(\lambda)$ is the objective function value for each move-up/static policy, being the average number of calls reached on time per time-step, decreases as λ increases. This ratio represents the expected percentage of calls reached on time, and we expect that the ratio keeps decreasing as the system workload increases. It is easy to see that when λ is large enough, the average number of calls reached per time-step becomes a constant under any move-up/static policy. This is because a newly-freed ambulance immediately becomes busy again from where it becomes free; there is little time to perform move to any location.

In Figure 5.2 (b), we also see that the objective values increase and then decrease. However, this pattern is more pronounced in Figure 5.2 (b) than Figure 5.2 (a). Furthermore, give the same strategy, i.e. static or move-up, Figure 5.2 (a) and 5.2 (b) show that the performance under scenarios with $h = 11$ is better (or no worse) than that under scenarios with $h = 4$ for any given λ value.

More specifically, when λ is low, the performance is insensitive to the hospital location h. This is because the two ambulances are predominantly free and have enough time to reach the nodes giving the best reward for the next call regardless of where they become free. When λ increases, we find that the scenarios with $h = 11$ give better performance than those with $h = 4$. This is because, with an increased λ, it is more likely that only one ambulance is free for the next call-arrival and it may not have enough time to reach the best-reward location before the call-arrival. The location at which each ambulance becomes



(a) Scenarios with $h = 11$



(b) Scenarios with $h = 4$

Figure 5.2: Objective values under the optimal move-up policies and static policies for the 120 scenarios

free is more important to the overall performance. Given the relatively high probability of transportation, i.e. $p_{\text{transport}} = 0.8$, a busy ambulance is most likely to become free at the hospital. As shown in Figure 5.1, when $h = 4$, the ambulance that has just finished transportation is at the fourth worst location (node 4) in terms of $r(k)$ (the one-free-ambulance reward), 2 minutes away from the second best location (node 6) and 6 minutes away from the best location (node 10). On the other hand, when $h = 11$, the ambulance that has just finished transportation (at node 11) is only 1 minute away from the best location (node 10) and the rewards near node 11 are also reasonably high.

To summarise, when a busy ambulance becomes free under scenarios with $h = 11$, it is closer to high-reward locations than under scenarios with $h = 4$. Therefore, the performance under scenarios with $h = 11$ is generally better.

As in Section 4.6, we compute the relative improvement to quantify the benefit from move-up. Recall that the relative improvement is equal to $\frac{g-g^b}{g^b}$ where g and g^b represent the objective value under an optimal move-up policy and the corresponding optimal static policy, respectively. For all the scenarios, we find that move-up always leads to better performance. The relative improvement figures also increase first and then decrease as λ increases. Intuitively speaking, when λ is small, both ambulances are most likely to be free and stand by at the nodes giving the best reward for the next call, which are the same under both the static and move-up policies. Therefore, move-up makes little difference. When λ is large, as discussed earlier, there is little opportunity for performing more intelligent move-up, which is why the size of the improvement is also small.

Summarising, the minimum relative improvement is 0.7%, which occurs with $\lambda = 1$ call/min and $h = 11$; the average relative improvement over all scenarios is 2.1%; the maximum relative improvement is 9.8%, which occurs with $\lambda = \frac{4}{15}$ calls/min and $h = 11$.

5.3.2 Example Moves Using Optimal Move-up Policies

The move-up policies even for these small-scale problems contain a large number of decisions and they are difficult to report. Therefore, we review some moves that occur under the optimal move-up policies, which we expect to see in systems with more ambulances. The

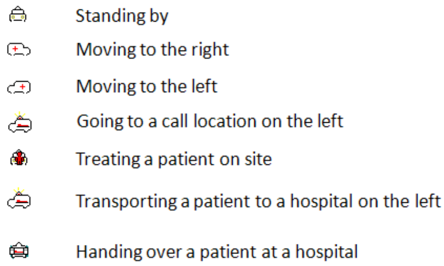


Figure 5.3: Definitions for the icons used in Figures 5.4, 5.5 and 5.6.

methodology of the review is to start with a specified state and show the sequential moves according to a common sample path. Here a sample path refers to a sequence of events in a certain period of time. The sample path we use is defined as follows: at time-step 1, a call occurs at node 12 and there is no call arrival for at least the next 14 minutes; the on-site treatment duration for the call is 1 minute and transportation is required with the at-hospital hand-over duration being 1 minute as well. We choose our starting state to be $((6, \text{Free}), (13, \text{Free}))$ – one ambulance is standing by at node 6 and the other ambulance is at node 13 at time-step 0. To help the reader visualise the sampled moves, plots using icons defined in Figure 5.3 are provided.

For all the optimal move-up policies, we find that there is only one target stand-by configuration for two ambulances, which is node 6 and node 13; in other words, when two ambulances are at nodes 6 and 13, they stay put. This is expected because $r(6, 13)$ is the global maximum of $r(k_1, k_2), \forall k_1, k_2 \in N$.

Consider the scenario shown in Figure 5.4 where $\lambda = 1/15$ calls/min and $h = 4$. The utilisation associated with the move-up policy is about 0.34. At time-step 1, the ambulance at node 13 is dispatched to the call at node 12; under the optimal move-up policy, the free ambulance immediately moves from node 6 to node 10 for stand-by, which is expected as $r(10)$ is the global maximum of $r(k), \forall k \in N$; once the busy ambulance becomes free at time-step 13, it moves from node 4 to node 6 and meanwhile, the other ambulance moves from node 10 to node 13.

Next consider the scenario shown in Figure 5.5 where $\lambda = 1/15$ calls/min and $h = 11$. The utilisation associated with the move-up policy is about 0.32. This scenario differs from

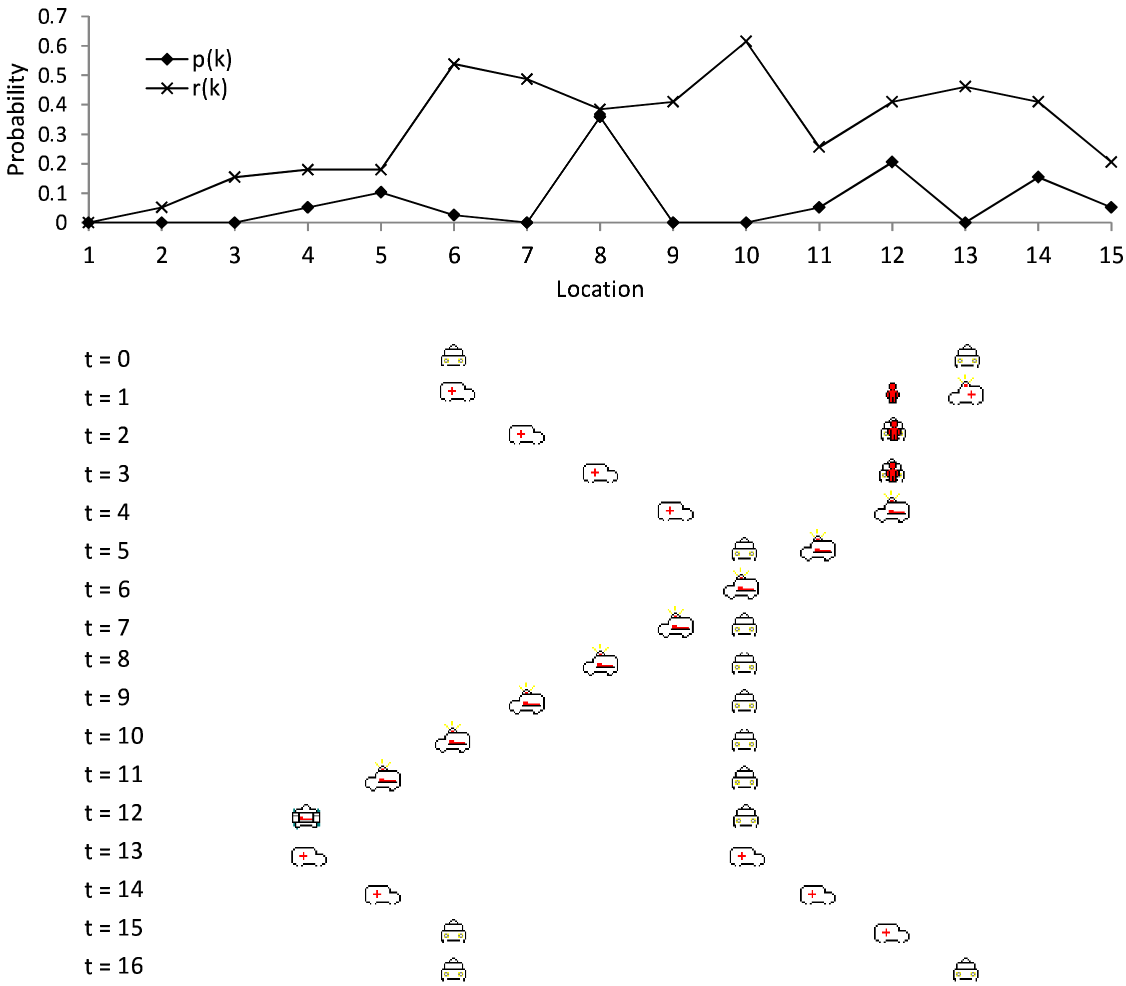


Figure 5.4: An example of ambulance moves under the optimal move-up policy with $\lambda = 1/15$ calls/min and $h = 4$.

the scenario we just reviewed in the hospital location. At time-step 1, the ambulance at node 13, as before, is dispatched to node 12. However, unlike the earlier case, the ambulance at node 6 stays put. The busy ambulance becomes free at time-step 5 and then it moves from node 11 to node 13.

Recall that the ambulance at node 6 would immediately move to node 10 under the scenario where $\lambda = 1/15$ calls/min and $h = 4$ (Figure 5.4). The change in the ambulance's behavior at node 6 is because the busy ambulance is likely to become free at node 11 or node 12 very soon due to the changed hospital location, the short travel times to the hospital, and the short service times on site and at the hospital.

If the free ambulance moves towards 10, *overlapping coverage* around node 10 is likely to form in the near future, which is suboptimal in this case. Here overlapping coverage refers to the case where some nodes are covered by multiple ambulances while some nodes are not covered by any ambulances. Note that overlapping coverage is not always suboptimal; we expect a certain amount of overlapping coverage for large-scale problems. Intuitively, a high-demand location requires coverage by multiple ambulances for the near future to have quick response times. To summarise based on the two scenarios reviewed above, we see that where a hospital is located may play a major role in determining the moves that occur under an optimal move-up policy.

The third scenario we review as shown in Figure 5.6 has $\lambda = 1/4$ calls/min and $h = 4$. The utilisation associated with the move-up policy is about 0.68. At time-step 1, the ambulance at node 13 is dispatched to node 12; the free ambulance at node 6 stays put until the busy ambulance has reached node 8 at time-step 7 while travelling to the hospital at node 4. At this moment, the free ambulance at node 6 starts travelling to node 10 for stand-by. At time-step 12, the busy ambulance becomes free at node 4 and then the two ambulances reach nodes 6 and 13, respectively, at time-step 15 for stand-by.

We see that when the ambulance at node 13 gets dispatched, the ambulance at node 6 waits until the busy ambulance is 'close enough' to the hospital. This 'wait-and-see' behaviour is caused by the high arrival rate and the lower rewards at nodes 7, 8 and 9 in comparison with those at node 6 (the ambulance's current location) and node 10 (giving

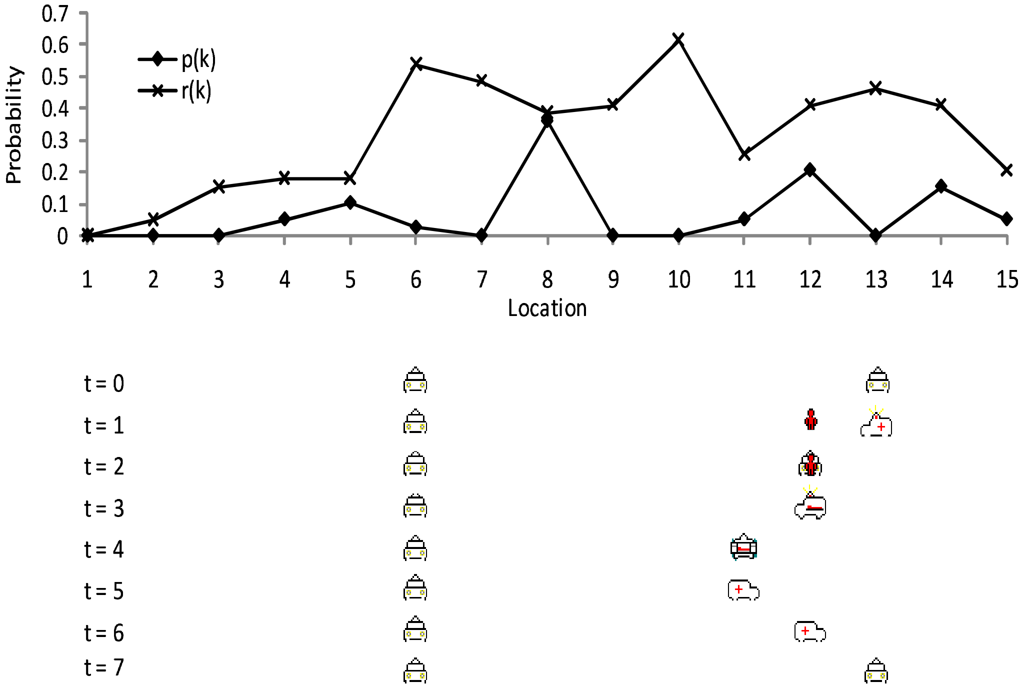


Figure 5.5: An example of ambulance moves under the optimal move-up policy with $\lambda = 1/15$ calls/min and $h = 11$.

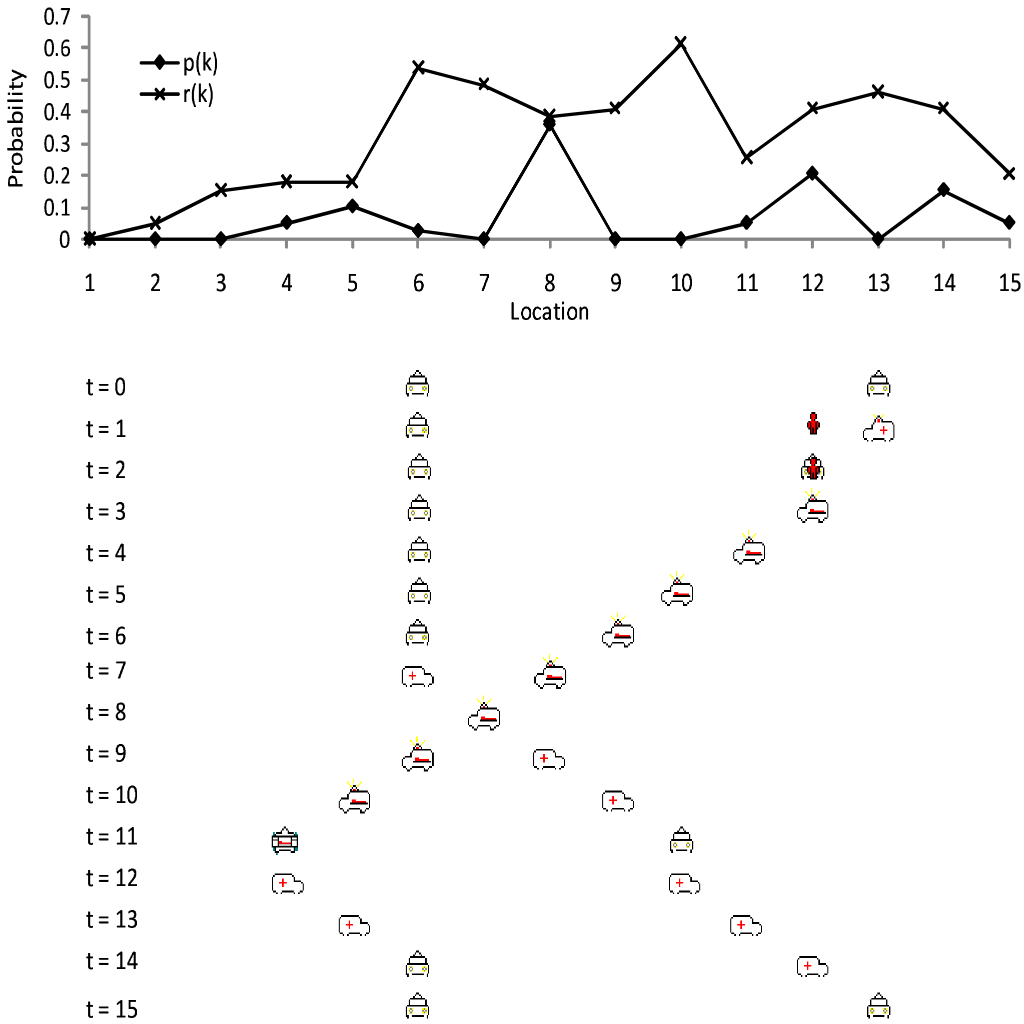


Figure 5.6: An example of ambulance moves under the optimal move-up policy with $\lambda = 1/4$ calls/min and $h = 4$.

the best one-free-ambulance reward). Compared to the first two scenarios we just reviewed, the arrival rate under this scenario is increased about 4 times. The high arrival rate (on average 1 call per 4 minutes) means that an ambulance is likely to get dispatched while travelling to a target stand-by node. So the ambulance could get a worse reward, e.g. $r(8)$, than the guaranteed reward, $r(6)$, of staying put and than the optimistically high reward, $r(10)$, of moving to node 10. This is why the free ambulance at node 6 prefers to wait in the hope that the busy ambulance becomes free on site. If not, it 'reluctantly' moves from node 6 to node 10 when the busy ambulance is close enough to the hospital at node 4; this move-away-from-hospital is again to avoid coverage overlapping around the hospital, as the busy ambulance is expected to become free at node 4 very soon.

The last scenario we review has $\lambda = 1/4$ calls/min and $h = 11$. The utilisation associated with the move-up policy is about 0.65. After the ambulance at node 13 gets dispatched to node 12 at time-step 1, the free ambulance at node 6 stays put regardless of the status of the other ambulance; the busy ambulance becomes free at time-step 5 and reaches node 13 at time-step 7 for stand-by. These moves are the same as those presented in Figure 5.5 for the second scenario we reviewed.

This scenario differs from the previous scenario in the hospital location. Recall that under the previous scenario (Figure 5.6), the ambulance at node 6 moves to 10 once the busy ambulance has reached node 8 while travelling to the hospital at node 4. Because the hospital is now at node 11, the free ambulance at node 6 is even more reluctant to travel through nodes 7, 8 and 9 to node 10, as the busy ambulance is likely to become free around node 10 very soon. These moves reinforce the observation of reluctant moves for the previous scenario and they also suggest that the hospital location may play a major role in determining moves under an optimal move-up policy.

We are aware that the utilisations (0.68 and 0.65) for the last two scenarios reviewed above are probably very high compared to those encountered in reality. However, when the arrival rate is relatively high in realistic-sized problems for which the travel time to a high-reward location is much larger than just a few minutes, we expect similar behaviours, i.e. an ambulance may be reluctant to move from its initial location to a higher-reward location along a path on which the reward at some location is worse than those at the initial location and the final location,

To our knowledge, the moves of the forms described in Figures 5.4 - 5.6 have never been discussed before and they offer new insights into optimal move-up policies. We emphasise that a high-reward location may not be an optimal stand-by location because of a high arrival rate and low rewards along a path on which an ambulance moves towards the location. This feature is not commonly recognised in the literature where the reward collected by move-up is, typically, equal to the reward at the final destination.

5.3.3 Insights into Relative Values

Besides studying the properties of optimal move-up policies, we are also interested in the characteristics of the relative values obtained by solving the optimality equations via value iteration.

Let V, as before, denote the relative value function over S under an optimal move-up policy. In the context of our move-up problem, the difference, $V(s_1) - V(s_2), s_1, s_2 \in S$, represents the extra number of calls that can be reached on time in the long term (assuming $V(s_1) > V(s_2)$) by starting in state s_1 rather than in state s_2.

A principal empirical finding is that the relative value, under an optimal move-up policy, follows an increasing trend if there is no event occurring in the system and at least one ambulance is moving either due to move-up or serving a call. Here an event refers to a call arrival, a completion of on-site or at-hospital service. We now sample some states to demonstrate this insight more carefully.

First, consider the case in which two ambulances are free at nodes k_1 and k_2; in addition, at least one ambulance does not stay put under an optimal move-up policy. Then we have

$$V((k_1, \text{Free}), (k_2, \text{Free})) < V((k_1', \text{Free}), (k_2', \text{Free})),$$

where k_1' and k_2' are the successors of nodes k_1 and k_2, respectively, under the optimal move-up policy.

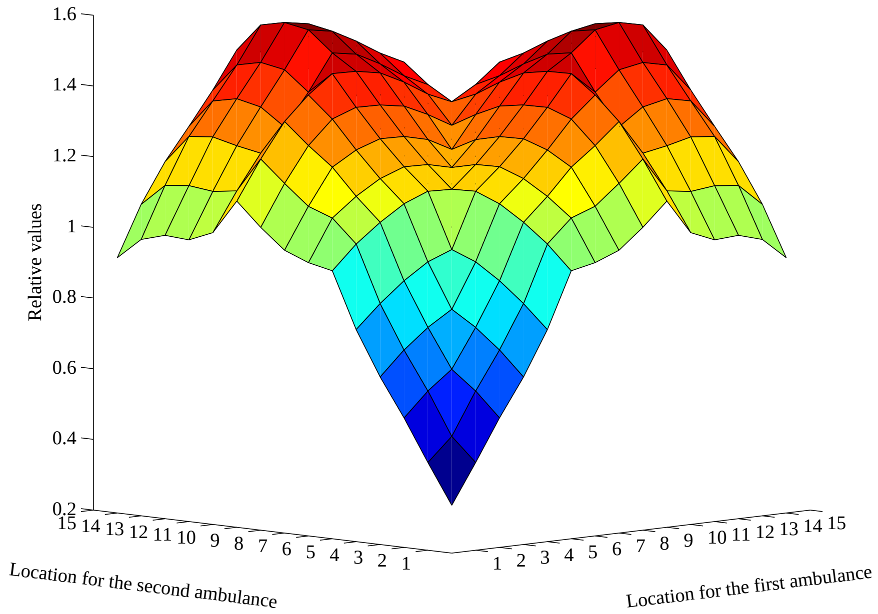


Figure 5.7: The relative values associated with the optimal move-up policy given both ambulances being free when $\lambda = 1/60$ calls/min and $h = 11$.

This observation is expected because the system is still moving towards an 'optimal' configuration (stand-by nodes) in which both ambulances stay put under an optimal move-up policy; state $((k'_1, \text{Free}), (k'_2, \text{Free}))$ is one-step closer to achieve the optimal configuration and so the system is in a better state.

Figure 5.7 shows an example of the relative values given both ambulances are free when $\lambda = 1/60$ calls/min and $h = 11$. We see that the plot is symmetric along the pairs of locations (k,k), k=1,...,15, which is expected by the two ambulances being indistinguishable. We also see that there are two 'peaks' which represent the relative values for state $((6, \text{Free}), (13, \text{Free}))$ and state $((13, \text{Free}), (6, \text{Free}))$; the two peaks are expected, as they correspond to the optimal stand-by configuration given both ambulances being free.

Next, consider the case in which the first ambulance is free at node k and the second ambulance is travelling to (but not yet reach) a call location for on-site service or a hospital for hand over. Then we have

$$V((k, \text{Free}), (i, j)) < V((k', \text{Free}), \text{next}(i, j)), \quad i \neq j,$$

and

$$V((k, \text{Free}), (i, \text{H})) < V((k', \text{Free}), (\text{next}(i, h(i)), \text{H})), \quad i \neq h(i),$$

where k' is the successor of node k under an optimal move-up policy, $\text{next}(i, j)$ is the successor of node i along an shortest path from i to j and $h(i)$ is the hospital that is closest to node i.

Regarding the free ambulance, it is possible that $k' = k$. Nevertheless, as the busy ambulance is one-step closer to the scene or a hospital, it is expected to be one-step closer to become free. Therefore, the system is in a better state after one time-step.

For the states in which both ambulances are busy, we also see similar results. For example, we have

$$V((i, j), (x, y)) < V((\text{next}(i, j), j), (\text{next}(x, y), y)), \ i \neq j \text{ and/or } x \neq y,$$

and

$$V((i, \text{H}), (j, \text{H})) < V((\text{next}(i, h(i)), \text{H}), (\text{next}(j, h(j)), \text{H})), \ i \neq h(i) \text{ and/or } j \neq h(j).$$

Overall, we think the investigation of relative values provides useful insights which can be used to guide the development of approximate models for practical move-up problems.

5.4 Summary

This chapter has been devoted to the study of optimal move-up for two ambulances in order to maximise the system performance over an infinite horizon. A DP model was formulated for this move-up problem. In contrast to the single-ambulance move-up models discussed in the previous two chapters, this model, which involves two ambulances, allows us to gain some understanding of the impact of ambulance interactions on determining ambulance moves under an optimal move-up policy.

A set of simplified small-scale scenarios was used for computational experiments. The optimised move-up policies were shown to provide extra benefits compared to the optimised static policies. Insights into optimal move-up policies and relative value functions were also discussed. Some of the insights presented in Section 5.3 are employed in the integer program presented in Chapter 8. We plan to incorporate more insights gained from this model in future research.

The performance improvement with move-up presented in this chapter is an encouragement to continue our research for the development of large-scale move-up models presented in the subsequent chapters.

6

Simulating Static Ambulance Location Policies for Large-Scale Ambulance Operations

6.1 Overview

The previous two chapters were focused on the formulation of more realistic exact dynamic programming models for move-up than those in the literature. Exact dynamic programming move-up models, which can provide useful insights, quickly become intractable for problems of realistic size. Exploring and developing ambulance move-up models for large-scale ambulance operations is the main focus of the second part of this thesis.

The most realistic approach for evaluating an ambulance location policy (a static policy or a move-up policy) is simulation. In practice, many EMS providers are already using commercial simulation packages for their strategic planning. As briefly mentioned in Chapter 1, The Optima Corporation based in Auckland, New Zealand, specialises in the development of software for ambulance logistics. Their simulation software, called Optima Predict, is now used to improve operational efficiency in ambulance operations in a number of different countries including the UK, Denmark, Australia and New Zealand.

The first version of this software, called BartSim, was developed by Andrew Mason[1] from the Engineering Science Department at the University of Auckland, working in collaboration with Shane Henderson (now at Cornell University) [33]. Optima Predict is a sophisticated

[1] Mason [32] gives a review on the development and applications of this software in a forthcoming book chapter.

package written in C++ for simulating ambulance operations. It provides feedback for complex decision-making, such as recommending the location of a new ambulance base, staff rostering, etc.

A move-up model by Richards [39], which was discussed in Chapter 2, has been embedded into Optima Predict. Optima sees the value of move-up and is always interested in seeking and exploring alternative move-up approaches. They support this research by providing the authors with access to Optima Predict and its source code. Our research has benefited from the existing analytic tools in Optima Predict, and the system has provided a platform for the development of additional features. In fact, all large-scale move-up models presented in this thesis involve simulation-based optimisation: Optima Predict plays a key role in this research.

The purpose of chapter is to familiarise the reader with Optima Predict through the demonstration of three 'optimised' static ambulance location policies under three simplified scenarios. Moreover, these static policies are used to benchmark the performance of the move-up models in the next two chapters. The rest of this chapter is divided as follows. Section 6.2 presents the assumptions and major input parameters used to conduct simulation experiments in Optima Predict. Section 6.3 describes a methodology using a simulation-based local search algorithm to find high-quality static policies. Section 6.4 constructs three simplified scenarios and analyses the results of the corresponding optimised static policies. Similar analysis is performed on the 'optimised' move-up policies for comparison in the following chapters. This chapter ends with a summary in Section 6.5.

6.2 Simulation Environment

In this section, we explain the major assumptions and input parameters used to conduct our simulation experiments in Optima Predict. First, the road network and the allocation of ambulance bases and hospitals on the network are described. Second, key operational rules regarding the dispatch policy and ambulance locating strategy are discussed. We also explain the procedure to create artificial call datasets, which are used for our simulation-based optimisation and evaluation.

Auckland, which is the largest city in New Zealand, is used for simulation experiments. The road network that we use models the actual road network on the avenue level. It is comprised of a set of vertices and arcs. Along each arc we define two 'tiers' of travel speeds that are constant over time. One tier defines the 'normal' travel speed, at which an ambulance is either free and travelling along an arc towards a base or is transporting a patient to a hospital. The second tier defines the 'faster' speed, at which an ambulance is travelling along an arc to a call location with lights and sirens turned on. We assume an ambulance always travels from one location to another along the fastest path.

All four major hospitals in Auckland are included in our experiments. The ambulance base locations approximate real ambulance base locations around Auckland. However, most of the ambulance bases and call demand in rural and remote areas are excluded in the simulation experiments.

Our focus on the performance of EMS operations in urban areas is motivated by challenges faced by the local EMS provider, St John. In Auckland, calls are prioritised into two categories [33]:

> Priority 1: Calls involving life-threatening symptoms for which an ambulance should respond at maximum possible speed with the use of lights and sirens.
>
> Priority 2: Other calls for which an ambulance may respond at standard traffic speeds.

St John published their response targets and actual performance regarding priority 1 calls in 2011 [1] which are summarised in Table 6.1. Call locations are divided into three groups: (1) urban areas, (2) rural areas, and (3) remote areas. For each category, there are two response targets to meet. For example, the second row of Table 6.1 indicates that 50% of priority-1 calls in urban areas are expected to be reached within 8 minutes, while the actual performance for 2011 was 46%. The data suggest that St John does reasonably well in rural and remote areas, but has difficulty meeting the two targets in urban areas. Thus we focus on the performance of their operations around urban areas covering approximately 600 km^2 in total. The geographical allocation of hospitals and ambulance bases is shown in

	Target time	Target success	Actual success
Urban	8 min	50%	46%
Urban	20 min	95%	93%
Rural	12 min	50%	52%
Rural	30 min	95%	92%
Remote	25 min	50%	60%
Remote	60 min	95%	95%

Table 6.1: St John Ambulance's response time targets and actual success for priority-1 calls in urban, rural, and remote areas of Auckland, 2011 [1].

Figure 6.1. The number above each ambulance base represents the base index. In total, 16 ambulance bases are included in our experiments. Note that for the more dynamic move-up approach, an ambulance may also stand by at a street corner, which is controversial in EMS operations: it is possible that ambulance crews may sit idle in the vehicle for a long time, which may cause discomfort and frustration from the crews' frustration . At this stage, we do not consider street corners for stand-by in our experiments.

The dispatch policy implemented here involves dispatching the closest ambulance to a call if there is at least one free ambulance. Otherwise, the call is entered into a queue and served in a first-in-first-out fashion. This closest-ambulance dispatch policy is commonly used in EMS operations. Research that investigates the impact of different dispatch policies is becoming an active area, e.g. see Lim et al. [29]. As in the DP models studied in the previous chapters, we assume that there is no dispatch delay or mobilisation delay, as discussed in Chapter 1. If there is dispatch delay, it is associated with every call and we do not expect that it would alter an optimised ambulance location policy which does not consider dispatch delay. Therefore, we do not include such delay for simplicity. If there is mobilisation delay, an improvement using move-up can be achieved by putting ambulances on the road to eliminate such delay. We do not think that purposely eliminating mobilisation delay is the correct motivation for performing move-up. Therefore, we do not include any mobilisation delay at this stage of the research; if a move-up strategy is beneficial in our experiments, we can reasonably expect it to be more effective than the static strategy.

One of the main objectives of this research is to develop move-up strategies that reduce

Figure 6.1: Locations of ambulance bases and hospitals for computational experiments.

response times for large-scale EMS operations. Major factors that justify move-up are changes in ambulance availability, travel times, arrival rate, and the spatial distribution of call demand. In this research, we investigate and compare the performance of static and move-up strategies applied to simplified systems in which travel times, arrival rate and the spatial distribution of call demand do not change over time.

The ambulance availability is changed solely due to serving calls; the number of ambulances on duty is fixed throughout the entire simulation period. In practice, the number of ambulances on duty typically changes over time according to the changes in call demand, travel times, etc. A scheduling problem naturally arises and research on this problem is also becoming active, see Erdoğan et al. [17].

We assume that all calls are of the same type (priority-1 calls from urban areas) with a response time target of 8 minutes, and that the call arrivals follow a Poisson process with a constant arrival rate λ. In future research, we plan to test varying ambulance location strategies in more complicated systems which consider changing travel times, multiple call priorities, dispatch and mobilisation delays, etc.

Artificial call data are created using code written in R by Bulog and Frankovich. R is a software package used for statistical computing and graphics. The purpose of this module is to generate artificial call data on a realistic network. Currently, data can be generated for Auckland, Edmonton, Toronto and New York. The use of call data for the static model and the move-up models presented later can be summarised by two steps. The first step is to use a common training dataset to seek the best policy and the second step is to use a set of test datasets to estimate the policy's performance. We briefly explain the R module and its assumptions as follows.

The spatial distribution of call demand is based on the population distribution around Auckland obtained from Statistics New Zealand [47]. Auckland is divided into 399 suburbs and the population in each is recorded in the population dataset. As call demand in rural and remote areas is not considered in our experiments, we set the population in these suburbs to zero. The normalised population represents the spatial distribution of call demand on the suburb level. We further assume that in each suburb there are 1000 possible call locations, which are uniformly distributed in the suburb; each call location has an equal probability of being chosen if a call occurs in that suburb.

The treatment time, which is the time that an ambulance spends on site, has an exponential distribution with mean 12 minutes. After treating the patient at the call scene, the ambulance transports the patient to the closest hospital with probability 0.8, which was used in the experiments by Maxwell [34]. Transporting to the closest hospital is not always the case in practice; the choice of hospital depends on several factors such as the location and type of the emergency call, the medical resources at different hospitals, etc. The hand-over time an ambulance spends at a hospital also has a exponential distribution with mean 12 minutes. These service-time related parameters used for our experiments are default inputs for the R module.

Note that ambulance bases, hospitals, and call arrivals are allowed to be located off the road network and we calculate the off-network travel time using the Euclidean distance from the closest node on the network and a specified off-network driving speed. Therefore, each path between two points may have off-network travel at the beginning and/or the end of the path.

In our research, we do not have a 'warm-up'/transient period for simulations, i.e. every call in a call dataset is used for the purpose of policy evaluation. The last day of a call dataset contains no calls, which is a 'wind-down' period to allow ambulances to complete services and return to bases. The simulation is completed at the end of the wind-down period.

One may argue that the transient period should not be included for performance evaluation. However, for our experiments presented later, the optimisation for a static/move-up model uses the same training dataset to evaluate each policy; the starting configuration when using the training dataset to seek an optimised move-up policy and using each test dataset to evaluate the optimised move-up policy is always the same as the configuration defined by the optimised static policy. To summarise, we make fair comparisons when selecting the best static/move-up policy and estimating the performance of the optimised static and move-up policies. As we are more interested in the performance difference between varying policies rather than the performance for each policy, we think the transient period has little impact on making comparisons.

6.3 Optimising a Static Ambulance Location Policy

We remind the reader that a static ambulance location policy includes the assignment of each ambulance to a home base. Whenever an ambulance becomes free at the conclusion of a call, it returns to its home base. As discussed in Chapter 1, finding good static policies has received a great deal of interest. Early models assumed that each ambulance is always free at its home base when a call arrives, which is not the case in reality. More recently, many models aiming to optimise static policies in more realistic settings have been developed.

A majority of recent mathematical models employs the idea of approximate hypercube (AH) when calculating ambulance busy probabilities. Such models are typically nonlinear and heuristics were therefore developed to solve them. The original AH model is an approximation of the 'exact' hypercube model, both of which were proposed by Larson [25, 26] to estimate a set of performance measures under a fixed static policy. Larson's models, which are based on the queueing theory, assumed a system-wide exponentially distributed service time and a single server per base. The first assumption was relaxed by Jarvis [23]. Jarvis's AH model allowed one to consider general service times that depend on both demand locations and server locations. The second assumption was relaxed by Budge et al. [11]. Their AH model allows multiple servers per base. Erdoğan et al. [17] proposed a tabu search which uses the AH model in [11] to optimise static policies.

For the large-scale move-up models presented in the next two-chapters, we use the static strategy to benchmark their performance. Static policies were also used to benchmark the performance of a move-up model proposed by Maxwell [34], which is discussed in the next chapter.

To seek a high-quality static policy (which we refer to as an optimised static policy), we present a hill-climbing (local search) algorithm which uses the neighborhood definition given in [17]. Furthermore, instead of using a mathematical model (AH) to evaluate the performance of a solution, we use simulations in Optima Predict, so our evaluation is more accurate. In the hope of finding a globally optimum solution, we use repeated hill climbing with multiple initial solutions; the details are given shortly.

We emphasise that during this optimisation process, a common training dataset is used to evaluate a solution at each iteration while running the local search algorithm with each initial solution. The percentage of calls reached on time, i.e. within 8 minutes, is used as the performance measure. Next, we briefly outline the main components of this simulation-based local search algorithm before stating the algorithm step-by-step.

6.3.1 Solution and Objective

We first define the notation that we use in order to introduce the model and our solution technique. Let N denote the total number of ambulances on duty. Let B denote the number of bases which are indexed by numbers from 1 to B. At any iteration of the algorithm, a solution is specified by a vector $A = (n_1, ..., n_B)$ where $n_b, b = 1, ..., B$ represents the number of ambulances assigned to base b. The solution must satisfy $\sum_{b=1}^{B} n_b = N$. The objective $f(A)$ to maximise is the number (percentage) of on-time calls measured via simulation on a common training call dataset. A call is reached on time if the response time is within 8 minutes. Note that since a common training call dataset is used, i.e. we evaluate a solution on a deterministic call dataset, the number and percentage of on-time calls are equivalent objectives.

6.3.2 Initial Solution

An initial solution is created using a simple random allocation scheme. More specifically, given there are B bases and N ambulances on duty, we randomly generate N integers between 1 and B. The number of ambulances assigned to base $b, b = 1, ..., B$ is equal to the number of times the value b appears in this series of N integers.

6.3.3 Local Search Operations

There is only one operation, *assign*, for this local search algorithm. A neighbour solution of A is y=$assign(A, i, j)$ – the result of reassigning one ambulance from base i to base j .

6.3.4 Neighborhood Structure

The neighbourhood of a solution is constructed by performing *assign* on pairs of bases. It is obvious that this operation can only be performed on base i and j if there is at least one ambulance at i prior to the reassignment.

6.3.5 Updating Rule

In our algorithm, the current solution is updated as soon as an improved neighbour solution is found. The scanning of the neighbourhood of the new solution starts where the scanning of the previous was interrupted.

6.3.6 Step-by-Step Description of the Algorithm

Algorithm 2 A simulation-based local search algorithm to optimise the ambulance-to-home-base solution

```
Create a training call dataset for use in simulation. Create an initial solution A
as described in Section 6.3.2. Evaluate f(A) via simulation using the training call
dataset and set LocalOptimal = False

While LocalOptimal = False

    LocalOptimal = True
    For i = 1 to B
        For j = 1 to B
            if i ≠ j and n_i ≥ 1 and f(assign(A, i, j)) > f(A) then
                set A = assign(A, i, j)
                set LocalOptimal = False
```

We now present this simulation-based local search method in Algorithm 2. Note that the run times of the algorithm are expected to be reduced by storing evaluated solutions, so that if a solution has been evaluated before, a simulation is no longer required. For our experiments below, storing evaluated solutions is not implemented yet. However, we do plan to add this enhancement to the algorithm, which is clearly desirable for commercial use.

6.4 Computational Experiments

In this section, we report the results for the optimised static policies for varying arrival rates and ambulance fleet sizes. Three scenarios are generated for the experiments. The baseline scenario, namely Scenario 1A, has an arrival rate of 9 calls/hr and 12 ambulances on duty. The arrival rate and the ambulance fleet size for this scenario are calibrated such that the expected percentage of calls reached on time using the optimised static policy will be similar to that in St John's 2011 annual report, i.e. approximately 47% (Table 6.1). Scenarios 1B and 2 are variations of this baseline scenario. Scenario 1B has the same arrival rate as Scenario 1A but has 4 more ambulances on duty, i.e. ambulances are operating in a less busier environment. Scenario 2 has the same number of ambulances on duty as Scenario 1B but its arrival rate is 12 calls/hr such that the ambulance utilisation will be similar to the one in Scenario 1A. For each of the three scenarios, the (λ, N), i.e. the arrival rate and the number of ambulances on duty, is listed in Table 6.2. In the next two chapters, the optimised static policies are used to benchmark the performance of the move-up models for each scenario. Results in this section are revisited for comparative analysis.

Scenario	Arrival rate	Number of ambulances on duty
1A	9 calls/hr	12
1B	9 calls/hr	16
2	12 calls/hr	16

Table 6.2: The arrival rate and the number of ambulances on duty in Scenarios 1A, 1B and 2.

We use the methodology introduced in section 6.3 to obtain the three optimised static policies. For each scenario, the training dataset used for Algorithm 2 contains 49 days of call data plus one call-free day. We generate 15 initial ambulance-to-home-base assignment solutions, as described in Section 6.3.2, and the final best solution obtained by running Algorithm 2 with each of these initial solutions, is chosen to form the optimised static policy. The policy selection process introduces a selection bias. We estimate the true performance of an optimised policy using 40 independent test datasets. All the test datasets, which are generated in the same way as the training datasets, contain 89 days of call data plus one call-free day.

As Scenarios 1A and 1B have the same arrival rate, we use the same training dataset for optimisation and the same 40 test call datasets for statistical analysis. The training dataset in Scenarios 1A and 1B contains 10459 calls. For Scenario 2, the training dataset contains 13930 calls. Note that the training dataset for Scenarios 1A and 1B is not a subset of the training dataset for Scenario 2 which has a higher arrival rate; they are generated independently. Similarly, the test datasets for Scenario 2 are also generated independently with respect to those for the other two scenarios.

Algorithm 2 is implemented in C++ and embedded in Optima Predict. We conduct our experiments on a Windows workstation with a 2.4GHz 32-bit Quad Core Intel CPU and 4 GB of RAM. For each of the three scenarios, the associated 15 initial solutions used in Algorithm 2 all lead to the same locally optimal solution. This result suggests that there are only a few locally optimal solutions and that the locally optimal solution obtained may also correspond to the globally optimal solution for the associated training dataset. Therefore, the static policies are expected to be of high quality.

Table 6.3 reports the CPU time for each iteration of Algorithm 2 and the total CPU time to obtain each optimised static policy. Note that the total CPU time is for 15 runs of Algorithm 2.

	Scenario 1A	Scenario 1B	Scenario 2
CPU time/iteration	34 seconds	34 seconds	49 seconds
Total CPU time	32.3 hours	98.3 hours	102.2 hours

Table 6.3: The CPU time for each iteration of Algorithm 2 and the total CPU time to obtain each optimised static policy using the corresponding training dataset.

Table 6.4 summarises the following performance measures for each static policy: (1)

the 95% confidence interval for the expected percentage of calls reached on time, (2) the 95% confidence interval for the expected average response time, (3) the 95% confidence interval for the expected utilisation and (4) the 95% confidence interval for the expected queueing-up probability (the probability that a call is entered into a queue). Keep in mind that the statistics are estimated using the test datasets associated with each scenario[1].

From Table 6.4, we see that, for the same arrival rate, the percentage of calls reached on time is higher in Scenario 1B than in Scenario 1A. This is expected, as more ambulances are available in the system and thus more calls are expected to be reached on time. As we increase the arrival rate (Scenario 2) for the same number of ambulances on duty, the percentage of calls reached on time drops by about 9.23%. The same reasoning applies to the difference in the values for the average response time, utilisation and the queueing-up probability for the three scenarios.

The optimised ambulance-to-home-base assignment solution for each of the three scenarios is listed in Table 6.5. In Scenario 1A, 5 ambulance bases are not assigned any ambulance, 10 ambulance bases are assigned one ambulance, respectively, and 1 ambulance base is assigned two ambulances. For the same arrival rate, Scenario 1B has 4 more ambulances than Scenario 1A. From Table 6.5, we see that 3 ambulance bases, i.e. bases 5, 9 and 16, not assigned an ambulance in Scenario 1A are now assigned one ambulance each. There are also two ambulance bases, i.e. bases 7 and 12, assigned 2 ambulances each. However, base 10, which is assigned 2 ambulances in Scenario 1A, is assigned one ambulance in Scenario 1B. Comparing the solutions for Scenario 1B and Scenario 2, which have the same number of ambulances on duty but different arrival rates, the number of ambulances assigned to each base is the same except for bases 15 and 16.

We now introduce the reader to both existing and new analysis tools in Optima Predict. These tools are designed to provide useful feedback for an ambulance location strategy. Similar analysis is performed in the next two chapters when we present our move-up models. This provides a consistent basis for comparisons of different ambulance location strategies.

[1] When calcuating a confidence interval for a performance measure introduced in Sections 6.4, 7.6, and 8.5, we follow the procedure outlined in [27], which assumes that each observation for the performance measure based a test dataset is a sample point from a normal distribution.

	Scenario 1A	Scenario 1B	Scenario 2
Expected percentage of calls reached on time	$47.39\% \pm 0.2\%$	$66.34\% \pm 0.1\%$	$56.1\% \pm 0.1\%$
Expected average response time (minutes)	10.3 ± 0.03	7.66 ± 0.02	8.93 ± 0.01
Expected utilisation	$46.2\% \pm 0.3\%$	$31.2\% \pm 0.2\%$	$45.3\% \pm 0.2\%$
Expected queueing-up prob.	0.018 ± 0.001	0	0.006 ± 0.002

Table 6.4: The 95% confidence intervals for the expected percentage of calls reached on time, average response time, utilisation and queueing-up probability under each of the three optimised static policy, estimated using the corresponding test datasets.

Base index	Scenario 1A	Scenario 1B	Scenario 2
1	0	0	0
2	0	0	0
3	1	1	1
4	1	1	1
5	0	1	1
6	1	1	1
7	1	2	2
8	1	1	1
9	0	1	1
10	2	1	1
11	1	1	1
12	1	2	2
13	1	1	1
14	1	1	1
15	1	1	2
16	0	1	0

Table 6.5: The ambulance-to-home-base assignment solution associated with the optimised static policy for each of the three scenarios, i.e. the number of ambulances assigned to each base.

Optima Predict allows a user to visualise an optimised ambulance-to-home-base assignment solution on a computerised map. We use Scenario 1A as an example; Figure 6.2 shows the solution along with the spatial distribution of call demand on the Auckland road network. The bracketed number next to each base index indicates the number of ambulances assigned to that base. A set of 500x500 grid cells is used to show the spatial distribution of call demand. Each cell has dimensions of approximately 400x400 meters. The spatial distribution of call demand over this layer is estimated by evenly distributing the normalised population for each suburb into the set of grid cells whose centroids are contained within the suburb. The low call density around bases 1, 2, 5, 9 and 16 is consistent with the disuse of these bases.

Next we employ Optima Predict to estimate statistics for a set of performance measures in which both operations researchers and EMS providers are interested, which are

- At-base dispatch proportion: the proportion of calls being responded to by an at-base

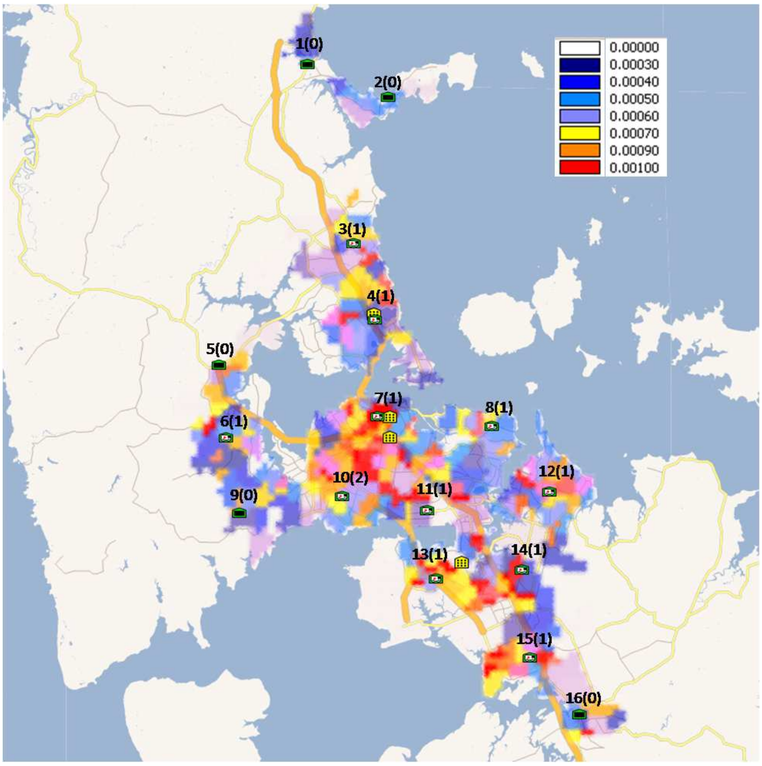


Figure 6.2: Optimised ambulance-to-home-base assignment solution under the static policy and spatial distribution of call demand for Scenario 1A.

ambulance. An at-base ambulance is one that is free and standing by at a base.

- On-road dispatch proportion: the proportion of calls being responded to by an on-road ambulance. An on-road ambulance is one that is free and driving towards a base.
- At-base coverage proportion: for calls served by at-base ambulances, the proportion that is reached on time.
- On-road coverage proportion: for calls served by on-road ambulances, the proportion that is reached on time.
- Average driving distances per vehicle per day: total driving distances divided by the number of days and the number of vehicles on duty.

As before, the test datasets under the corresponding scenario are used to generate the 95% confidence intervals for the performance measures described above, which are reported in Table 6.6. An important observation is that for each of the three scenarios, the number of calls served by on-road ambulances is a significant proportion of the total calls, i.e. the on-road dispatch proportions are about 35.4%, 22.8% and 35.1%. To our knowledge, existing mathematical models aiming to evaluate a fixed static policy or to optimise the static policy ignore those calls served by on-road ambulances. Our empirical results suggest that taking into account those calls that are served by on-road ambulances is important for both evaluation and optimisation of the static strategy, which is also true regarding any move-up strategy.

One means of understanding an ambulance location strategy is to visualise the corresponding geographical coverage. We use Scenario 1A to introduce three new geographical coverage plots that we have added into Optima Predict. Figure 6.3 is a coverage probability plot that illustrates the probability, estimated from the training dataset, of covering a cell by at least one free ambulance. A cell is considered covered if its centroid can be reached on time, i.e. 8 minutes, by at least one ambulance. The call coverage probability for each cell is estimated by sampling coverage at each time instant that a call arrives. Here coverage

	Scenario 1A	Scenario 1B	Scenario 2
Expected at-base dispatch proportion	$64.6\% \pm 0.2\%$	$77.2\% \pm 0.1\%$	$64.9\% \pm 0.2\%$
Expected on-road dispatch proportion	$35.4\% \pm 0.2\%$	$22.8\% \pm 0.1\%$	$35.1\%0.1\%$
Expected at-base coverage proportion	$50.5\% \pm 0.2\%$	$68.5\% \pm 0.2\%$	$59.8\% \pm 0.1\%$
Expected on-road coverage proportion	$41.7\% \pm 0.2\%$	$58.8\% \pm 0.2\%$	$49.2\% \pm 0.1\%$
Expected average driving distances per ambulance per day (km)	433 ± 2	297 ± 1	415 ± 1

Table 6.6: The 95% confidence intervals for the expected at-base dispatch proportion, on-road dispatch proportion, at-base coverage proportion, on-road coverage proportion and average driving distances per day under each of the three optimised static policies, estimated using the corresponding test datasets.

refers to cells that are covered at some time instant. This estimation process is equivalent to a random sampling approach as call arrivals follow a Poisson process for which inter-arrival times are independent. Therefore, a number of independent observations from the same probability distribution are taken for the estimation.

Figure 6.3 contrasts coverage in different areas. Briefly, areas around bases 4, 7, 10 and 13 have good coverage and there is no coverage in areas around bases 1, 2, 5, 9 and 16; the maximum coverage probability is less than 0.8.

The next two plots are a decomposition of the coverage probability plot. The first one is the at-base coverage probability plot and the second one is the on-road marginal-coverage probability plot. The at-base coverage probability plot shown in Figure 6.4 illustrates the probability of covering a cell by at least one at-base ambulance. The on-road marginal-coverage probability plot shown in Figure 6.5 illustrates the probability of covering a cell by at least one on-road ambulance and it is not covered by any at-base ambulances. These two probabilities are also estimated using the same methodology that estimates the coverage probability for each cell. Note that the sum of these two probabilities for a cell is equal to the coverage probability for the cell.

We observe that there is notable on-road marginal-coverage in areas around the three hospitals near bases 7 and 13. This is because about 74% of calls which require transportation to the closest hospital are transported to one of these three hospitals. This means that most ambulances become free at one of these hospitals and then return to their home base. Therefore, on-road marginal-coverage around these three hospitals is a notable feature in the plot.

6.5 Summary

This chapter has been devoted to the introduction of the simulation platform, Optima Predict, used to conduct computational experiments for testing and comparing varying ambulance location strategies in large-scale EMS operations. The major assumptions and input parameters were presented in the experimental setup section. A methodology for

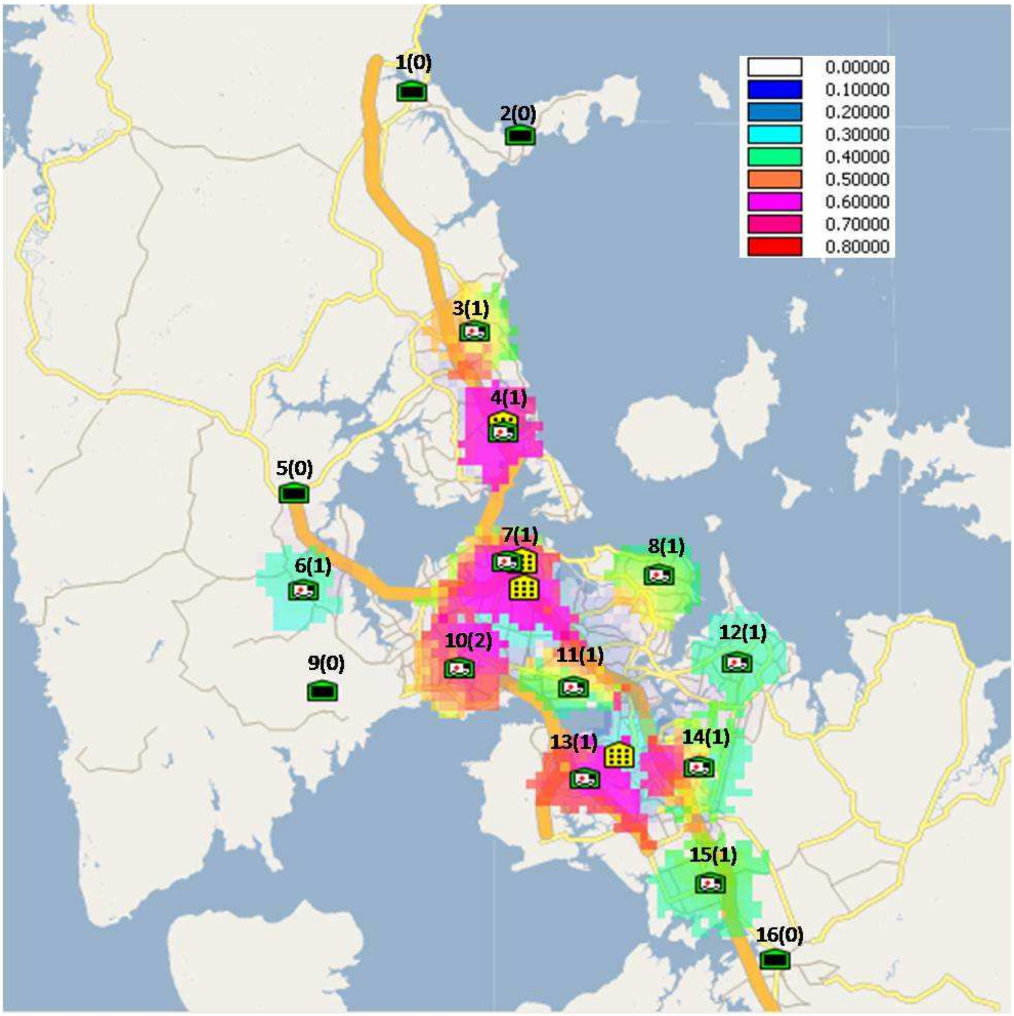


Figure 6.3: Coverage probability plot for the optimised static policy under Scenario 1A, estimated using the training dataset. The number in each bracket indicates the number of ambulances assigned to each base.

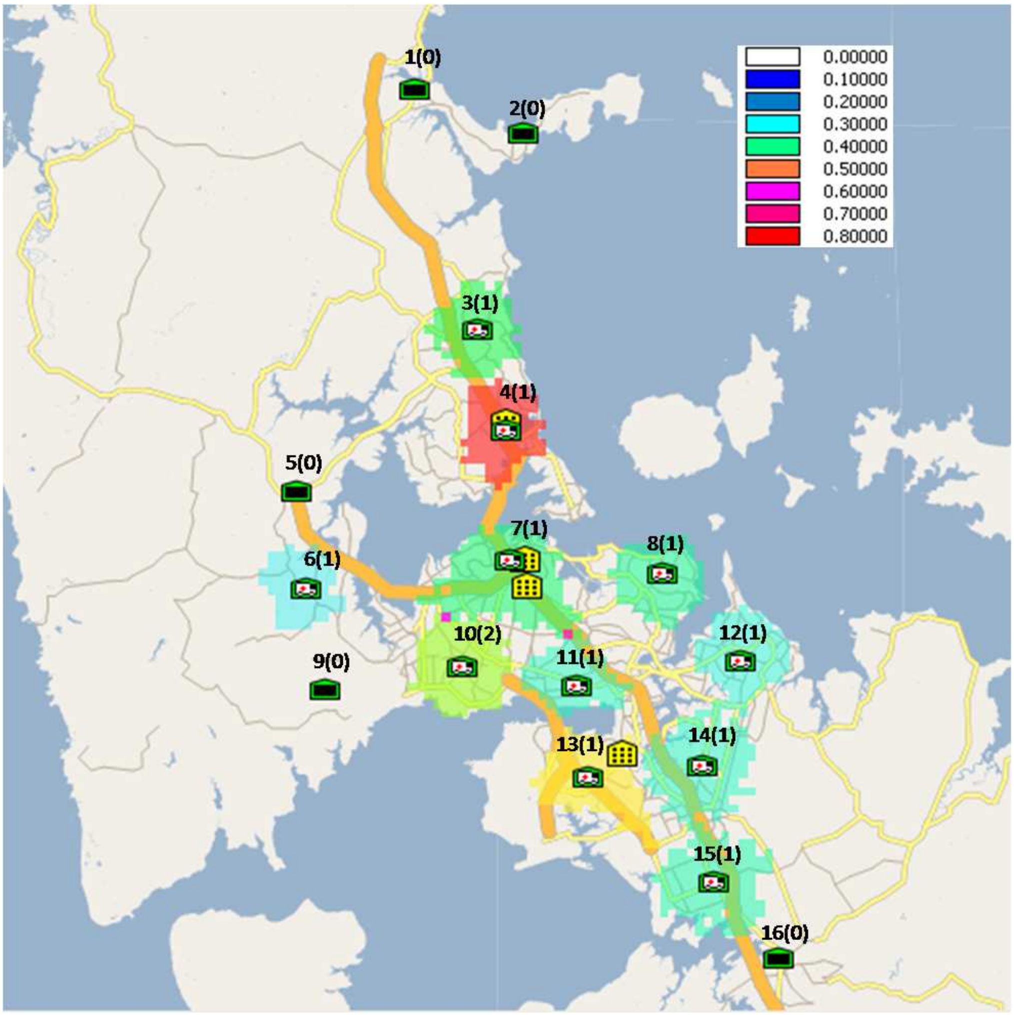


Figure 6.4: At-base coverage probability plot for the optimised static policy under Scenario 1A, estimated using the training dataset.

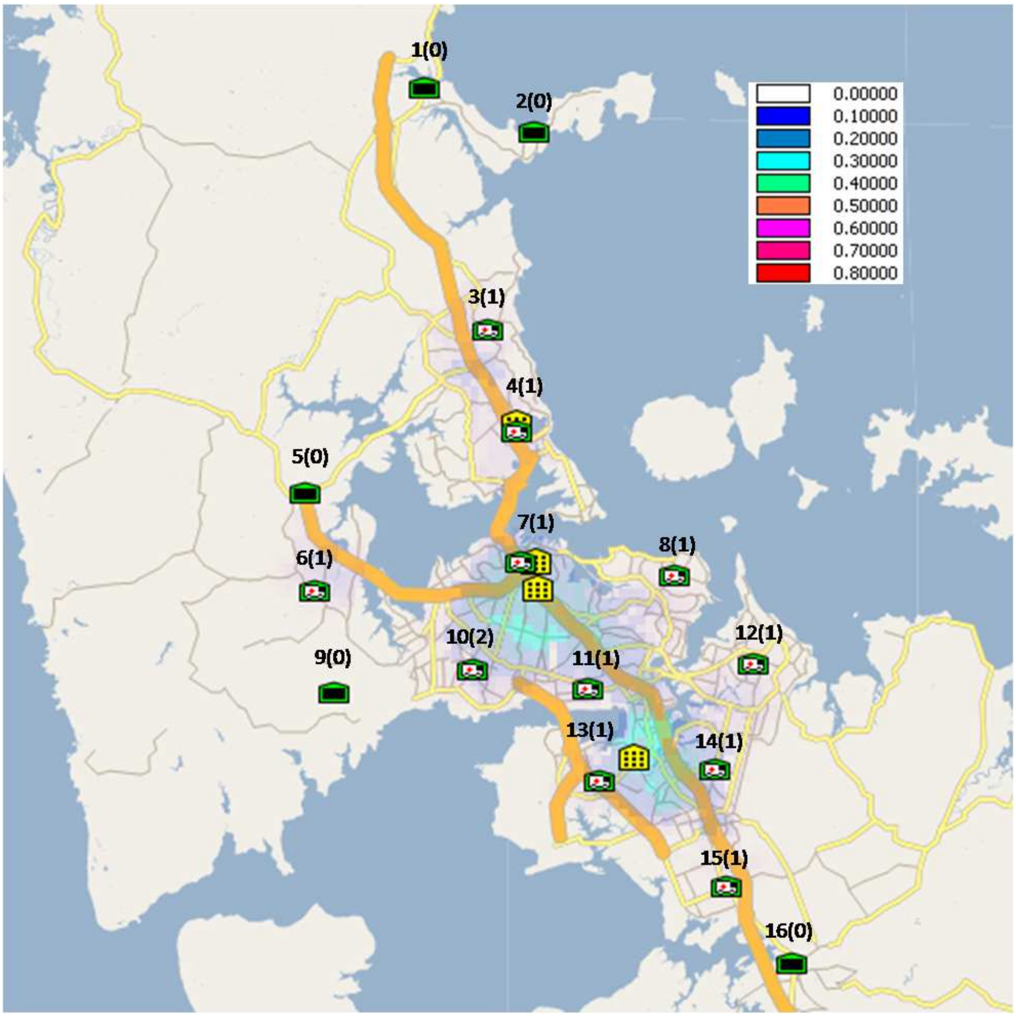


Figure 6.5: On-road marginal-coverage probability plot for the optimised static policy under Scenario 1A, estimated using the training dataset.

optimising static ambulance location policies was also discussed.

Finally, computational results using the optimised static policies for three simplified scenarios with varying arrival rates and ambulance fleet sizes were analysed. The results will be compared to those using three move-up models presented in the next two chapters.

7

Large-Scale Ambulance Move-up Models Using a Priority List

7.1 Overview

In Chapter 6, we introduced the simulation software we will use to conduct our research on large-scale move-up models. We also set up three scenarios and the associated static policies to benchmark the performance of the move-up models we will study. We are interested in seeking move-up models that can reduce response times when compared to the static model. This chapter and the next chapter are devoted to the formulation of three move-up models for practical problems.

In Chapter 2, we surveyed a set of move-up models which have been developed for realistic-sized EMS operations. Three types of move-up policies can be derived from the models in the literature:

- Newly-freed-ambulance move-up policy: whenever there is a newly-freed ambulance, a target base is decided; a move-up event is triggered at the conclusion of each call; typically, the ambulance then follows the fastest path to the base. For models generating such a policy, see Restrepo [30], and Maxwell [34]

- Compliance-table move-up policy: for each number n of free ambulances, there is a pre-defined ambulance configuration $C(n)$ – a set of stand-by locations. Whenever the number of free ambulances changes, the dispatcher chooses a set of moves to reach the target configuration. Typically, the moves are decided by minimising total travel times. For models generating/evaluating such a policy, see Gendreau et al. [21] and

Alanis et al. [2].

Comparing a compliance-table move-up policy and a newly-freed-ambulance move-up policy, the former considers all free ambulances for move-up while the latter only considers one ambulance for move-up. In addition, under a compliance-table move-up policy, a move-up decision is required not just at the conclusion of a call but also at an initiation of a call.

It is also worth mentioning that a compliance table typically has nested configurations in the sense that some or all of the stand-by locations defined in configuration $C(n)$ are also included in configuration $C(n + 1)$. In practice, ambulance crews often find frequent changes of their target stand-by location extremely frustrating. In the case of using a fully-nested compliance table, i.e. all the stand-by locations in $C(n)$ are included in $C(n + 1)$, a target configuration can be achieved by changing the target stand-by location of at most one ambulance when the number of free ambulances changes. This feature of nesting is designed in the hope to reduce ambulance crews' frustration.

- Dynamic move-up policy: at each move-up time instant (which may be at the initiation/conclusion of a call, or determined by some rules), free ambulances are moved to an appropriate configuration according to the location and status of every ambulance using real-time information. Compared to a compliance-table move-up policy, it does not force free ambulances into a pre-determined configuration contained in the compliance table; it is a generalisation of the first two types of move-up approaches. We give more discussions about this type of policy in the next chapter.

In this chapter, we explore the development of newly-freed-ambulance policies and compliance-table move-up policies. The motivation for this chapter originates from the work by Maxwell [34]. As discussed in Chapter 2, Maxwell proposed an ADP model to optimise newly-freed-ambulance move-up policies. An interesting observation made by Maxwell is that, assuming the arrival rate is constant over time (which is one of our assumptions in this research), a newly-freed-ambulance move-up policy based on the ADP model can be fully characterised using a so-called *priority list* derived from weighted basis functions. Informally speaking, a priority list ranks the 'attractiveness' of assigning $n, n = 1, 2, ...$ ambulances to each stand-by location (which is always an ambulance base according to our assumptions). We give its exact definition shortly.

The follow-up observation we can make is that optimising the tunable parameters in the Maxwell ADP model is equivalent to optimising the order of the items in the priority list. A major drawback of this numerical optimisation scheme is that changing tunable parameters may not change the order of the items in the priority list. This means the same priority list can be derived from different sets of tunable parameters. Consequently, identical policies can be evaluated multiple times and unnecessary plateaus may be encountered, which can have a negative impact on the numerical optimisation scheme. Furthermore, in the Maxwell ADP model, the attractiveness of having $n, n = 1, 2, ...$ ambulances stand by at a given base is determined by the same tunable parameter. This means the searching space may not be rich enough in the sense that some priority lists may not be generated based on the Maxwell ADP model. Since the more general optimisation problem is to find an optimised priority list, we consider that optimising numerical values (tunable parameters) associated with the Maxwell ADP model is not addressing the optimisation problem in the most direct way. We therefore propose an alternative method which is a simulation-based local search algorithm that directly sorts the items in the priority list.

The concept of making move-up decisions using a priority list has also motivated us in the development of compliance-table move-up policies. Intuitively, when we have n ambulances available, we attempt to move the n ambulances into a configuration consisting of the stand-by locations having priorities from 1 to n in the list. Consequently, a fully-nested compliance table is formed; details are given shortly. Obviously, one can seek a high-quality priority-order based compliance-table move-up policy by optimising the underlying priority list using our simulation-based local search algorithm presented in this chapter.

As discussed in Chapter 2, Gendreau et al. [21] proposed an integer program aiming to form high-quality compliance tables. The degree of nesting, i.e. the minimum number of stand-by locations in $C(n)$ that must be included in $C(n+1)$, is an input parameter.

The model assumes that given n ambulances available at some move-up time instant, these ambulances are always in the stand-by locations defined by configuration $C(n)$ before the next call-arrival. We think this assumption oversimplifies the complex EMS operations, which is avoided in our simulation-based local search algorithm.

An alternative to using simulations for the evaluation process in our local search algorithm is to use the Markov chain model by Alanis et al. [2] which can reduce run times substantially. The Markov chain model is designed to approximate the performance of a given compliance table. The computational results, involving response time statistics, using their model were shown to be close to using simulations based on data from the city of Edmonton, Canada. However, we prefer to use simulations for three reasons: (1) we have a highly advanced simulation package, Optima Predict, readily available for use which offers more accurate policy evaluations, (2) the analytic tools in Optima Predict allow us to obtain insights into the characteristics of ambulance location policies, and (3) an optimised compliance-table move-up policy is determined before it is used in practice, and so long run times are not as important as the accuracy of the performance estimation.

The remaining sections of this chapter are organised as follows. In Section 7.2, we define the form of a priority list. In Section 7.3, we detail the construction of a newly-freed-ambulance move-up policy using a priority list, which is referred to as a ranked-base free-ambulance move-up policy. In Section 7.4, we show how to construct a compliance table move-up policy based on a priority list, which is referred to as a ranked-base all-ambulance move-up policy. In Section 7.5, we describe a simulation-based local search algorithm to seek a high-quality priority list used to construct a ranked-base free-ambulance move-up policy or ranked-base all-ambulance move-up policy. In Section 7.6, we perform computational experiments to compare the move-up policies optimised using our local search algorithm to the (benchmark) static policies for the three scenarios that we already presented in the previous chapter; in addition, we show that our local search algorithm is expected to yield solutions that are as good as the ones based on the ADP approach by Maxwell [34] with respect to the ranked-base free-ambulance move-up strategy. This chapter ends with a summary and a discussion about motivations for the next chapter.

7.2 Definition of a Priority List

In this section, we define the form of a priority list. Let B denote the number of ambulance bases which are indexed from 1 to B. Let $L = (b_1 \leftarrow m_1, b_2 \leftarrow m_2, ..., b_N \leftarrow m_N)$ denote a priority list consisting of N entries where N is the total number of ambulances on duty and entries are indexed from 1 to N. An entry $b_k \leftarrow m_k, k = 1, ..., N$ has two elements where b_k represents a base index and m_k is an integer equaling the number of times base b_k has appeared in L between the first entry and the kth entry (inclusive). An entry $b_k \leftarrow m_k, k = 1, ..., N$ is considered 'satisfied' by some ambulance configuration if at least m_k ambulances are assigned to base b_k. The priority of an entry refers to the index of the entry in L. The lower the index, the higher the priority. For example, the first entry $b_1 \leftarrow m_1$ has priority 1 and the fifth entry $b_5 \leftarrow m_5$ has priority 5. Then we can say the first entry has a higher priority than the fifth entry.

We would like to note a structural property of a priority list. If base $b \in \{1, ..., B\}$ is contained in a priority list more than once, the priority of entry $b \leftarrow m$ must be higher than the priority of entry $b \leftarrow m'$ where $m' > m$.

To give an example of a priority list, assume $N = 9$, and $B = 3$. A priority list may look like the one shown in Table 7.1.

Priority list
$1 \leftarrow 1$
$2 \leftarrow 1$
$1 \leftarrow 2$
$3 \leftarrow 1$
$2 \leftarrow 2$
$3 \leftarrow 2$
$1 \leftarrow 3$
$3 \leftarrow 3$
$2 \leftarrow 3$

Table 7.1: An example of a priority list given that there are 9 ambulances on duty and 3 ambulance bases

7.3 Ranked-base Free-ambulance Move-up Policy

In this section, we explain how to construct a newly-freed-ambulance move-up policy using a priority list, which we refer to as a ranked-base free-ambulance move-up policy.

Assume there is a newly-freed ambulance, we first calculate the number of free ambulances n_b assigned to each base $b, b = 1, ..., B$ excluding this newly-freed ambulance, which is denoted by a vector $(n_1, n_2, ...n_B)$. A free ambulance is considered assigned to a base if it is driving to the base or already standing by at the base. Given $(n_1, n_2, ...n_B)$, we then find which entries in the priority list are satisfied. The entry $b_k \leftarrow m_k, k = 1, ..., N$ is satisfied if $m_k \leq n_{b_k}$. The base to which the newly-freed ambulance is assigned is the associated base for the 'unsatisfied' entry with the highest priority.

We use the priority list shown in Table 7.1 as an example of how to implement this policy. Assume before the newly-freed ambulance is assigned to a base, we have $n_1 = 1, n_2 = 1$ and $n_3 = 1$ in which case each base is assigned one ambulance. Therefore entries $1 \leftarrow 1$, $2 \leftarrow 1$ and $3 \leftarrow 1$ are satisfied. Then the unsatisfied entry with the highest priority is entry $1 \leftarrow 2$ with priority 3. Therefore, the newly-freed ambulance is assigned to base 1, resulting in this base having two assigned ambulances.

7.4 Ranked-base All-Ambulance Move-up Policy

In this section, we show how to construct a compliance-table move-up policy using a priority list, which we refer to as a ranked-base all-ambulance move-up strategy. Recall that under a compliance-table move-up, a compliance table specifies, for any given number of free ambulances n, a pre-defined ambulance configuration $C(n)$ that gives a base for each of the free ambulances. Whenever the number of free ambulances changes, first the dispatcher decides a set of moves regarding free ambulances in order to reach a target configuration.

The compliance table is constructed by simply selecting the bases associated with entries having priorities from 1 to n in a given priority list to define configuration $C(n), n = 1, ..., N$. The main feature of this table is that configurations are nested in the sense that bases associated with configuration $C(n)$ are also included in configuration $C(n + 1)$, for

$n = 1, ..., N-1$. We solve an assignment problem to decide the set of moves to be made by the free ambulances in order to reach a target configuration, where the cost of assigning an ambulance to a base is the travel times from the ambulance's current location to the base's location along the fastest path.

7.5 Simulation-based Local Search on the Priority List

In the previous two sections, we have discussed how to use a priority list to construct a ranked-base free-ambulance move-up policy or a ranked-base all-ambulance move-up policy. In this section, we propose a simulation-based local search algorithm to sort the priority list for use under the associated move-up strategy. The main components of the algorithm will be explained followed by a step-by-step description of the algorithm.

7.5.1 Solution and Objective

Under the assumption that there are no capacity constraints, each base can accommodate all N ambulances. This means that all entries $b \leftarrow n, b = 1, ...B, n = 1, ...N$, should be considered in the local search algorithm to seek an optimised priority list. However, we impose an artificial capacity constraint for each base to limit run times of our algorithm. The artificial capacity M is set to be the maximum number of ambulances assigned among all bases under an optimised static policy. We expect that the artificial capacity constraints have little impact on the quality of the final solution because we do not think a base would get assigned with a large number of ambulances in an optimised move-up policy for large-scale EMS operations. Therefore, at any iteration of the algorithm, a solution is fully specified by an *extended* priority list L^e. An extended priority list L^e has the same form as a priority list, but with more entries. More specifically, M entries corresponding to $b = 1, ..., B$ are included in L^e.

Regarding the objective, the objective, $f(L^e)$, to maximise is the number (percentage) of calls reached on time measured via simulation on a common training call dataset, which is the same objective as we used in Algorithm 2 in the previous chapter.

7.5.2 Initial solution

We follow the approach taken by Maxwell [34] for creating an initial solution. Maxwell [34] showed a transformation algorithm which uses basis functions weighted by the tunable parameters to construct the corresponding priority list. The initial solution is created by setting the tunable parameters to be 1. The algorithm is presented as follows.

Each ambulance base $b = 1, ..., B$ contributes one basis function ϕ_b. Each basis function ϕ_b is an Erlang loss function specifying the probability of a call-arrival being refused for service; the queuing capacity is n_b where n_b is the number of ambulances assigned to base b[1]. We can write $\phi_b(n_b)$ as

$$\phi_b(n_b) = \frac{(\lambda_b/\mu_b)/n_b!}{\sum_{m=0}^{n_b}(\lambda_b(n_b)/\mu(n_b))^m/m!},$$

where λ_b and μ_b are the arrival rate and service rate of emergency calls for base b, respectively. The arrival rate λ_b is approximated by summing the arrival rates of locations (centroids of cells) for which base b is the closest base (assuming lexicographical ordering on b to break ties). The average service time $1/\mu_b$ is estimated by summing the demand-weighted average response time, at-scene treatment time, transportation time and hospital transfer time for locations served by base b.

Let constant $\phi_b(n_b)^+ = \phi_b(n_b) - \phi_b(n_b - 1)$ and then we rank $\phi_b^+(n_b), b = 1, ..., B, n_b = 1, ..., M$, in non-increasing order (assume lexicographical ordering on b then n_b to break ties) where M is the maximum artificial capacity at each base. The initial solution can be extracted by copying the corresponding base b and n_b from the first BM entries in this ranking in the form of $b \leftarrow n_b$.

7.5.3 Local Search Operations

There are two operations for our local search algorithm. One operation in this algorithm is called *move* which is to move an entry from its current position into a new position in the extended priority list L^e. Thus, we let *move*(L^e, i, j) denote the solution resulting from

[1] Recall that an ambulance is considered assigned to a base b if it is at base b or driving to base b.

moving entry i in L^e into a new position which is just above entry j. The other operation is called *swap* which is to swap the positions of two entries. We let *swap*(L^e, i, j) denote the solution resulting from swapping entry i and j in L^e.

7.5.4 Neighbourhood Structure

The neighbourhood of L^e is constructed by performing the two operations on pairs of entries in L^e. However, not all pairs of entries need to be considered. First of all, if any *move* or *swap* on a pair of entries leads to the priority of entry $b \leftarrow m$ being higher than $b \leftarrow m'$ for some base b and $m > m'$, then we consider such a solution as an infeasible solution, and we exclude it from the neighbourhood. This is because such a solution does not satisfy the structural property of an extended priority list. Secondly, all neighbouring solutions of L^e should differ from L^e and each other in the sense that at least one of the top N entries is different between any pair of these solutions. This is because, as we mentioned earlier, if the top N entries of two extended priority lists are the same, then they lead to the same policy.

7.5.5 Updating Rule

The current solution is updated as soon as a better neighbour solution is found. The scanning of the neighbourhood of the new solution starts where the scanning of the previous one was interrupted.

7.5.6 Stopping Rule

The algorithm stops when no neighbouring solution leads to a better objective value. The optimised priority list for use is extracted from the locally optimal solution by selecting the first N entries.

7.5.7 Step-by-Step Description of the Algorithm

We now present this simulation-based local search scheme in Algorithm 3. The neighbourhood of a solution is scanned by performing *move* on all possible pairs of entries that lead to feasible and different solutions followed by performing *swap* on all possible pairs of entries that lead to feasible and different solutions.

As in Algorithm 2, storing evaluated solutions can be added to Algorithm 3 and the run times are expected to be reduced. The experiments below do not employ this feature yet and we also plan to add it to the algorithm for commercial use.

Algorithm 3 A simulation-based local search algorithm to optimise the priority list

Create a training call dataset for use in simulation. Create an initial solution L^e as described in Section 7.5.2 and let $|L^e|$ denote the number of entires in L^e. Evaluate $f(L^e)$ via simulation and set LocalOptimal = False.

Define a Boolean function F which returns True if a solution created by a *move* or *swap* operation is feasible; Otherwise False.

While LocalOptimal = False

    LocalOptimal = True
    For $i = 1$ to $|L^e|$
        if $i < N$ then set $u = 1, v = N + 1$
        else set $u = 1, v = N$
        For $j = u$ to v
            if $i \neq j$ and $i+1 \neq j$ and $F(move(L^e, i, j))$=True and $f(move(L^e, i, j))> f(L^e)$ then
                set $L^e = move(L^e, i, j)$
                set LocalOptimal = False
    For $i = 2$ to $|L^e|$
        if $i \leq N$ then set $u = 1, v = i - 1$
        else set $u = 1, v = N$
        For $j = u$ to v
            if $F(swap(L^e, i, j))$=True and $f(swap(L^e, i, j))> f(L^e)$ then
                set $L^e = swap(L^e, i, j)$
                set LocalOptimal = False

7.6 Computational Experiments

We perform computational experiments for the three scenarios studied in Section 6.4. The arrival rate and the number of ambulances on duty for each scenario is shown in Table 6.2 on page 81. The experiments are conducted with two aims in mind. Firstly, we compare the performance of the move-up policies obtained from Algorithm 3, (i.e. the optimised ranked-base free-ambulance move-up policies and ranked-base all-ambulance move-up policies) to the (benchmark) static policies from several perspectives. Secondly, we compare the effectiveness of Algorithm 3 and a numerical optimisation method similar to that used in Maxwell's ADP model aiming to optimise ranked-base free-ambulance move-up policies.

7.6.1 Results

For each of three scenarios, we use Algorithm 3, which is coded in C++ and embedded into Optima Predict, to obtain an optimised ranked-base free-ambulance move-up policy and ranked-base all-ambulance move-up policy. We use the Coin-OR-branch-and-cut (Cbc) solver to address the assignment problems arising during the simulation of a ranked-base all-ambulance move-up policy. The training dataset used in Algorithm 3 to obtain the two move-up policies is the same to that used in Algorithm 2 to obtain the benchmark static policy. For simplicity, we refer to the benchmark static policies as static policies.

We conduct our experiments on a Windows workstation with a 2.4GHz 32-bit Intel Quad Core CPU and 4 GB of RAM. The CPU time per iteration (i.e. per function evaluation), the number of evaluations, and the total CPU time to obtain each optimised move-up policy are displayed in Table 7.2. Here a function evaluation means measuring the number of calls reached on time in the corresponding training dataset for a given solution.

As shown in Table 7.2, evaluating a ranked-base all-ambulance move-up policy takes significantly longer than evaluating a ranked-base free-ambulance move-up policy; the extra CPU time is about 2 to 3 minutes. This is mainly because there are twice the number of move-up decisions to make per function evaluation in a ranked-base all-ambulance move-up policy compared to a ranked-base free-ambulance move-up policy. A secondary reason is that making a move-up decision, i.e. solving an assignment problem, in the former takes slightly longer than the simple look-up process needed for the latter. Nevertheless, it takes no more than 20 milliseconds to make a move-up decision in both policies. The total CPU time ranges from about 11 hours to 14 hours to obtain the optimised ranked-base free-ambulance move-up policies, while it ranges from about 59 hours to 108 hours to obtain the optimised ranked-base all-ambulance move-up policies.

The policy selection in Algorithm 3 leads to a selection bias. Therefore, for each scenario, we use 40 independent test datasets to estimate the performance under the two optimised base-ranking move-up policies. These are the same test datasets that were used to evaluate the static policy in Section 6.4.

Figures 7.1 (a) - (c) depict the performance of all three policies for the three scenarios. The horizontal axis in each of these figures indicates the index of each test dataset, whereas the vertical axis gives the percentage of calls reached on time. For each scenario, the test datasets are indexed in an increasing order of the percentage of calls reached on time under the static policy.

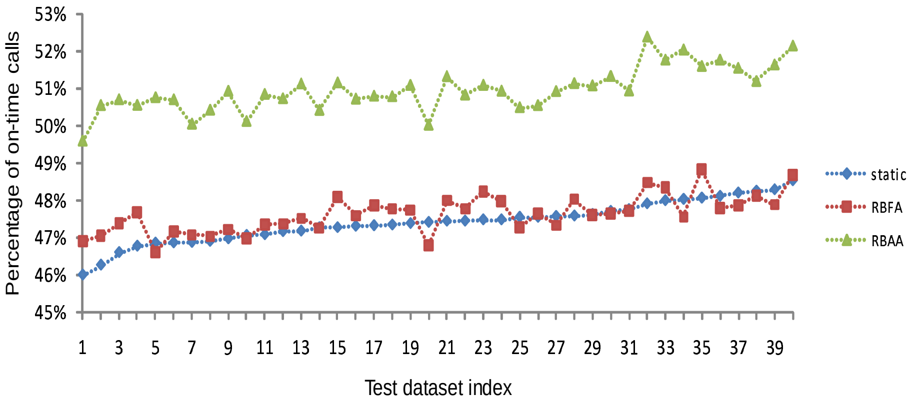
The results for the expected percentages of calls reached on time and the expected average response times, expressed in 95% confidence intervals, are reported in Table 7.3. Keep in mind that they are estimated based on the sample observations from the 40 test datasets under the corresponding sceanrios.

From Figures 7.1 (a) - (c), we observe that for each of the three scenarios, the optimised ranked-base all-ambulance move-up policy outperforms both the static policy and the optimised ranked-base free-ambulance move-up policy for all test datasets. The three optimised ranked-base all-ambulance move-up policies improve the percentage of calls reached on time over the corresponding static policies by about 3.54%, 5.24% and 5.58%, respectively (Table 7.3). For each of the three optimised ranked-base free-ambulance move-up policies, we see that the improvement figure (less than 0.6%) is much smaller; the number of test datasets for which it outperforms the static policy is 26, 36 and 37 out of 40, respectively.

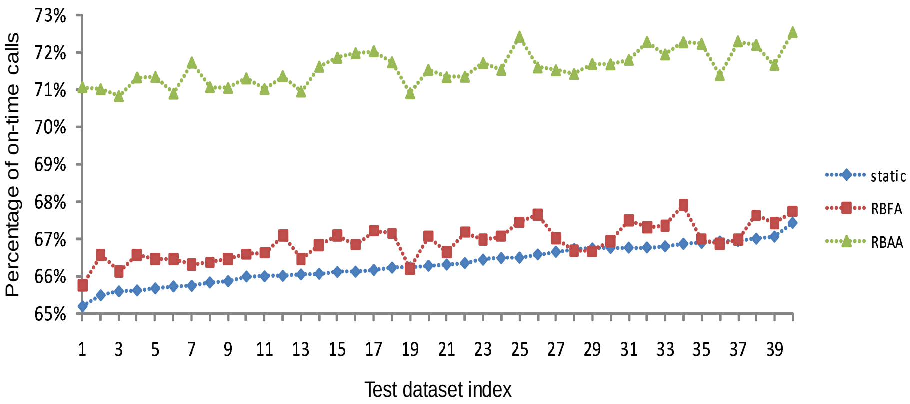
The second response-time performance measure that we use to compare these ambulance location policies is the average response time. From Table 7.3, we see that the optimised

	CPU time/evaluation (seconds)		Total No. of evaluations		Total CPU time (hours)	
Policy	RBFA	RBAA	RBFA	RBAA	RBFA	RBAA
Scenario 1A	38	147	1022	1428	10.79	58.31
Scenario 1B	30	173	2685	2246	22.37	107.93
Scenario 2	44	182	1071	1114	13.09	56.32

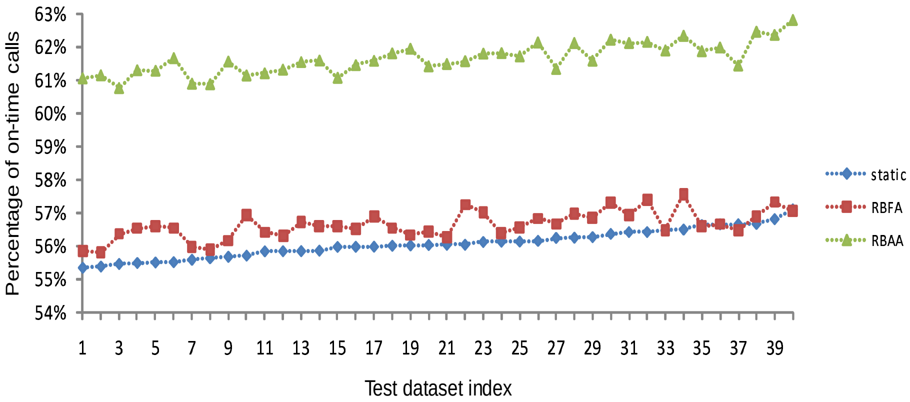
Table 7.2: The CPU time per function evaluation, the number of evaluations and the total CPU time required to obtain the optimised ranked-base free-ambulance move-up policy (RBFA) the ranked-base all-ambulance move-up policy (RBAA) from Algorithm 3 for each of the three scenarios.



(a) Scenario 1A



(b) Scenario 1B



(c) Scenario 2

Figure 7.1: The percentage of calls reached on time using the optimised static policy, ranked-base free-ambulance move-up policy (RBFA), ranked-base all-ambulance move-up policy (RBAA) for each of the test datasets under Scenarios 1A, 1B and 2, respectively.

	Exp. percentage of calls reached on time			Exp. Avg. response time (minutes)		
Policy	Static	RBFA	RBAA	Static	RBFA	RBAA
Scenario 1A	$47.39\% \pm 0.2\%$	$47.64\% \pm 0.2\%$	$50.97\% \pm 0.2\%$	10.3 ± 0.03	10.13 ± 0.03	9.65 ± 0.03
Scenario 1B	$66.34\% \pm 0.1\%$	$66.91\% \pm 0.1\%$	$71.58\%0.1\%$	7.66 ± 0.02	7.67 ± 0.02	7.1 ± 0.02
Scenario 2	$56.1\% \pm 0.1\%$	$56.67\% \pm 0.1\%$	$61.67\% \pm 0.1\%$	8.93 ± 0.02	8.88 ± 0.02	8.22 ± 0.02

Table 7.3: The 95% confidence intervels for the expected percentage of calls reached on time and average response time under the optimised static policy, ranked-base free-ambulance move-up policy (RBFA) and ranked-base all-ambulance move-up policy (RBAA) fore each scenario, estimated using the associated test datasets.

ranked-base all-ambulance move-up policies lead to the smallest average response times for all three scenarios. Using the average response times under the three static policies as the yardstick, the decreases are about 39 seconds, 34 seconds, and 20 seconds; under the optimised ranked-base free-ambulance move-up policies, there is a decrease of about 10 seconds for Scenario 1A and a decrease of about 3 seconds for Scenario 2, while there is no noticeable difference for Scenario 1B. In practice, reduction in response times by even a few seconds can be the difference between life and death for patients with life-threatening conditions. Based on our empirical results, the ranked-base all-ambulance move-up strategy is more effective than the ranked-base free-ambulance move-up strategy in terms of improving the percentage of calls reached on time and reducing the average response time.

Next we use Optima Predict's analytic tools we introduced in Section 6.4 to explore these move-up policies and compare them with the static policies in various aspects.

Tables 7.4 - 7.6 extend Table 6.6, reporting the results for the performance measures using the two move-up policies and the static policy for each of the three scenarios for comparisons. Regarding the driving distances, we change to measure the average extra driving distances per ambulance per day under each move-up policy with respect to the corresponding static policy.

Furthermore, the ranked-base all-ambulance move-up strategy involves moving ambulances out of bases and changing the target bases for on-road ambulances. Therefore, we also report results for four additional performance measures, which are estimated using the sample observations on the 40 test datasets for each scenario, regarding this move-up strategy. The additional performances are:

- Average number of attempted idle-at-base moves per vehicle per day: an attempted idle-at-base move refers to moving an at-base ambulance to another base.

- Average number of redirections per vehicle per day: a redirection refers to the case in which an on-road ambulance that is driving to a target base is redirected to a different target base. Note that the ambulance may not reach the newly assigned target base.

- Average number of back-to-base redirections per vehicle per day: a back-to-base redirection refers to the case in which an on-road ambulance is redirected to the base where it started moving due to an attempted idle-at-base move. Such a redirection is perhaps one of the most frustrating moves from the crews' point of view.

- Average number of relocations per vehicle per day: a relocation refers to the case in which the target base for a free ambulance is changed. Clearly, each sample result based on a test dataset for this performance measure is the sum of the sample results based on the same test dataset for the average number of attempted idle-at-base moves per vehicle per day and the average number of redirections per vehicle per day.

Tables 7.4 - 7.6 show that, for each of the three scenarios, the ranked-base all-ambulance move-up policy and the static policy lead to the lowest and the highest proportions of dispatching at-base ambulances to calls; in other words, the ranked-base all-ambulance move-up policy results in the highest proportions of dispatching on-road ambulances to calls. Specifically, the proportions of dispatching on-road ambulances to calls using the three static policies are about 35.4%, 22.8%, and 35.1%, respectively. For the ranked-base free-ambulance move-up policies, these proportions are about 45.6%, 29.9%, and 46.1%, respectively, and for the ranked-base all-ambulance move-up policies, they are about 50.8%, 31.9% and 46.5%, respectively. These figures reinforce the observation made in the previous chapter: consideration of the on-road-ambulance performance is important in designing ambulance location policies.

Moreover, recall that Alanis et al. [2] proposed a Markov chain model for evaluating a fixed ranked-base all-ambulance (compliance-table) move-up policy. We consider it as an alternative to the simulation evaluation. The model, as discussed in Chapter 2, assumes that there is at most one on-road ambulance whenever the number of free ambulances changes. For each of our three scenarios, we find that under the optimised ranked-base all-ambulance move-up policy, the average percentage of on-road ambulances is about 48.3%, 34.8%, and 46.5%, respectively. We see that there is a signifcant proportion of ambulances being on the road whenever the number of free ambulances changes for our experiments.

Therefore, we think the Markov model may not be adequate as a performance estimation tool given the complexity of our problems.

When a call arrives, it is either responded to by an at-base ambulance or an on-road ambulance. Here an on-road ambulance may just have become free before serving a call in the queue. It is easy to see that the at-base coverage proportion times the at-base dispatch proportion plus the on-road coverage proportion times the on-road dispatch proportion is the overall percentage of calls reached on time. We observe that the ranked-base free-ambulance move-up policies yield a pretty small increase (no more than 2%) in each of these two coverage proportions compared to the static policies. For the ranked-base all-ambulance move-up policies, we see a greater improvement in both of the two proportions; the improvement in at-base coverage proportion is approximately between 5% and 8% on the static policies and the on-road coverage proportion is improved by at least 3%. To summarise, the ranked-base all-ambulance move-up strategy improves both the at-base coverage proportion and on-road coverage proportion more than the ranked-base free-ambulance move-up strategy.

The two move-up strategies result in approximately 80km to 90 extra driving per ambulance per day compared to the static ambulance location strategy for the three scenarios. In addition, the ranked-base all-ambulance move-up strategy also involves ambulances being moved out of bases and redirected on the road, which often lead to crew frustrations in practice. We observe that for each of the three scenarios, there are about 6, 5, and 6 attempted idle-at-base moves per ambulance per day, respectively; there are about 29, 28, and 39 redirections per ambulance per day, respectively; the number of back-to-base redirections per ambulance per day is about 3; in total, the number of relocations is about 35, 33 and 44 per ambulance per day, respectively. We notice that as the number of ambulances on-duty and the arrival rate increase, the number of relocations increases, which is expected, as more move-up decisions involving more ambulances are required.

It is also interesting to see that the number of redirections is much larger than the number of attempted idle-at-base moves. We think this is mainly because many ambulances are on the road when a move-up decision is made, as suggested by the statistics of on-road dispatch proportions.

	Static	RBFA	RBAA
Expected at-base dispatch proportion	$64.6\% \pm 0.2\%$	$54.4\% \pm 0.2\%$	$49.2\% \pm 0.2\%$
Expected on-road dispatch proportion	$35.4\% \pm 0.2\%$	$45.6\% \pm 0.2\%$	$50.8\% \pm 0.2\%$
Expected at-base coverage proportion	$50.5\% \pm 0.2\%$	$52.5\% \pm 0.2\%$	$57.2\% \pm 0.2\%$
Expected on-road coverage proportion	$41.7\% \pm 0.2\%$	$41.8\% \pm 0.2\%$	$45.1\% \pm 0.2\%$
Expected Avg. extra driving distances per vehicle per day (km)	0	85.4 ± 0.3	76.4 ± 0.3
Expected Avg. number of attempted idle-at-base moves per vehicle per day	N/A	N/A	5.7 ± 0.2
Expected Avg. number of redirections per vehicle per vehicle day	N/A	N/A	29.2 ± 0.1
Expected Avg. number of back-to-base redirection per vehicle per day	N/A	N/A	2.9 ± 0.1
Expected Avg. number of relocations per vehicle per day	N/A	N/A	34.9 ± 0.1

Table 7.4: The 95% confidence intervals for the performance measures (defined on pages 87 and 110) under the optimised static policy, ranked-base free-ambulance move-up (RBFA) policy and ranked-base all-ambulance move-up (RBAA) policy, estimated using the test datasets for Scenario 1A with 12 ambulances and 9 calls/hr.

	Static	RBFA	RBAA
Expected at-base dispatch proportion	$77.2\% \pm 0.1\%$	$70.1\% \pm 0.1\%$	$68.1\% \pm 0.2\%$
Expected on-road dispatch proportion	$22.8\% \pm 0.1\%$	$29.9\% \pm 0.1\%$	$31.9\% \pm 0.2\%$
Expected at-base coverage proportion	$68.5\% \pm 0.2\%$	$68.4\% \pm 0.2\%$	$74.3\% \pm 0.1\%$
Expected on-road coverage proportion.	$58.8\% \pm 0.2\%$	$61.2\% \pm 0.2\%$	$65.2\% \pm 0.1\%$
Expected Avg. extra driving distances per vehicle per day (km)	0	86.2 ± 0.2	87.3 ± 0.3
Expected Avg. number of attempted idle-at-base moves per vehicle per day	N/A	N/A	5.0 ± 0.1
Expected Avg. number of redirections per vehicle per day	N/A	N/A	28.1 ± 0.1
Expected Avg. number of back-to-base redirections per vehicle per day	N/A	N/A	2.5 ± 0.1
Expected Avg. number of relocations per vehicle per day	N/A	N/A	33.1 ± 0.1

Table 7.5: The 95% confidence intervals for the performance measures (defined on pages 87 and 110) under the optimised static policy, ranked-base free-ambulance move-up (RBFA) policy and ranked-base all-ambulance move-up (RBAA) policy, estimated using the test datasets for Scenario 1B with 16 ambulances and 9 calls/hr.

	Static	RBFA	RBAA
Expected at-base dispatch proportion	$64.9\% \pm 0.2\%$	$53.9\% \pm 0.1\%$	$53.5\% \pm 0.1\%$
Expected on-road dispatch proportion	$35.1\% \pm 0.2\%$	$46.1\% \pm 0.1\%$	$46.5\% \pm 0.1\%$
Expected at-base coverage proportion	$59.8\% \pm 0.1\%$	$61.6\% \pm 0.1\%$	$67.7\% \pm 0.1\%$
Expected on-road coverage proportion	$59.2\% \pm 0.1\%$	$50.8\% \pm 0.2\%$	$54.6\% \pm 0.2\%$
Expected Avg. extra driving distances per vehicle per day (km)	0	92.9 ± 0.2	70.4 ± 0.3
Expected Avg. number of attempted idle-at-base moves per vehicle per day	N/A	N/A	5.6 ± 0.1
Expected Avg. number of redirections per vehicle per day	N/A	N/A	38.7 ± 0.1
Expected Avg. number of back-to-base redirections per vehicle per day	N/A	N/A	3.0 ± 0.1
Expected Avg. number of relocations per vehicle per day	N/A	N/A	44.3 ± 0.1

Table 7.6: The 95% confidence intervals for the performance measures (defined on pages 87 and 110) under the optimised static policy, ranked-base free-ambulance move-up (RBFA) policy and ranked-base all-ambulance move-up (RBAA) policy, estimated using the test datasets for Scenario 2 with 16 ambulances and 12 calls/hr.

Overall, the four additional performance measures introduced above for a ranked-base all-ambulance move-up policy plus the extra driving distances are viewed as the criteria to quantify the move-up 'costs' in this research. In practice, trade-offs between reduced response times and move-up costs need to be considered carefully. Nair and Miller-Hooks [35] proposed a multi-objective model to quantify the benefit of move-up. The idea is that if one move-up strategy results in the same or a better percentage of calls reached on time while having smaller move-up costs than another, it is certainly a more viable strategy. Following their methodology, the ranked-base free-ambulance move-up strategy leads to a smaller improvement in the percentage of calls reached on time than the ranked-base all-ambulance move-up strategy. Meanwhile, it is also associated with smaller move-up costs in the sense that driving distances for this strategy are similar to the ranked-base all-ambulance move-up strategy, but no costs are incurred in the other three criteria. Therefore, it is hard to judge, between these two move-up strategies, which one is superior to the other.

In Section 6.4, we introduced three geographical coverage plots to help understand ambulance location policies from the coverage perspective. We think this approach is useful to summarise the difference between different policies in an easy-to-see and easy-to-understand format. We use Scenario 1A's training dataset to introduce these plots using the static policies. Here we continue to use Scenario 1A's training dataset and provide a pair-wise comparison of the plots for the ranked-base all-ambulance move-up policy and the static policy. We do not compare all the policies, as our intention is to merely emphasise the usefulness of this approach for providing 'simple' summaries of ambulance location policies to EMS managers.

The coverage probability plot and its two decomposition plots which are paired with the respective plot for the static policy are shown in Figures 7.2, 7.3 and 7.4. All these plots are based on the training dataset. For each plot with respect to the associated static policy, the number in each bracket indicates the number of ambulances assigned to the corresponding base. For each plot with respect to the associated ranked-base all-ambulance move-up policy, the priority of each base is presented, e.g. $P(4) = 3$ means that the priority of base 4 is 3.

Regarding the coverage probability plots as shown in Figure 7.2, we see that in order to improve the percentage of calls reached on time under the ranked-base all-ambulance move-up policy, there is a reduction in coverage around bases 3, 11 and 14 while areas around bases 6, 8, 12, and 15 get improved coverage.

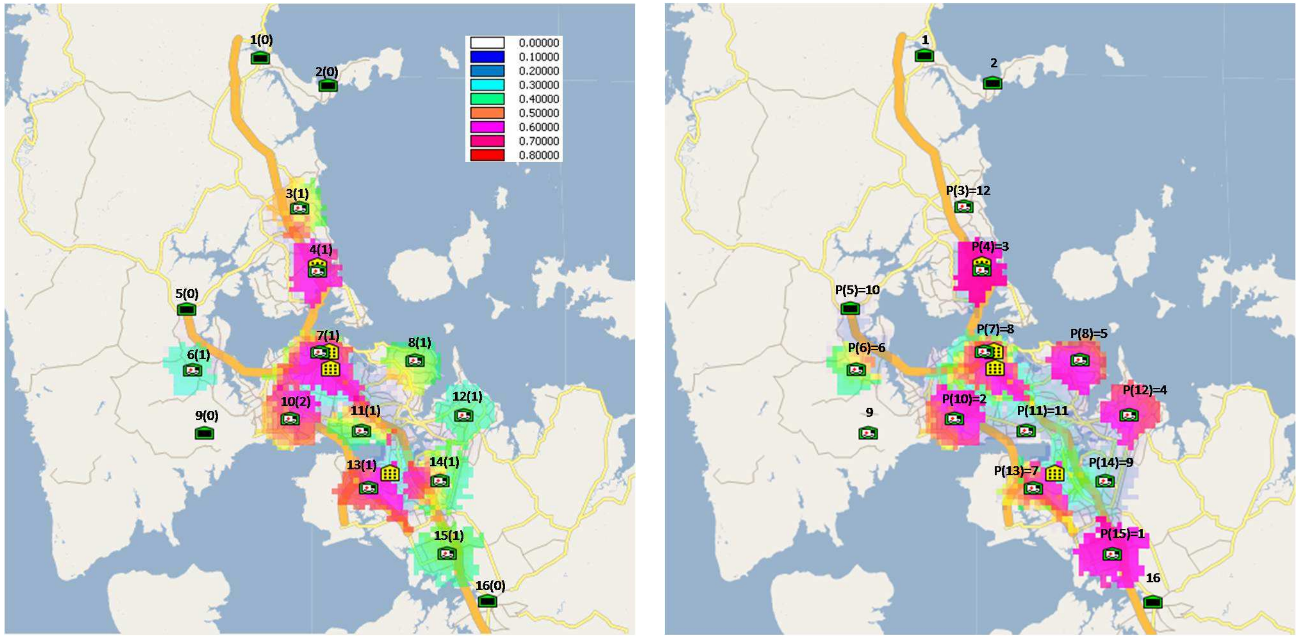
Regarding the at-base coverage probability plots as shown in Figure 7.3, we observe that under the ranked-base all-ambulance move-up policy, areas around bases 4, 8, 12, and 15 get increased at-base coverage, indicating that the probability of having at least one ambulance available at each of these bases when a call arrives is increased due to move-up. We also see that although bases 3, 5, and 11 are included in the underlying priority list for the ranked-base all-ambulance move-up policy, there is no noticeable at-base coverage around these bases. This is because they have low priorities, i.e. $P(3) = 12, P(5) = 10$ and $P(11) = 11$, and rarely get assigned an ambulance due to the relatively heavy workload, i.e. 12 ambulances on duty with the utilisation of approximately 0.46. In other words, the higher the priority is for a base, the higher the at-base coverage probabilities in areas around that base are expected to be.

In the on-road marginal-coverage probability plots as shown in Figure 7.4, we see that under the ranked-base all-ambulance move-up policy, there is a notable increase in on-road marginal-coverage around the central areas of Auckland compared to the static policy, indicating that more ambulances are on the road due to move-up.

7.6.2 Effectiveness Comparison Between Algorithm 3 and Numerical Optimisation

As mentioned earlier, a newly-freed-ambulance move-up policy based on Maxwell's ADP model [34] can be transformed into a ranked-base free-ambulance move-up policy, which motivates us to develop Algorithm 3 to 'directly' optimise this move-up problem, instead of optimising a value function in [34]. Here we compare these two optimisation approaches. For simplicity, we refer to a newly-freed-ambulance move-up policy based on Maxwell's ADP model as a ranked-base free-ambulance move-up policy.

Maxwell [34] investigated two streams of search methods to find an optimised ranked-

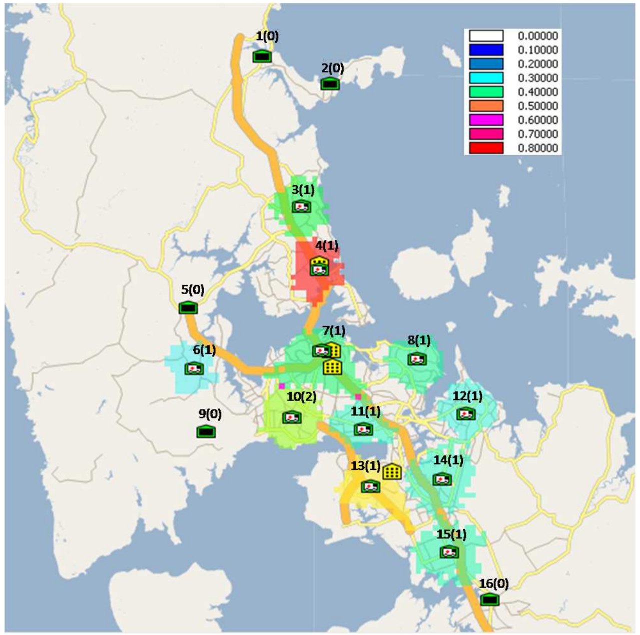


]

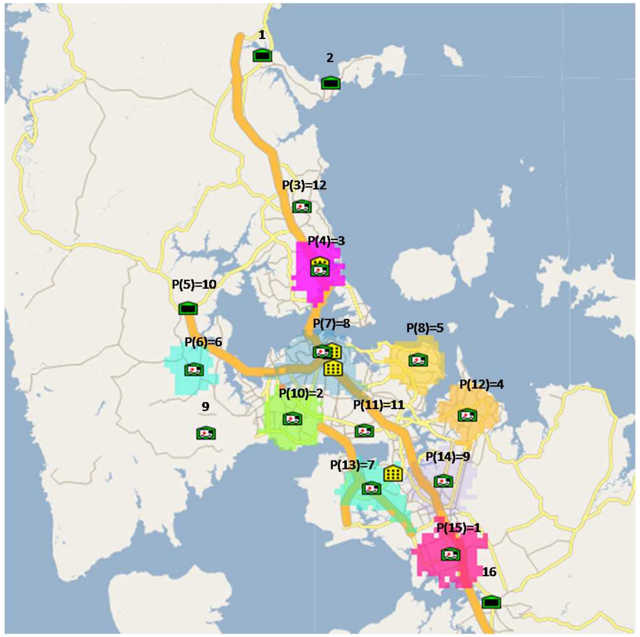
(a) Coverage probability for the static policy. The number in each bracket is the number of ambulances assigned to each base.

(b) Coverage probability for the ranked-base all-ambulance move-up policy. The priority of each base b is indicated by $P(b)$.

Figure 7.2: Coverage probability comparisons between the static policy and the ranked-base all-ambulance move-up policy for Scenario 1A, estimated using the training dataset.

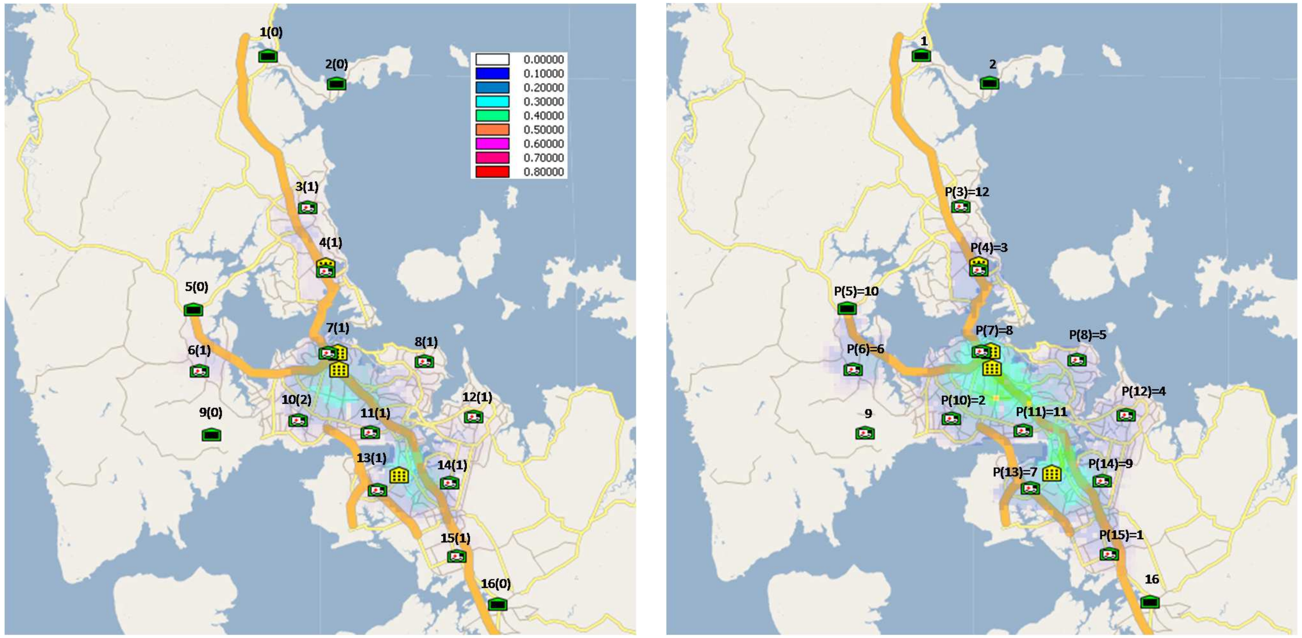


(a) At-base coverage probability plot for the static policy. The number in each bracket is the number of ambulances assigned to each base.



(b) At-base coverage probability plot for the ranked-base all-ambulance move-up policy. The priority of each base b is indicated by $P(b)$.

Figure 7.3: At-base coverage probability comparisons between the static policy and the ranked-base all-ambulance move-up policy for Scenario 1A, estimated using the training dataset.



(a) On-road marginal-coverage probability for the static policy. The number in each bracket is the number of ambulances assigned to each base.

(b) On-road marginal-coverage probability for the ranked-base all-ambulance move-up policy. The priority of each base b is indicated by $P(b)$.

Figure 7.4: On-road marginal-coverage probability comparisons between the static policy and the ranked-base all-ambulance move-up policy for Scenario 1A, estimated using the training dataset.

base free-ambulance move-up policy: (1) the least-squares method, and (2) two direct search methods: Nelder-Mead and unconstrained-optimisation-by-quadratic-approximation (NUOBYQA). The least-squares method is a regression method aiming to get a good fit to the true value function, which is a standard approach for approximate dynamic programs. The approximate value function is then used in lieu of the true value function for the decision-making process.

In contrast to the regression method, direct search methods tune the parameters so as to maximise the performance of the resulting policy rather than to better approximate the true value function. For a good review of direct search methods, see Lewis et al. [28]. Briefly, direct search methods are derivative-free search methods which are guided solely by the function values.

The Nelder-Mead algorithm that we refer to is the *deterministic* version originally developed by Nelder and Mead [36] in the 1960s. It aims to find a local optimum in an iterative fashion. The term deterministic emphasises that the function to be optimised is a deterministic function. Although this 'old' algorithm is a heuristic which does not guarantee convergence, it is one of the most popular direct search methods, which has enjoyed a lot of successes in many fields including statistics, engineering, and the physical and medical sciences [15]. To extend its use in stochastic settings, a *stochastic* Nelder-Mead algorithm has recently been proposed by Chang [13]. Here the term stochastic emphasises that the function to be optimised involves uncertainties. Additional work on the Nelder-Mead algorithm with stochastic functions can also be found in [4].

The NUOBYQA method, proposed by Powell [37], is a relatively new direct search algorithm based on quadratic approximation.

The computational experiments by Maxwell [34] involved two case studies – Edmonton and Melbourne. The deterministic Nelder-Mead algorithm was used to optimise the ranked-base free-ambulance move-up policies for Edmonton and Melbourne. The NUOBYQA method was used only for Edmonton but not Melbourne. This is because NUOBYQA is computationally infeasible given the number of tunable parameters for Melbourne.

For the case of Edmonton, the results for the first 250 function evaluations using the three numerical tuning algorithms were compared. The CPU time per function evaluation for each algorithm was approximately the same. (The CPU time to determine the set of tunable parameters to evaluate next for each algorithm was insignificant.) The results showed that the least-squares method found a reasonably good solution very quickly, but there was no significant improvement with further tuning. Using the performance of the least-squares method as the yardstick, Nelder-Mead found a better solution with a fewer number of evaluations in comparison with NUOBYQA, but was eventually outperformed by NUOBYQA. For the case of Melbourne, Nelder-Mead dominated the least-square method very quickly (after one function evaluation).

Overall, the results suggest that the direct search approach seems to be more effective. Therefore, we are motivated to test the performance of direct search for Maxwell's ADP model under each of the three scenarios introduced in Section 6.4. As our goal is to implement move-up for large-scale problems, we choose to examine the performance of the deterministic Nelder-Mead algorithm in this work rather than the NUOBYQA method which is not suitable for high-dimensional tuning.

Since the deterministic Nelder-Mead algorithm is also employed for the move-up model presented in the next chapter, it is worth summarising the algorithm here. For a careful and modern description of the algorithm, we refer the reader to Lagarias [24].

The algorithm is a *simplex* direct search method to optimise a given function $f : \mathbb{R}^n \to \mathbb{R}$. The term simplex should not be confused with the simplex algorithm for linear programming. A simplex here refers to a polytope of $n + 1$ vertices in n dimensions where n is the number of tunable parameters. For example, a simplex in $\mathbb{R}^2$ is a triangle, and a simplex in $\mathbb{R}^3$ is a tetrahedron.

The algorithm works with a set of $n+1$ points $x_0, ..., x_n \in \mathbb{R}^n$ that are considered as the vertices of a working simplex and the corresponding set of function values $f(x_i), i = 0, ..., n$. To crate a working simplex, vertex x_0 is decided by the user and the other n vertices are created based on some heuristics utilising x_0. Typically, a right-angled simplex at x_0 is created by setting

$$x_j := x_0 + h_j e_j, \quad j = 1, .., n,$$

where h_j is a step-size in the direction of unit vector e_j in $\mathbb{R}^n$.

A sequence of transformations of the working simplex is then performed by 'moving' one or multiple vertices. Each transformation is determined by one of the following operations: (1) reflection, (2) expansion, (3) outside/inside contraction, and (4) shrinking. An operation of (1), (2), or (3) creates a new test point and hence, only one function evaluation is performed; an operation of shrinking is more computationally expensive, as it creates n new test points and n function evaluations are required for a transformation. A number of input parameters are needed to control transformations of the simplex, which are detailed in [24].

A practical implementation of the algorithm needs some criteria to ensure termination in a finite amount of time. Some or all of the following four termination conditions are often used: (1) the working simplex is sufficiently small, i.e. some or all vertices are close enough, (2) the function values of the vertices are close enough, (3) the number of iterations exceeds some pre-determined value, and (4) the number of function evaluations exceeds some pre-determined value. Note that one iteration refers to performing one of the operations mentioned above, i.e. one transformation, which requires one or n function evaluations.

For each of our scenarios, the function to maximise is the number of calls reached on time in the corresponding training dataset. The vertex x_0 is defined by setting the tunable parameters to be 1 in Maxwell's ADP model, which is also the initial solution used in our local search algorithm. A right-angled simplex at x_0 is created as the initial working simplex.

We use termination conditions (2), (3) and (4) for the experiments. For termination condition (2), function values are close enough if the root-mean-square deviation is less than ϵ, i.e.

$$\sqrt{\frac{\sum_{i=0}^{i=n}(f(x_i) - \overline{f})^2)}{n+1}} < \epsilon$$

where $\overline{f} = \frac{\sum_{i=0}^{i=n} f(x_i)}{n+1}$ and ϵ is set to 10^{-4}.

The maximum allowable value for the number of iterations is set to 1000 and that for the number of function evaluations is set to 2500.

Figures 7.5 - 7.7 compare the performance of the Nelder-Mead algorithm and Algorithm 3 for our three scenarios. The x-axis represents the function evaluation index and the y-axis represents the best function value found after the K^{th} function evaluation.

Table 7.7 reports, for each scenario, the performance of the Nelder-Mead algorithm and Algorithm 3 with respect to the total CPU time required to obtain an optimised ranked-base free-ambulance move-up policy, the percentage of calls reached on time in the training dataset under the optimised policy, and the 95% confidence interval for the expected percentage of calls reached on time under the optimised policy, measured using the test datasets. The CPU time per function evaluation for the Nelder-Mead algorithm is not reported, as it is the same as that for Algorithm 3. Note that for the Nelder-Mead algorithm, the lowest indexed set of the tunable parameters that gives the best function value is selected to define an optimised ranked-base free-ambulance move-up policy; there are multiple sets of the tunable parameters leading to the same best function value.

From Figures 7.5 - 7.7, we observe that Algorithm 3 requires more CPU time before termination than the Nelder-Mead algorithm for all three scenarios. However, we think Algorithm 3 is slightly more effective. For each of the three scenarios, Algorithm 3 finds a better solution after no more than 350 evaluations with respect to the Nelder-Mead algorithm. Furthermore, when Nelder-Mead is terminated under Scenarios 1B and 2, Algorithm 3 continues to make improvements. Note that for all three scenarios, the Nelder-Mead search is completed because termination condition (2) is met, i.e. function values are close enough.

From Table 7.7, we see that Algorithm 3 takes as many as 15 extra hours before termination compared to Nelder-Mead. For each of the three scenarios, the best function value found by Algorithm 3 is slightly better than the one found by Nelder-Mead; approximately, an extra 0.6%, 1.6%, and 1.1% of calls can be reached on time, respectively, using the the optimised ranked-base free-ambulance move-up policy obtained from Algorithm 3. In terms of the expected percentage of calls reached on time, there is no significant difference between the policies derived from the two algorithms for Scenario 1A. For Scenario 1B and 2, the policies obtained from Algorithm 3 are slightly better than those obtained from

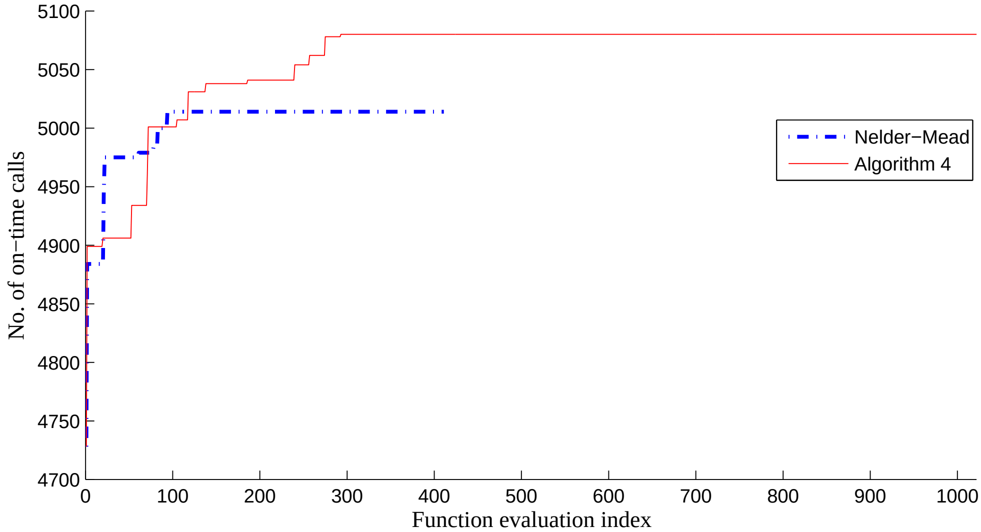


Figure 7.5: Comparisons between Nelder-Mead and Algorithm 3 with respect to the best function value for the training dataset found after the K^{th} function evaluation when seeking an optimised ranked-base free-ambulance move-up policy for Scenario 1A.

Nelder-Mead; the improvement figures are approximately 1.04% and 0.54%, respectively.

The results suggest that Algorithm 3 is more effective than the Nelder-Mead algorithm. The trade-off is more CPU time. Given that the total run times are not a primary concern for this optimisation problem, we therefore recommend Algorithm 3.

7.7 Summary and Remarks

This chapter has been devoted to the analysis of the ranked-base free-ambulance move-up model and ranked-base all-ambulance move-up model. The former decides a target base for a newly-freed ambulance, while the latter allocates $n = 1, 2...$ free ambulances into a pre-determined configuration $C(n)$ whenever the number of free ambulances changes. The fundamental element to form move-up policies based on both models is a priority list which ranks the 'benefit' of assigning $n = 1, 2, ...$ ambulances to each base. Naturally, a simulation-based local search algorithm was proposed aiming to find the 'best' priority list for use in each of these two move-up models.

Computational results for three simplified scenarios were examined from various per-

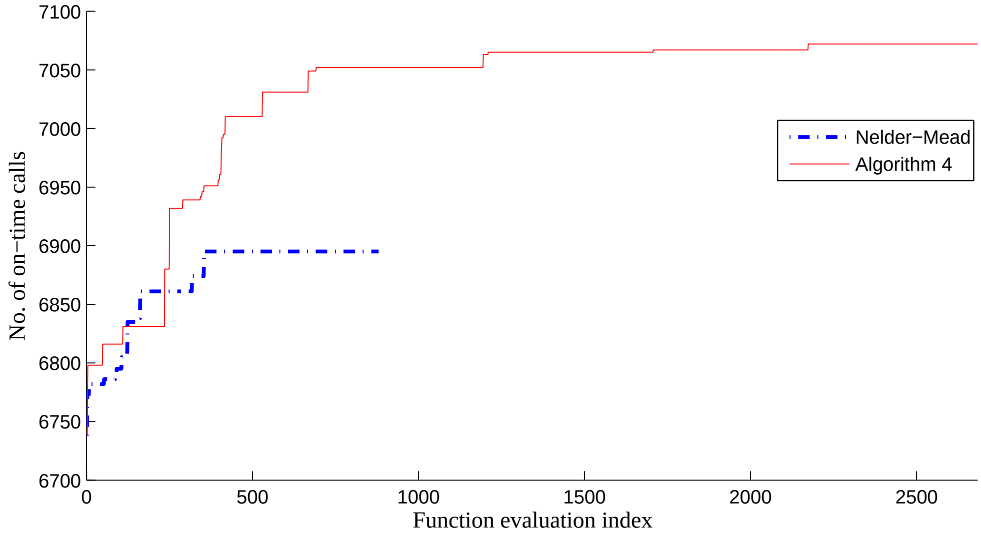


Figure 7.6: Comparisons between Nelder-Mead and Algorithm 3 with respect to the best function value for the training dataset found after the K^{th} function evaluation when seeking an optimised ranked-base free-ambulance move-up policy for Scenario 1B.

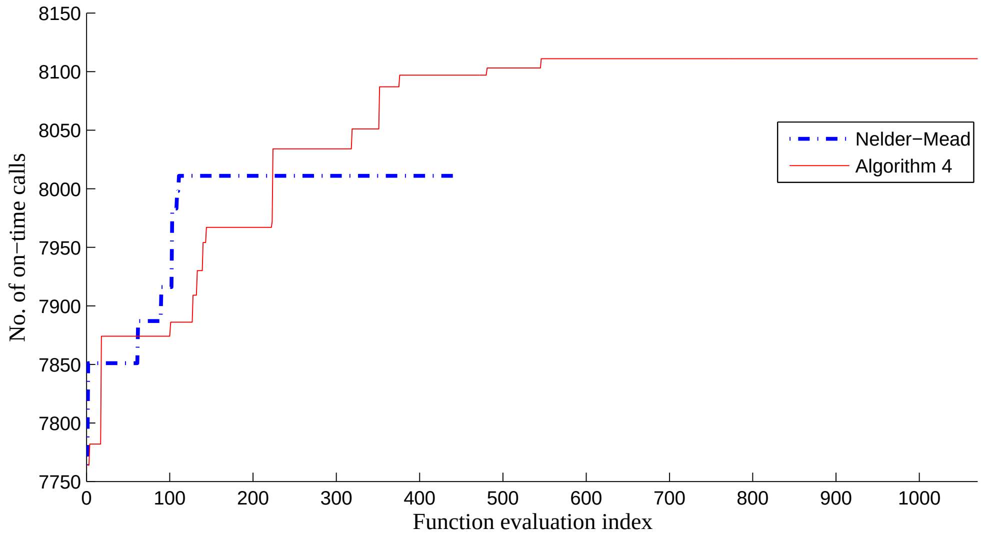


Figure 7.7: Comparisons between Nelder-Mead and Algorithm 3 with respect to the best function value for the training dataset found after the K^{th} function evaluation when seeking an optimised ranked-base free-ambulance move-up policy for Scenario 2.

(a) Scenario 1A

	Nelder-Mead	Algorithm 3
Total CPU time (hours)	4.34	10.79
Percentage of calls reached on time in the training dataset	47.93%	48.57%
Expected percentage of calls reached on time	$47.60\% \pm 0.2\%$	$47.64\% \pm 0.2\%$

(b) Scenario 1B

	Nelder-Mead	Algorithm 3
Total CPU time (hours)	7.33	22.38
Percentage of calls reached on time in the training dataset	65.92%	67.35%
Expected percentage of calls reached on time	$65.86\% \pm 0.2\%$	$66.91\% \pm 0.1\%$

(c) Scenario 2

	Nelder-Mead	Algorithm 3
Total CPU time (hours)	2.73	13.01
Percentage of calls reached on time in the training dataset	57.51%	58%
Expected percentage of calls reached on time	$56.1\% \pm 0.1\%$	$56.64\% \pm 0.1\%$

Table 7.7: The total CPU time to obtain the optimised ranked-base free-ambulance move-up policy from Nelder-Mead, the optimised ranked-base free-ambulance move-up policy from Algorithm 3, the percentage of calls reached on time in the corresponding training dataset under each policy, and the 95% confidence interval for the expected percentage of calls reached on time under each policy, estimated using the corresponding test datasets for each of the scenarios introduced in Section 6.4.

spectives. All the move-up and static policies for these scenarios were optimised in an attempt to maximise the percentage of calls reached on time. Average response times were also used to compare these policies. Our empirical results suggest that the ranked-base all-ambulance move-up policies give the best performance, whereas the ranked-base free-ambulance move-up policies are only marginally better than the static policies.

Although the ranked-base all-ambulance move-up polices were superior to the ranked-base free-ambulance move-up policies and the static policies as far as response time statistics are concerned, they were associated with higher 'costs'. The costs resulted from extra driving distances relative to the static policies, ambulances being moved from one base to another, on-road ambulances being redirected to different target bases, etc. In practice, the benefits of move-up need to be balanced with associated costs. Therefore, it is challenging to judge the viability of different ambulance location policies.

The effectiveness of Algorithm 3 and the (deterministic) Nelder-Mead algorithm applied to Maxwell's ADP model was also compared with respect to optimising the ranked-base free-ambulance move-up policies under our simplified scenarios. Results suggested that Algorithm 3 is more effective than the Nelder-Mead algorithm in terms of the solution quality vs CPU time.

As discussed in Section 7.1, a ranked-base all-ambulance move-up policy specifies a pre-determined configuration $C(n)$ (stand-by locations) for $n = 1, 2, ...$ free ambulances. In other words, such a policy suggests that there is only one 'good' configuration for any given number of free ambulances. However, we believe that for any large-scale EMS operations, there should be multiple 'good' configurations given a number of free ambulances. Recall the simplified examples involving one and two ambulances on small networks in Chapters 3, 4 and 5, where we showed that the optimal stand-by location for each free ambulance may vary depending on many factors such as the arrival rate, the spatial distribution of call demand, the location and the status of a busy ambulance, etc. We expect that a true optimal move-up policy for more ambulances on large networks follows similar behaviours. It is unlikely to find a true optimal move-up policy given the highly stochastic and complex nature of EMS operations. However, our research results on small-scale instances that we discuss next suggest that an optimal move-up policy fits the description of a dynamic move-up policy. Therefore, the next chapter explores dynamic move-up policies based on an integer program.

8

Integer Programming Move-up Model

8.1 Overview

In this chapter, we propose an integer linear programming (IP) model to form dynamic move-up policies. In contrast to a compliance-table move-up policy, a dynamic move-up policy does not force ambulances into a unique configuration given a number of free ambulances. Instead, it decides the appropriate configuration according to the location and status of every ambulance using real-time information.

As discussed in Chapter 2, Gendreau et al. [20] first proposed an IP model to form dynamic move-up policies, which was later revisited by Richards [39]. Andersson [3] also proposed an IP model in the same vein. All these models share a common feature in terms of computing the 'reward' of a given configuration (stand-by locations). The reward of a configuration is computed by summing up rewards collected from demand points. In contrast to these models, we compute the reward of a configuration based on the number of ambulances assigned to each base; this is inspired by the approximation architecture of the ADP move-up model by Maxwell [34], which was also the motivation for the two move-up models presented in the previous chapter.

In Section 5.3, we used simplified examples to show move-up insights involving two ambulances under the DP framework. One main insight is that the (relative) value of moving free ambulances into a target configuration is increasing as the ambulances get closer to the target configuration. We utilise this insight in our IP model by having a cost function based on ambulance travel times; the reward of a target configuration is offset by the cost function. Furthermore, we also observed that busy ambulances which are likely to become free in the near future, such as at-hospital ambulances, may have an impact on where free ambulances should move; this insight is also employed in our IP model, which is discussed shortly.

The challenge is to determine the input parameters, detailed below, associated with the reward function and the cost function so as to obtain a high-quality move-up policy based on the IP model. We address this challenge by using the (determinsitic) Nelder-Mead algorithm discussed in Section 7.6.2: some of the input parameters in our IP model are tuned by Nelder-Mead. By employing Nelder-Mead, our IP model can be viewed as an ADP model which is an extension of Maxwell's ADP model [34]. From the ADP perspective, we are constructing an approximate value function which is tuned via some numerical fitting algorithm. In contrast to Maxwell's ADP model, we do not require the marginal rewards for a base to be determined by a single tunable parameter. Instead, every marginal reward is a tunable parameter. Our approximation architecture is further extended by having a cost for moving an ambulance to a base.

A key input parameter to create the initial working simplex for the Nelder-Mead algorithm is a priority list optimised to form a ranked-based all-ambulance (compliance-table) move-up policy for the EMS system under study. We believe that the priority list contains valuable information that can be used to assess the 'quality' of each base; moreover, it is able to capture the highly complex ambulance interactions implicitly to some extent. Therefore, our IP model can also be viewed as a generalisation of the ranked-base all-ambulance move-up model.

The remaining sections of this chapter are organised as follows. In Section 8.2, we present the model assumptions. In Section 8.3, we show the formulation of the IP model. In Section 8.4, we explain the tuning process. In Section 8.5, we perform computational experiments to compare the performance of move-up policies based on our IP model to the move-up policies and static policies presented in the previous chapters; we also investigate the effectiveness of Nelder-Mead for the tuning process, followed by a discussion about whether move-up is worthwhile or not. We then summarise this chapter in Section 8.6.

8.2 Model Assumptions

In this section, we present the model assumptions. Recall that in Section 5.3, our experiments showed that the locations of busy ambulances may play a key role when deciding the stand-by locations of free ambulances. Undesirable overlapping coverage may be formed if the busy ambulances are not considered. In this model, we assume that at-hospital ambulances are the only busy ones that need to be considered, as they are most likely to become free in the near future. More specifically, our model considers an at-hospital ambulance to be free for move-up, meaning that it can be assigned to a base. This is a heuristic approach we use to anticipate the benefit that at-hospital ambulances can provide.

We assume that for each base, there is an artificial capacity M, which is set to be the maximum number of ambulances assigned to a base under an optimised static policy. This is because we do not think an 'optimised' move-up policy will assign a very large number of ambulances to a base.

Finally, recall that in our small-scale DP models, the move-up path to a location under an optimal move-up policy is specified node-by-node, which may not be the fastest path due to factors such as the arrival rate and spatial distribution of call demand. In this IP model, we assume that an ambulance always travels along the fastest path to a base. As the IP model does not take the rewards (collected from drive-by areas) during move-up into account, this assumption of reaching a target location as fast as possible is appropriate at this stage. We think it is natural to explore the impact of rewards collected on the road on ambulance moves in future research.

8.3 Model Formulation

The objective of the IP model we propose can be viewed as a reward function minus a cost function. As in the ranked-base all-ambulance move-up model, whenever the number of free ambulances changes, a move-up decision that maximises the objective is made. The reward quantifies the benefit of a target configuration while the cost quantifies the 'loss' due to travel required in order to achieve the target configuration. We now define the parameters associated with the reward function and cost function below.

Formally, let B, as before, denote the number of ambulance bases which are indexed by numbers from 1 to B and let $\phi_b^+(m), b = 1, ..., B, m = 1, ..., M$ denote some tunable marginal reward generated by assigning the mth ambulance to base b.

The cost of assigning ambulance v to base b, at some move-up decision time t, is the product of a tunable weight w and a constant parameter c_{vb}. The constant parameter c_{vb} is calculated using the travel time from the current location of ambulance v to base b as follows.

In order to define c_{vb} at a move-up decision time t, we divide the free ambulances (including at-hospital ambulances) into three categories: (1) at-base or newly-freed ambulances, (2) on-road ambulances, and (3) at-hospital ambulances.

For the first category, the parameter c_{vb} is equal to the fastest path's travel time t_{xb}^v from the current location x of ambulance v to base b, i.e. $c_{vb} = t_{xb}^v$.

For the second category, i.e. an ambulance that is on the road and driving towards a base, to define c_{vb}, we introduce a parameter R_{vb} representing a so-called *regret-travel-time*. To define R_{vb}, we need the following parameters:

$$
\begin{aligned}
o^v &= \text{the initial position where the on-road ambulance } v \text{ started moving,}\\
t_{ox}^v &= \text{the fastest path's travel time from position } o^v \text{ to current position } x,\\
t_{ob}^v &= \text{the fastest path's travel time from position } o^v \text{ to base } b \text{ for ambulance } v.
\end{aligned}
$$

The regret-travel-time R_{vb} is defined as

$$R_{vb} = t_{ox}^v + t_{xb}^v - t_{ob}^v.$$

If $R_{vb} \leq E$ where E is a regret-travel-time threshold, then we put $c_{vb} = \alpha t_{xb}^v$ where $0 < \alpha < 1$ is a discount factor. Otherwise we put $c_{vb} = t_{xb}^v$. The reason we discount the parameter t_{xb}^v if $R_{vb} \leq E$ for an on-road ambulance is that we prefer to keep the ambulance driving in the same general direction and yet allow the ambulance to be redirected to an 'underfilled attractive' base that is located in high demand areas.

We view this discounting feature as a soft move-up constraint to prevent on-road ambulances from performing moves that may be considered pointless or frustrating from a crew's perspective, e.g. an on-road ambulance performs a U-turn and drives to a base that is far away from its current location, but very close to where it became free for move-up. Such moves may be able to improve the system performance, but reduce the practicality of the policy. Experiments without this discounting feature, which are not presented in this work, show that the system performance (percentage of calls reached on time) is similar to that with the discounting feature. However, we find that with discounting, move-up costs (defined in Section 7.6) are much smaller, which are detailed in Section 8.5.1.

We are aware that for practical use, soft and hard move-up constraints should be defined after consulting with the EMS provider and ambulance crews. In this research, our focus is mainly on the theoretical improvements based on this IP model.

For the third category, i.e. an at-hospital ambulance v that is regarded as free, we have

$$c_{vb} = t^v_{xb} + t^v_{\mathrm{Free}}$$

where t^v_{Free} is the remaining expected hand-over time duration estimated at move-up time t. For our computational experiments presented in Section 8.5, we assume that the hand-over time at any hospital follows the same exponential distribution with rate μ_{H}. Therefore, we have $c_{vb} = t^v_{xb} + 1/\mu_{\mathrm{H}}$. In more realistic systems, we expect that t^v_{Free} depends on how long an ambulance has been busy at a hospital, e.g. the longer the ambulance has been busy, the smaller the t^v_{Free}. The addition of the remaining expected hand-over time duration helps to counter the assumption that an at-hospital ambulance is free for move-up. Such an ambulance takes longer to travel to a base than a truly free ambulance.

Next we define the decision variables. Let $x_{bm}, b = 1, ..., B, m = 1, , ..., M$ denote a binary variable equaling 1 if at least m ambulances are assigned to base b and 0 otherwise. Let V be the set of ambulances that are actually free at move-up time t. Let Q be the set of at-hospital ambulances at move-up time t. Let $y_{vb}, \forall v \in V \cup Q, b = 1, ..., B$ be a binary variable equaling 1 if ambulance v is assigned to base b and 0 otherwise.

Now the model can be formulated as:

$$\text{maximise} \quad \sum_{b=1}^{B}\sum_{m=1}^{M}\phi_b^{+}(m)x_{bm} - w\sum_{v\in V\cup Q}\sum_{b=1}^{B}c_{vb}y_{vb} \tag{8.1}$$

subject to

$$x_{bm} \leq x_{bm'} \quad \forall b=1,...,B, \forall m=2,...,M, m'=m-1 \tag{8.2}$$

$$\sum_{m=1}^{M_b} x_{bm} = \sum_{v\in V\cup Q} y_{vb} \quad \forall b=1,...,B \tag{8.3}$$

$$\sum_{b=1}^{B} y_{vb} = 1, \quad \forall v \in V \tag{8.4}$$

$$\sum_{b=1}^{B} y_{vb} \leq 1, \quad \forall v \in Q \tag{8.5}$$

$$x_{bm} \in \{0,1\} \quad \forall b=1,...,B, m=1,...M \tag{8.6}$$

$$y_{vb} \in \{0,1\} \quad \forall v \in V\cup Q, b=1...B \tag{8.7}$$

Constraints (8.2) state that we cannot assign the mth ambulance to base b before we assign the $m-1^{th}$ ambulance to base b. Constraints (8.3) count the total number of ambulances assigned to each base. Constraints (8.4) state that each free ambulance must be assigned to one and only one base. Constraints (8.5) state that for an at-hospital ambulance, it does not have to be assigned to a base in which case this ambulance does not contribute any reward or incur any cost. In a solution, an at-hospital ambulance may not be assigned to a base if the associated cost is very large, as might occur if, for example, the remaining expected hand-over time is extremely long so that the ambulance is unlikely to become free in the near future, which means the cost for any assignment regarding this ambulance is very high. Constraints (8.6) and (8.7) are binary constraints.

8.4 Simulation-based Optimisation of Tunable Parameters

In the previous section, we presented an IP model to form dynamic move-up policies. In the objective function (8.1), the marginal rewards $\phi_b^+(m), b = 1, ..., B, m = 1, ..., M$, and the weight w are the parameters to be tuned in order to find a high-quality move-up policy. For this purpose, we employ the deterministic Nelder-Mead algorithm discussed in Section 7.6.2.

As before, we employ simulations for function evaluations in the Nelder-Mead algorithm; each function evaluation refers to measuring the percentage of calls reached on time in a common training dataset by running a simulation which solves the IP model whenever needed to make move-up decisions..

Recall that to create a initial working simplex for the Nelder-Mead algorithm, one vertex needs to be initialised by the user. We explain the initialisation process for the vertex in the next two subsections; this vertex corresponds to the un-tuned (initialised) marginal rewards and the weight. The move-up policy derived from the IP model using the un-tuned parameters is referred to as the *un-tuned* IP move-up policy. Once Nelder-Mead is terminated, the set of the tunable parameters giving the best performance defines the optimised IP move-up policy; if there are ties, the first evaluated set is chosen.

We do not claim that this algorithm is the best tuning method for our model and we do plan to test other search methods in future research.

8.4.1 Initialisation of Marginal Rewards

The initialisation process for the marginal rewards is a heuristic that utilises the optimised priority list generated for the ranked-base all-ambulance move-up policy, as we think this list provides valuable information to help identify attractive bases and it implicitly captures ambulance interactions to some extent. We summarise the process in the following steps:

1 Use Algorithm 3 to find the locally optimal priority list, i.e. the optimised ranked-base all-ambulance move-up policy.

2 Let $L^* = (b_1 \leftarrow m_1, ..., b_N \leftarrow m_N)$ be the optimised priority list. Set $\phi_b^+(m) = 0, \forall b =$

$1, ..., B, m = 1, ..., M$.

3 Simulate the optimised ranked-base all-ambulance move-up policy using the same training call dataset that is used in Step 1.

3.1 Let $c(b_k \leftarrow m_k), k = 1, ..., N$ be the number of calls that are reached on time and served by entry $b_k \leftarrow m_k$ in L^*. A call is considered served by entry $b_k \leftarrow m_k$ in L^* if this call is served by the ambulance that was going to/standing by at base b_k, and there were m_k ambulances assigned to base b_k just before the dispatch.

3.2 Set $\phi^+_{b_k}(m_k) = c(b_k \leftarrow m_k)/D, \forall k = 1, ..., N$ where D is the number of hours with call arrivals in the training call dataset. Keep in mind that for our experiments, the last day of the training call dataset has no call arrivals, which is a 'wind-down' period for ambulances to finish services and return to base.

Under this initialisation scheme, the values of $\phi^+_b(m), b = 1, ..., B, m = 1, ..., M$ can be interpreted as the additional number of calls that can be reached on time per hour after assigning an additional ambulance to each base.

8.4.2 Initialisation of Weight Applied to Travel Times

After initialising the marginal rewards as above, we perform a simulation-based discrete linear search to initialise the weight w. We assume that the initial value for w is positive and we start the search from ϵ with a step size of β. At iteration $K, K = 0, 1, ...,$ we set the value of w to be $\epsilon + \beta K$; we evaluate the move-up policy based on the IP model using the training dataset that is used to initialise the marginal rewards. The discrete linear search is terminated after θ iterations. The value that leads to the maximum number of calls reached on time is set to the initial value for w. In our experiments below, we put $\epsilon = 10^{-5}, \beta = 0.2$, and $\theta = 30$.

8.5 Computational Experiments

We perform computational experiments for the IP model under the three scenarios studied in the previous two chapters. The arrival rate and the number of ambulances on duty for each scenario are listed in Table 6.2 on page 81.

In Section 8.5.1, the results using the optimised dynamic move-up policies based on the IP model are compared to those using the static policies and the move-up policies presented in the previous two chapters. For simplicity, we refer to a move-up policy based on the IP model as an IP move-up policy for analysis below. In Section 8.5.2, we investigate the effectiveness of applying the Nelder-Mead algorithm in the tuning process. In Section 8.5.3, we discuss whether move-up is worthwhile or not compared to the static model.

8.5.1 Results

The IP model has been coded in C++ and embedded into Optima Predict. The experiments are conducted on a Windows workstation with a 2.4GHz 32-bit Quad Core Intel CPU and 4GB of RAM.

For all three scenarios, the discount factor α and the regret-travel-time threshold E associated with the cost function in the IP model are set to 0.5 and 2 minutes, respectively. The combination of this relatively heavy discount factor and a tight regret-travel-time threshold is chosen in an attempt to keep each on-road ambulance driving in the same general direction as before when a new move-up decision is made – a feature that we think may reduce the frustrations from an ambulance crew's perspective.

For each scenario, we implement the Nelder-Mead algorithm to tune the marginal rewards and the weight in the IP model. The one vertex needed to create the initial working simplex involved in Nelder-Mead is constructed as explained in Section 8.4.1 and Section 8.4.2. The input parameters regarding the transformations of the working simplex and termination conditions are the same as those used in Section 7.6.2. The training dataset used for the initialisation process and Nelder-Mead is the same as the one used in Algorithms 2 and 3 to obtain the optimised static policy, ranked-base free-ambulance move-up

	CPU time/evaluation (seconds)	Total CPU time (hours)
Scenario 1A	196	136.1
Scenario 1B	234	162.5
Scenario 2	284	197.2

Table 8.1: The CPU time per function evaluation and the total CPU time for Nelder-Mead to terminate under each scenario.

policy and ranked-base all-ambulance move-up policy. During each function evaluation in Optima Predict, the CBC solver is used to solve the IP model whenever the number of free ambulances changes. Once Nelder-Mead is terminated, the solution (for the tunable parameters) that gives the best function value is used to define the optimised IP move-up policy; if there are ties, the lowest indexed one is selected. As in Algorithms 2 and 3, the selection of the optimised IP move-up policy based on the training dataset introduces a bias. Therefore, independent test datasets are used to estimate the true performance of the policy. These datasets are the same as those used to evaluate the static policy and the other two move-up policies.

For each scenario, the CPU time per function evaluation, and the total CPU time until the Nelder-Mead algorithm is terminated are displayed in Table 8.1. We find that for all three scenarios, Nelder-Mead is terminated due to the number of function evaluations reaching the limit of 2500.

We see that the CPU time per evaluation ranges from 3 minutes to 5 minutes, approximately, and the Nelder-Mead algorithm takes about 6 to 9 days to terminate. The maximum CPU time to make a move-up decision based on the IP model is no more than 0.3 seconds for each of the three scenarios.

Figures 8.1 - 8.3 depict the tuning results for Nelder-Mead using the three training datasets. Each index shown on the x-axis corresponds to a simulation run made using a set of the tunable parameters for the IP model; the y-axis represents the number of calls reached on time in the corresponding training dataset.

Table 8.2 summarises, for each of the three training datasets, the percentage of calls reached on time arising from the un-tuned IP move-up policy (i.e. the policy based on the initial solution for the tunable parameters in the IP model) and the optimised IP move-up

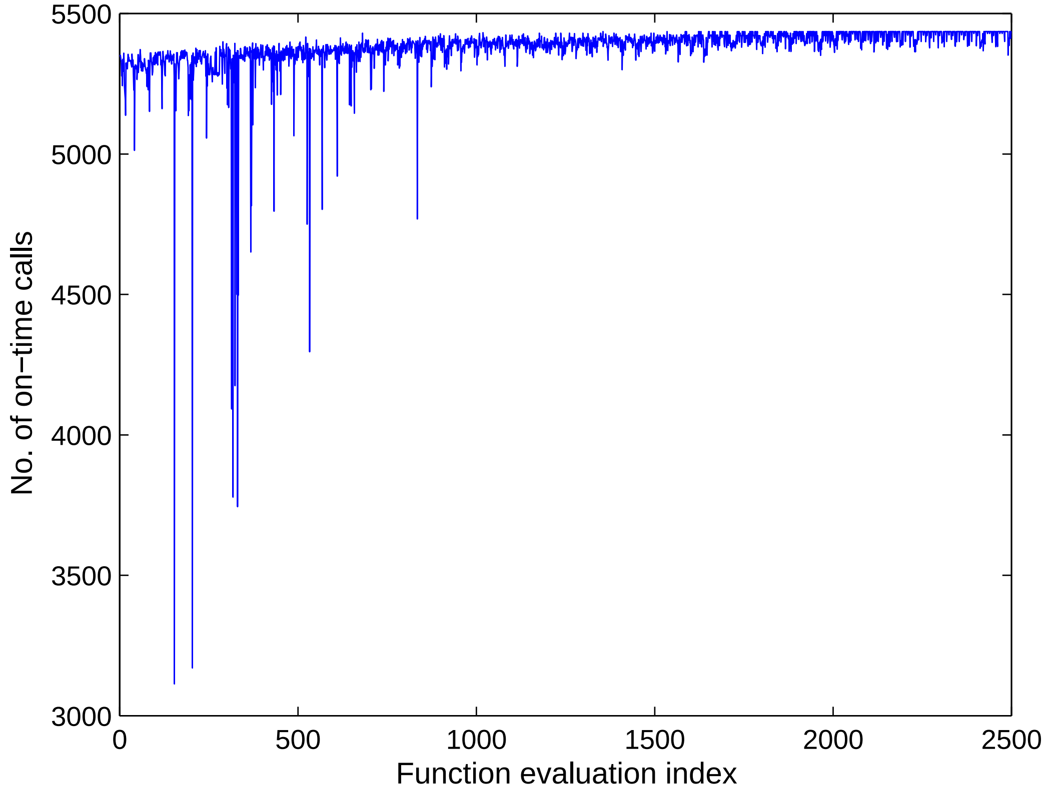


Figure 8.1: Tuning results by Nelder-Mead using the training dataset for Scenario 1A.

policy. We see that the solution qualities are improved by about 0.85%, 1.09% and 0.46%, respectively, for the three training datasets. Given the small size of the improvement figures and the large CPU time required, one may question the effectiveness of Nelder-Mead in terms of improving the solution quality for our model. We return to this question shortly. For now, we continue our analysis.

Figure 8.4 extends Figure 7.1 by showing, for each of the three scenarios, the percentage of calls reached on time under the optimised IP move-up policy in each test dataset. Tables 8.3 and 8.4 extend Table 7.3 by attaching, for each of the three scenarios, the 95% confidence intervals for the expected percentage of calls reached on time and the expected average response time under the optimised IP move-up policy, estimated using the test datasets.

In Figure 8.4, we observe that for each of the three scenarios, the performance of the

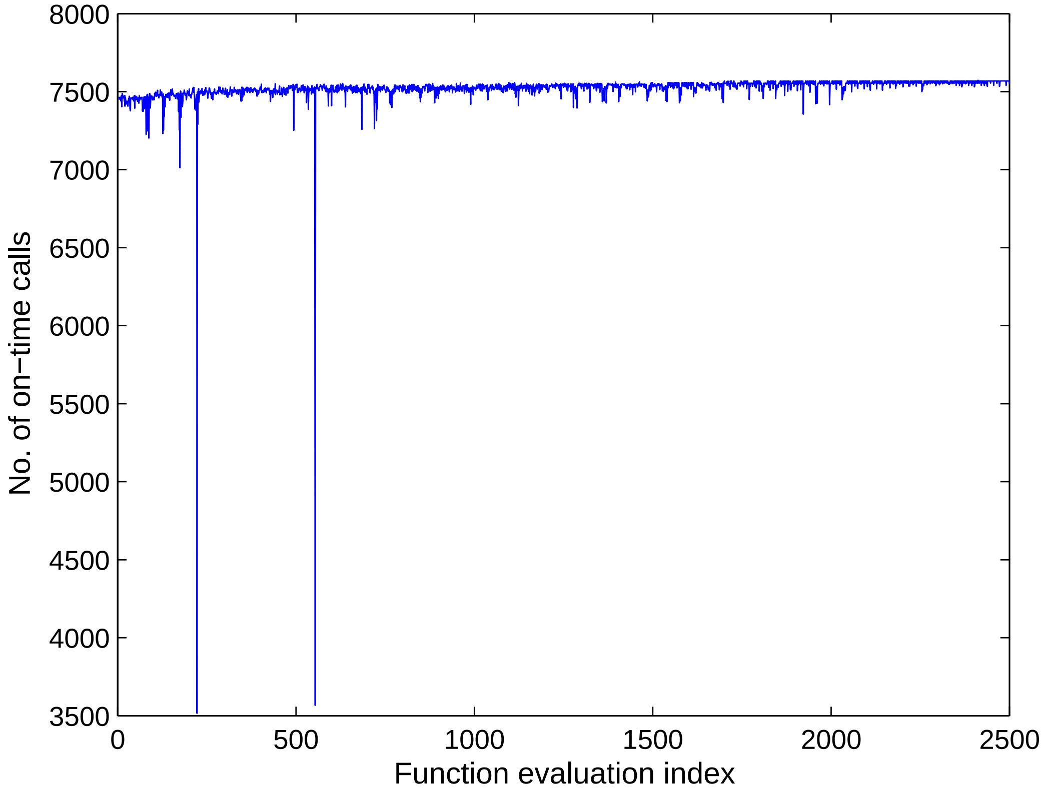


Figure 8.2: Tuning results by Nelder-Mead using the training dataset for Scenario 1B.

optimised IP move-up policy is similar to that of the optimised ranked-base all-ambulance move-up policy, both of which dominate the other two policies. In terms of the expected percentage of calls reached on time, the three IP move-up policies improve on the corresponding ranked-base all-ambulance move-up policies by no more than 0.38% (Table 8.3); these differences are statistically insignificant. The expected average response times are also statistically equivalent.

The empirical results above suggest that there are small differences in response-time statistics between the optimised IP move-up policies and ranked-base all-ambulance move-up policies. However, Tables 8.5 - 8.7 show that there are significant differences in the results for the move-up cost measures (defined in Section 7.6.1) under the optimised IP move-up policies and ranked-base all-ambulance move-up policies, which are estimated

	Un-tuned IP move-up	Optimised IP move-up	Gain
Scenario 1A	51.12%	51.97%	0.85%
Scenario 1B	71.32%	72.41%	1.09%
Scenario 2	62.94%	63.4%	0.46%

Table 8.2: The percentage of calls reached on time in the training dataset using an un-tuned IP move-up policy based on the initialisation procedure shown in Section 8.4 and the optimised IP move-up policy when using the un-tuned policy as the initial solution for each scenario.

	Static	RBFA	RBAA	IP move-up
Scenario 1A	47.39% $\mp$ 0.2%	47.64% $\mp$ 0.2%	50.97% $\mp$ 0.2%	51.31% $\mp$ 0.2%
Scenario 1B	66.34% $\mp$ 0.1%	66.91% $\mp$ 0.1%	71.58%0.1%	71.84% $\mp$ 0.1%
Scenario 2	56.1% $\mp$ 0.1%	56.67% $\mp$ 0.1%	61.67% $\mp$ 0.1%	61.76% $\mp$ 0.1%

Table 8.3: The 95% confidence intervals for the expected percentage of calls reached on time using the optimised static policy, ranked-base free-ambulance move-up policy (RBFA), ranked-base all-ambulance move-up policies (RBAA) and IP move-up policy for each scenario, estimated using the associated test datasets.

	Static	RBFA	RBAA	IP move-up
Scenario 1A	10.3 $\mp$ 0.03	10.13 $\mp$ 0.03	9.65 $\mp$ 0.03	9.62 $\mp$ 0.02
Scenario 1B	7.66 $\mp$ 0.02	7.67 $\mp$ 0.02	7.10 $\mp$ 0.02	7.09 $\mp$ 0.01
Scenario 2	8.93 $\mp$ 0.02	8.88 $\mp$ 0.02	8.22 $\mp$ 0.02	8.23 $\mp$ 0.02

Table 8.4: The 95% confidence intervals for the expected average response time using the optimised static policy, ranked-base free-ambulance move-up policy (RBFA), ranked-base all-ambulance move-up policies (RBAA) and IP move-up policy for each scenario, estimated using the associated test datasets.

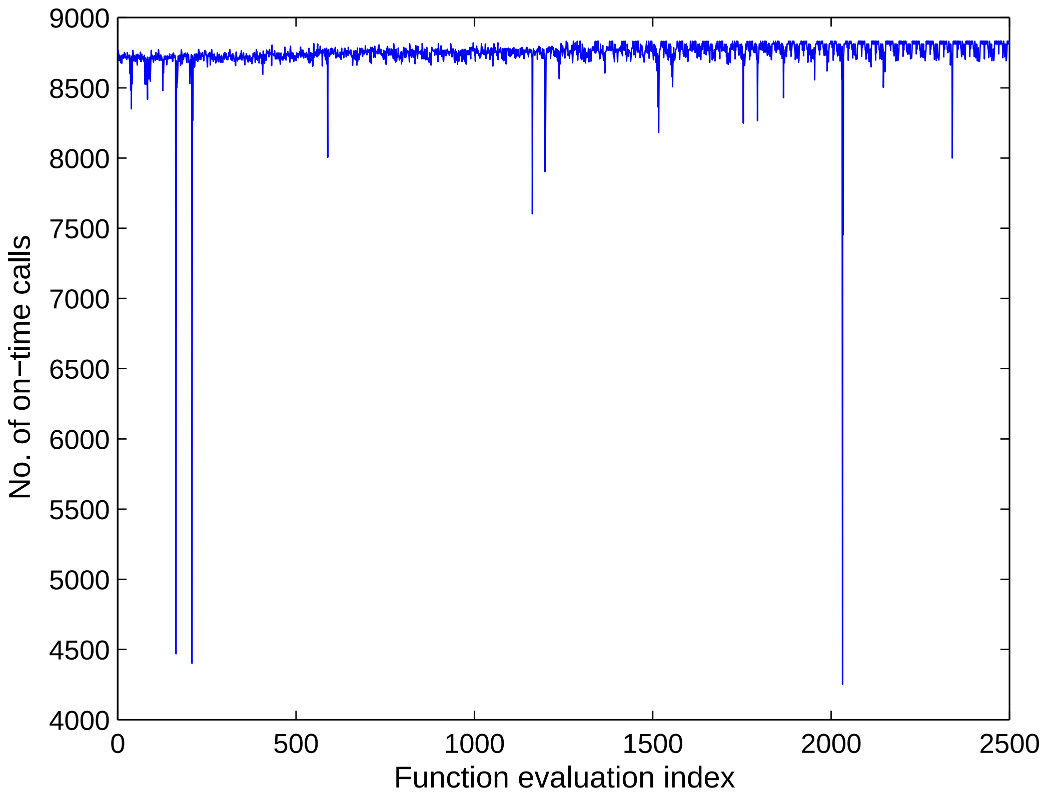
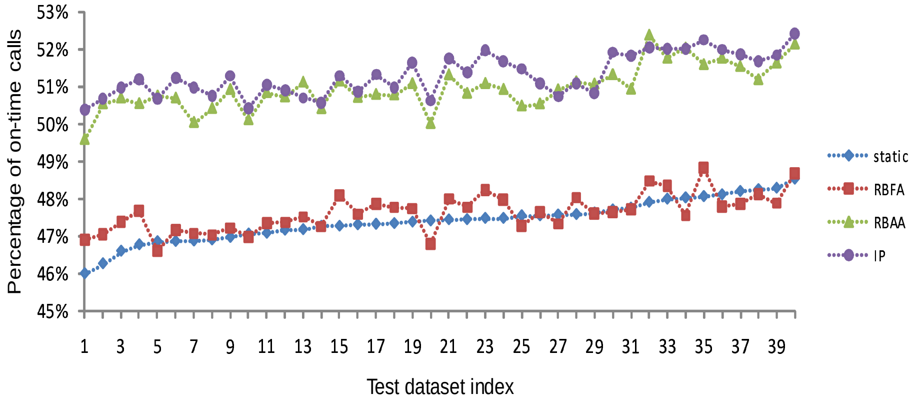


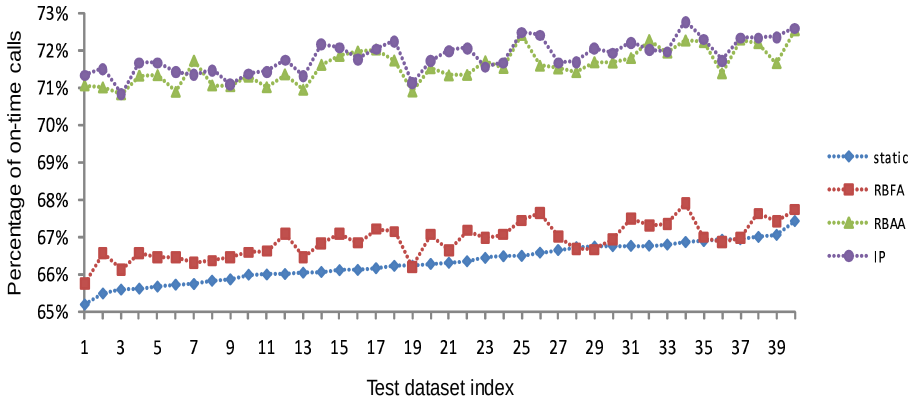
Figure 8.3: Tuning results by Nelder-Mead using the training dataset for Scenario 2.

using the corresponding test datasets. The last column of each of these tables shows the 95% confidence interval for the expected reduction in percetage with respect to each cost measure, estimated by calculating the ratio of the performance difference between the IP move-up policy and the ranked-base all-ambulance move-up polcy in each of the associated test datasets.

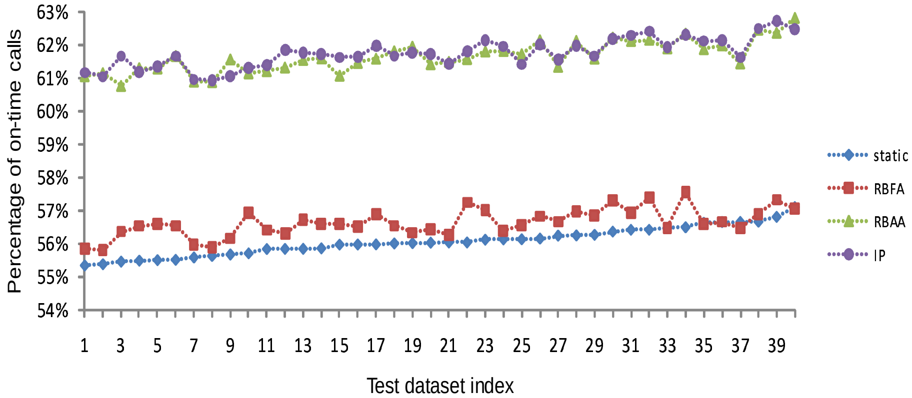
From Tables 8.5 - 8.7, we see that the optimised IP move-up policies outperform the optimised ranked-base all-ambulance move-up policies for all three scenarios, in terms of the move-up costs. For each of the three scenarios, using the results under the ranked-base all-ambulance move-up policy as the yardstick, the driving distances under the IP move-up policy are reduced by at least 14.4%; the average number of attempted idle-at-base moves per vehicle per day is reduced by at least 24.8%; the average number of redirections per



(a) Scenario 1A



(b) Scenario 1B



(c) Scenario 2

Figure 8.4: The percentage of calls reached on time using the optimised static policy, ranked-base free-ambulance move-up policy (RBFA), ranked-base all-ambulance move-up policy (RBAA) and IP move-up policy for each of the test datasets in Scenarios 1A, 1B and 2, respectively. The test data are sorted by the performance under the associated static policies

	RBAA	IP	Exp. reduction (%)
Exp. Avg. driving distances per day (km)	76.4 ∓ 0.3	65.5 ∓ 0.2	$14.4\% \pm 0.4\%$
Exp. Avg. number of attempted idle-at-base moves day	5.7 ± 0.2	4.3 ± 0.2	$24.8\% \pm 0.3\%$
Exp. Avg. number of redirections per day	29.2 ± 0.1	$16.9 \pm 0.7\%$	$41.9\% \pm 0.2\%$
Exp. Avg. number of back-to-base redirections per day	2.9 ± 0.1	0.89 ± 0.1	$69.1\% \pm 0.3\%$
Exp. Avg. number of relocations per day	34.9 ± 0.1	21.3 ± 0.1	$39.2\% \pm 0.1\%$

Table 8.5: The 95% confidence intervals for the move-up costs (defined in Section 7.6.1) under the optimised ranked-base free-ambulance move-up policy (RBFA) and IP move-up policy, estimated using the 40-test datasets for Scenario 1A with 12 ambulances and 9 calls/hr. The reduction figure for each performance meaure in the last column is estimated based on the ratio of the performance difference between the IP move-up policy and the ranked-base all-ambulance move-up polcy in each test dataset.

	RBAA	IP	Exp. reduction (%)
Exp. Avg. extra driving distances per vehicle per day (km)	87.3 ± 0.3	65.3 ± 0.3	$25.3\% \pm 0.2\%$
Exp. Avg. number of attempted idle-at-base moves per vehicle per day	5.0 ± 0.1	3.9 ± 0.1	$22.2\% \pm 0.3\%$
Exp. Avg. number of redirections per vehicle per day	28.1 ± 0.1	$18.6 \pm 0.1\%$	$33.8\% \pm 0.2\%$
Exp. Avg. number of back-to-base redirections per vehicle per day	2.5 ± 0.1	1.1 ± 0.1	$54.3\% \pm 0.4\%$
Exp. Avg. number of relocations per vehicle per day	34.9 ± 0.1	22.5 ± 0.1	$31.9\% \pm 0.1\%$

Table 8.6: The 95% confidence intervals for the move-up costs (defined in Section 7.6.1) under the optimised ranked-base free-ambulance move-up policy (RBFA) and IP move-up policy, estimated using the 40-test datasets for Scenario 1B with 16 ambulances and 9 calls/hr. The reduction figure for each performance meaure in the last column is estimated based on the ratio of the performance difference between the IP move-up policy and the ranked-base all-ambulance move-up polcy in each test dataset.

	RBAA	IP	Exp. reduction (%)
Exp. Avg. extra driving distances per vehicle per day (km)	70.4 ± 0.3	56.0 ± 12.7	$20.4\% \pm 0.2\%$
Exp. Avg. number of attempted idle-at-base moves per vehicle per day	5.6 ± 0.1	4.0 ± 0.2	$29.2\% \pm 0.3\%$
Exp. Avg. number of redirections per vehicle per day	38.7 ± 0.1	$23.1 \pm 0.1\%$	$40.3\% \pm 0.1\%$
Exp. Avg. number of back-to-base redirections per vehicle per day	3.0 ± 0.1	0.9 ± 0.1	$68.9\% \pm 0.3\%$
Exp. Avg. number of relocations per vehicle per day	44.3 ± 0.1	$27.1 \pm .1$	$38.9\% \pm 0.1\%$

Table 8.7: The 95% confidence intervals for the move-up costs (defined in Section 7.6.1) under the optimised ranked-base free-ambulance move-up policy (RBFA) and IP move-up policy, estimated using the 40-test datasets for Scenario 2 with 16 ambulances and 12 calls/hr. The reduction figure for each performance meaure in the last column is estimated based on the ratio of the performance difference between the IP move-up policy and the ranked-base all-ambulance move-up polcy in each test dataset.

vehicle per day is reduced by at least 33.8%; the average number of back-to-base redirections per vehicle per day is reduced by at least 54.3%; the average number of (total) relocations per vehicle per day is reduced by at least 31.9%.

The results for the move-up cost comparisons are encouraging, as they suggest that the optimised IP move-up policies give equally-good response-time performance compared to the optimised ranked-base all-ambulance move-up policies, but with reduced move-up costs. In other words, the IP move-up policies are more intelligent and viable. In practice, a reduction in driving distances can be translated into cost savings in fuel and vehicle maintenance, a smaller likelihood of traffic accident occurrences, and a decreased chance of developing health conditions such as back pains due to crews spending less time on the road. A reduction in the number of attempted idle-at-base moves, redirections, and back-to-base redirections means that less crew frustration due to move-up is expected. These reductions give the corresponding move-up policy extra competitiveness when compared to a static policy which, relatively speaking, has zero cost but produces worse response-time performance.

The increased intelligence of move-up based on the IP move-up model compared to the ranked-base all-ambulance move-up model is what we have hoped for based on our insights gained from our small-scale DP models. Recall that the key difference between the two move-up models is that the ranked-base all-ambulance move-up model forces n free ambulances into a unique target configuration $C(n)$, while the IP move-up model does not. Tables 8.8 - 8.10 report, for each of the three training datasets, the percentage of move-up decisions made with $n = 1, 2...$ free ambulances, the number of target configurations chosen for $n = 1, 2...$ free ambulances, and the percentage that the most chosen (top) configuration accounts for given there are $n = 1, 2, ..$ free ambulances[1].

Comparing the proportions of move-up decisions made with $n = 1, 2...$ free ambulances in Tables 8.8 and 8.9, where Table 8.8 corresponds to Scenario 1A's training dataset with 12 ambulances and 9 calls/hr and Table 8.9 corresponds to Scenario 1B's training dataset with 16 ambulances and 9 calls/hr, we see that a move-up decision is more likely to be made

[1]We only use training datasets for the following analysis, as we see similar results in the test datasets.

with more ambulances available in Scenario 1B's training dataset than in Scenario 1A's training dataset, which is expected because there are more ambulances under Scenario 1B than Scenario 1A. Table 8.10 corresponds to Scenario 2's training dataset with 16 ambulances and 12 calls/hr. We see that a move-up decision is more likely to be made with fewer ambulances available in Scenario 2's training dataset than in Scenario 1B's training dataset, which is, again, expected because of the increased arrival rate.

An analysis of the data showed that for each of the three scenarios, the top configuration for each number of free ambulances under the optimised IP move-up policy is the same as the configuration defined by the corresponding optimised ranked-base all-ambulance move-up policy. However, we see that for Scenario 1A's training dataset (Table 8.8), a majority (about 90.2%) of move-up decisions is made with 3 to 10 free ambulances. The percentage that the top configuration, for $n = 3, ..., 10$ ambulances, accounts for ranges from about 29.9% to 73.3%. In other words, the percentage that the non-top configurations, for $n = 3, ..., 9$ ambulances account for, ranges from about 26.7% to 70.1%. For Scenario 1B's training dataset (Table 8.9), about 92.2% of move-up decisions are made with 7 to 14 free ambulances. The percentage that the top configuration, for $n = 7, ..., 14$ ambulances, accounts for ranges from about 25.6% to 79.8%. For Scenario 2's training dataset (Table 8.10), about 92.7% of move-up decisions are made with 4 to 13 free ambulances. The percentage that the top configuration, for $n = 4, ..., 13$ ambulances, accounts for ranges from about 14.9% to 66.7%. These figures indicate that the non-top configurations account for a significant proprotion of move-up decisions, which supports our belief – there are multiple 'good' configurations for a given number of free ambulances and the one chosen for move-up should depend on not just the number of free ambulances, but also factors such as the locations of free ambulances and busy ambulances.

We make a note about the tuned weights for Scenario 1B with 16 ambulances on duty and 9 calls/hr and Scenario 2 with 16 ambulances on duty and 12 calls/hr. We find that the tuned weight under Scenario 2 is about four times larger than that under Scenario 1B[1]. This result means that it is more 'expensive' for an ambulance to move given the same travel

[1] Note that for fair comparisons, all the tunable parameters for Scenario 2 are normalised using the highest marginal reward for Scenario 1A as the reference.

time under Scenario 2 than Scenario 1B. Therefore, it is not surprising that the number of chosen target configurations for each given number of free ambulances under Scenario 2 (Table 8.9) is larger than that under Scenario 1B (Table 8.10).

Finally, recall that an IP move-up model by Richards [39] has been embedded into Optima Predict. This model, with its default input settings, is tested for our three scenarios. We find that the performance using this model is worse than that under the (benchmark) static policies, suggesting that a more careful calibration for the input settings is needed. However, as discussed in Chapter 2, the model requires, for each demand point, two input parameters, i.e. call demand and a target number of ambulances up to which each additional ambulance that can cover that demand point contributes to a reward function. Typically, the number of demand points is very large compared to the number of stand-by locations. Consequently, it may be difficult to use a systematic mechanism such as numerical optimisation to find a good set of input parameters for the model by Richards. We consider designing algorithms for the model input settings as a possible future research direction.

8.5.2 Effectiveness of the Nelder-Mead Algorithm

We now return to the question raised in Section 8.5.1: the computation required by Nelder-Mead for the tuning process is very heavy (approximately 6 to 9 days); however, the final best solution (for the tunable parameters) leads to a small performance improvement compared to the initial solution generated using our procedure presented in Sections 8.4.1 and 8.4.2. This result makes it natural to question whether Nelder-Mead is an effective algorithm for improving the solution quality for our IP model.

In this section, we show that Nelder-Mead is effective for our model and the reason for the small performance improvement using Nelder-Mead is because our initialisation procedure is of high quality, in which case there is little room for further improvement. To demonstrate the effectiveness of Nelder-Mead, we perform another Nelder-Mead run but this time, start from a poor initial solution for each of our three scenarios.

The vertex (initial solution) required to create the initial working simplex for each scenario is now defined by setting all the tunable parameters to 1. For the following analysis,

No. of free ambulances	Move-up decisions (%)	No. of target configurations	Top configuration (%)
1	1.92%	6	47.63%
2	3.85%	14	34.62%
3	7.01%	26	29.88%
4	10.66%	40	34.50%
5	13.79%	49	48.73%
6	15.80%	54	45.49%
7	15.81%	45	53.72%
8	13.52%	58	50.62%
9	9.00%	74	73.33%
10	4.60%	64	23.16%
11	1.72%	43	17.22%
12	0.31%	24	12.31%

Table 8.8: The percentage of move-up decisions made with $n = 1, ..., 12$ free ambulances, the number of target configurations chosen for $n = 1, .., 12$ free ambulances, and the percentage that the most chosen (top) configuration accounts for given there are $n = 1, ..., 12$ free ambulances in Scenario 1A's training dataset under the associated optimised IP move-up policy.

No. of free ambulances	Move-up decisions (%)	No. of target configurations	Top configuration (%)
1	0.04%	2	66.67%
2	0.11%	6	31.82%
3	0.29%	14	24.59%
4	0.65%	28	24.26%
5	1.36%	48	30.53%
6	2.91%	72	24.18%
7	5.69%	114	25.61%
8	9.36%	118	28.65%
9	13.09%	116	28.62%
10	15.84%	122	36.69%
11	16.88%	110	51.36%
12	15.19%	59	51.07%
13	10.53%	37	68.03%
14	5.60%	27	79.76%
15	2.07%	16	82.64%
16	0.39%	7	48.15%

Table 8.9: The percentage of move-up decisions made with $n = 1, ..., 16$ free ambulances, the number of target configurations chosen for $n = 1, .., 16$ free ambulances, and the percentage that the most chosen (top) configuration accounts for given there are $n = 1, ..., 16$ free ambulances in Scenario 1B's training dataset under the associated optimised IP move-up policy.

No. of free ambulances	Move-up decisions (%)	No. of target configurations	Top configuration (%)
1	0.71%	7	35.42%
2	1.26%	16	23.78%
3	2.34%	35	17.37%
4	3.91%	57	14.82%
5	5.79%	78	16.29%
6	8.21%	109	21.65%
7	10.89%	140	19.85%
8	13.22%	153	27.45%
9	14.38%	144	31.69%
10	13.49%	108	31.60%
11	11.02%	127	50.03%
12	7.65%	108	66.54%
13	4.19%	94	21.16%
14	1.63%	67	15.60%
15	0.46%	38	10.32%
16	0.07%	15	15.79%

Table 8.10: The percentage of move-up decisions made with $n = 1, ..., 16$ free ambulances, the number of target configurations chosen for $n = 1, .., 16$ free ambulances, and the percentage that the most chosen (top) configuration accounts for given there are $n = 1, ..., 16$ free ambulances in Scenario 2's training dataset under the associated optimised IP move-up policy.

	Un-tuned move-up	Optimised move-up	Improvement
Scenario 1A	37.12%	51.90%	14.78%
Scenario 1B	52.71%	71.46%	18.75%
Scenario 2	47.20%	62.28%	15.08%

Table 8.11: The percentage of calls reached on time in the training dataset for each scenario using an un-tuned IP move-up policy (in which all the tunable parameters are set to be 1) and an optimised IP move-up policy when using the un-tuned policy as the initial solution.

we refer to this simple initialisation procedure as the unit-value procedure, and our initialisation procedure in Section 8.4 as the simulation-based procedure.

Figures 8.5 - 8.7 compare, for each of our three scenarios, the Nelder-Mead tuning results for the two initial solutions generated by the unit-value procedure and the simulation-based procedure. The x-axis represents the function evaluation index and the y-axis represents, for the associated training dataset, the best function value found after the K^{th} function evaluation. Note that for each scenario, Nelder-Mead with the unit-value initial solution is, again, terminated due to the number of function evaluations reaching the limit of 2500.

From the plots, we see that the unit-value initial solutions perform significantly worse than the simulation-based initial solutions. Therefore, it is reasonable to consider the unit-value initial solutions as poor starting solutions, and the simulation-based initial solutions as good starting solutions.

We observe that when Nelder-Mead is applied to each of the three poor starting solutions, it is able to make a significant improvement on the solution quality. This observation supports our claim at the beginning of this section: Nelder-Mead is effective in improving the solution quality for our model and our simulation-based procedure provides high-quality initial solutions.

Table 8.11 summarises the performance of the unit-value initial solution (un-tuned IP move-up policy) and the corresponding final best solution (optimised IP move-up policy) with respect to the corresponding training dataset. For each scenario, Nelder-Mead is able improve the solution quality to a great extent; at least 14.78% extra calls are reached on time using the final best solution.

We make a comment about the solution quality vs the total run time for the two

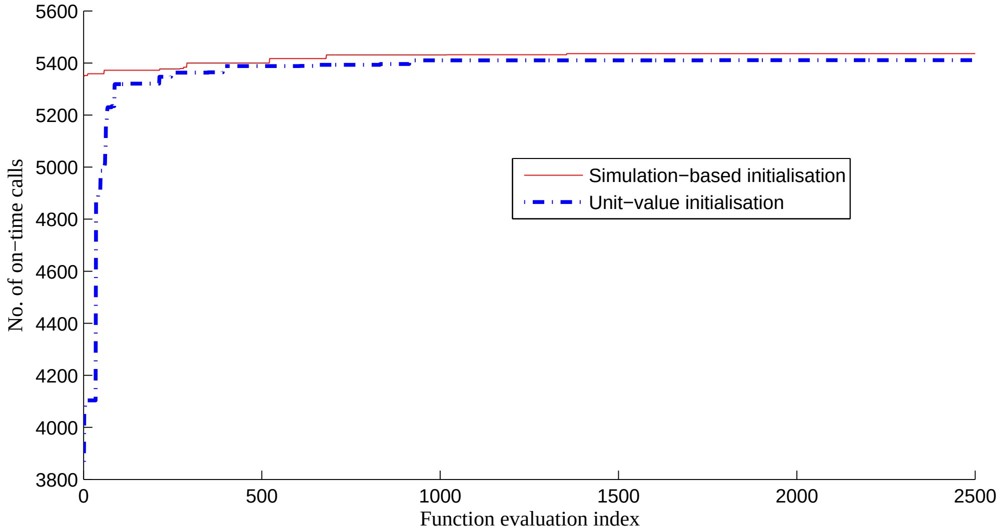


Figure 8.5: The best function value for the training dataset after the K^{th} evaluation by Nelder-Mead with an unit-value initial solution and a simulation-based initial solution under Scenario 1A.

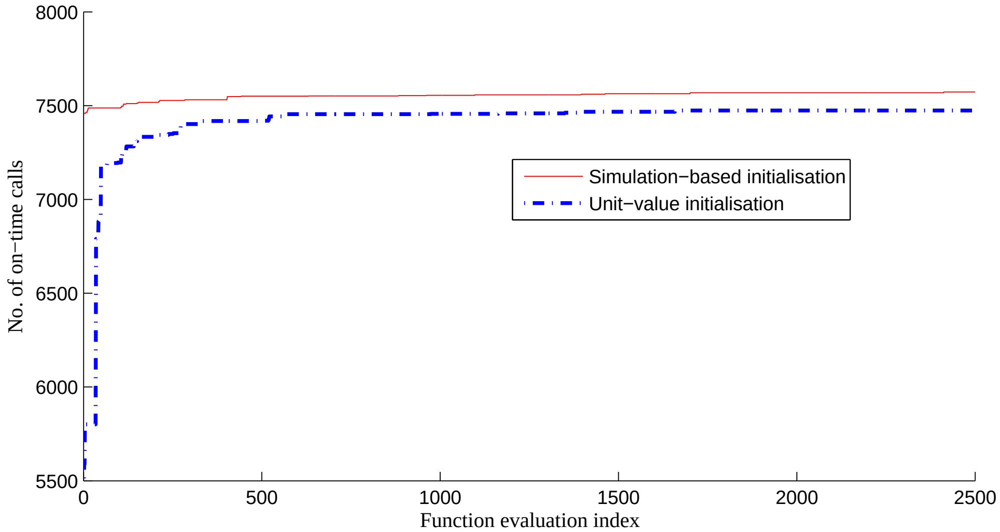


Figure 8.6: The best function value for the training dataset after the K^{th} evaluation by Nelder-Mead with an unit-value initial solution and a simulation-based initial solution under Scenario 1B.

initialisation procedures. Compared to the simple unit-value procedure, the simulation-based procedure requires simulations (1 run for initialising the marginal rewards and 30

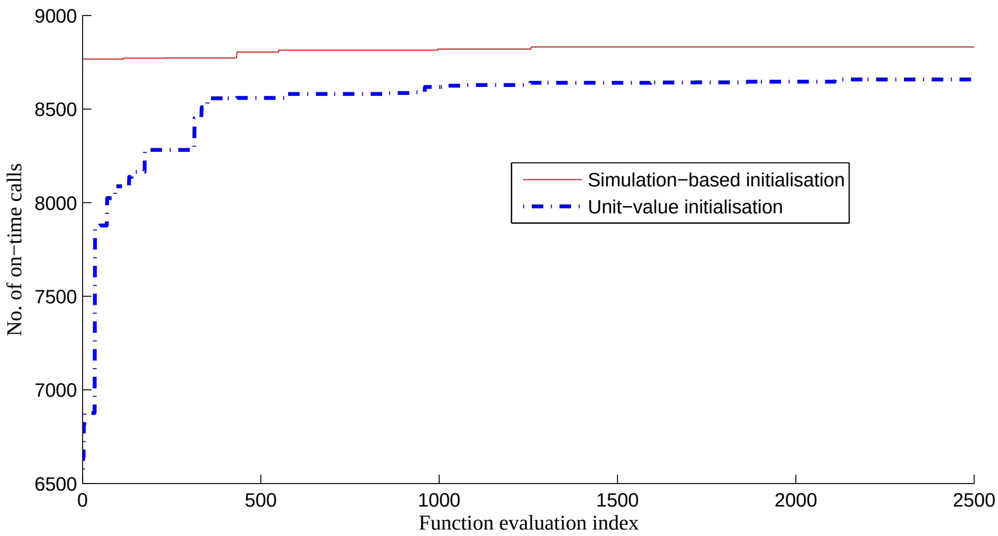


Figure 8.7: The best function value for the training dataset after the K^{th} evaluation by Nelder-Mead with an unit-value initial solution and a simulation-based initial solution under Scenario 2.

runs for initialising the weight for each of our scenarios) to create a solution, while the unit-value procedure is completed instantly. Moreover, keep in mind that the simulation-based procedure also requires an optimised ranked-base all-ambulance move-up policy as an input parameter, which can take days to obtain. However, as shown in Figures 8.5-8.7, when Nelder-Mead is applied to each of the (poor) unit-value initial solutions, it takes at least 240 simulation runs before the best function value is similar to (but never better than) the one based on the simulation-based initial solution. Therefore, if an optimised ranked-base all-ambulance move-up policy is available, we recommend the use of the simulation-based procedure. Otherwise, the unit-value procedure can be used, which is expected to give similar-quality solutions.

8.5.3 Benefits of Move-up

In this section, we would like to address the question whether move-up is worthwhile or not. Our understanding is that it is difficult to answer this question in a generic fashion; many factors, as discussed below, need to be taken into account.

Move-up often leads ambulance crews to spend more time on the road, resulting in an increased chance of developing neck pains and back pains [34]. On-road ambulances may be repeatedly redirected to different target bases in a short amount of time, which can be perceived as frustrating and pointless moves from the crews' perspective. Driving distances with move-up are expected to increase, which results in higher fuel and maintenance costs. Therefore, the willingness to implement move-up may vary significantly among different EMS providers.

Nevertheless, if a move-up strategy is shown to help reduce response times, we think it deserves to be considered for implementation given the social impact of EMS operations. Besides presenting the performance improvement by move-up to an EMS manager, we think it is also helpful to present the additional number ΔN of ambulances required in the static location model so as to meet/exceed the service levels provided by move-up; this information is useful, as purchasing an ambulance and maintaining it can cost up to 1 million dollars per year [34], which is a large capital investment.

For each of the three scenarios above, it is reasonable to consider the optimised IP move-up policy presented in Section 8.5.1 as the best policy when maximising the percentage of calls reached on time is the primary objective. To determine ΔN, we repeatedly increase the number of ambulances on duty by 1 and then run Algorithm 2 to obtain a re-optimised static policy. For each re-optimised static policy, we estimate its performance using the test datasets as before. We stop the increment when the last evaluated re-optimised static policy exceeds the performance given by the corresponding optimised IP move-up policy.

Table 8.12 reports, for each of the three scenarios, the 95% confidence intervals for the expected marginal increase in the percentage of calls reached on time under each re-optimised static policy, estmated using the test datasets. The statistics for the average response times under the re-optimised static policies are also shown in Table 8.12.

From Table 8.12, we see that for Scenario 1A, one additional ambulance is required by the static model to outperform the optimised IP move-up policy; the extra percentage of calls reached on time using the optimised 13-ambulance static policy compared to the 12-ambulance IP move-up policy is about 0.67%. For Scenario 1B, two additional ambulances

	Scenario 1A	Scenario 1B		Scenario 2	
No. of ambulances in the static model	13	17	18	17	18
Exp. marginal increase in on-time calls (%)	$4.57\% \pm 0.2\%$	$3.12\% \pm 0.1\%$	$2.81\% \pm 0.1\%$	$5.10\% \pm 0.2\%$	$3.62\% \pm 0.1\%$
Exp. average response time (minutes)	9.32 ± 0.02	7.36 ± 0.02	7.11 ± 0.01	8.32 ± 0.02	7.91 ± 0.01

Table 8.12: The number of ambulances in each re-optimised static policy, the 95% confidence intervals for the expected marginal increase in calls reached on time, and average response time under the associated static policy for each scenario. The confidence intervals are estimated using the results for the corresponding test datasets.

are required and the extra percentage of calls reached on time using the optimised 18-ambulance static policy compared to the 16-ambulance IP move-up policy is about 0.42%. For Scenario 2, we also need two addtional ambulances in the static model to outperform the optimised IP move-up policy by about 3.04%; however, we see that the performance with only one addtional ambulance in the static model is close to (about 0.54% worse than) that using the IP move-up policy. Therefore, it is arguable whether the second additional ambulance is worth the investment in practice for Sceanrio 2.

We also observe that for the same arrival rate, as the number of ambulances increases in the static model, the average response time decreases as expected. It is worth noting that although the average response time is not used as the objective for the static policy re-optimisation, we can use the average-response-time statistics to obtain the same results on the number of additional ambulances required in the static model to outperform the IP move-up policies.

8.6 Summary

This chapter has been devoted to the formulation of an IP move-up model. Some of the insights obtained from the small-scale DP model in Chapter 5 were employed. Furthermore, numerical optimisation (Nelder-Mead) was used to tune some of the model parameters in a hope of forming a high-quality move-up policy.

Computational experiments in Section 8.5.1 showed that after the tuning process, the optimised IP move-up policies and ranked-base all-ambulance move-up policies presented in the last chapter gave similar performance in terms of reducing response times, both of which outperform the optimised ranked-base free-ambulance move-up policies and static policies. However, the IP move-up policies resulted in smaller move-up costs, which means they were more effective than the ranked-base all-ambulance move-up policies.

The computational burden of implementing Nelder-Mead for our IP move-up model was very heavy while the improvement of solution quality was relatively small. Consequently, in Section 8.5.2, we investigated its effectiveness by initialising the tunable parameters in an ad-hoc way. The results suggested that Nelder-Mead was able to tune the parameters such that the system performance after tuning was significantly better; the small improvement observed in Section 8.5.1 was due to the high-quality initialisation scheme we proposed for the tunable parameters.

Finally, we addressed the question of whether move-up is worthwhile or not. The question is hard to answer, as move-up can lead to health issues for the crews, increased fuel and maintenance costs, etc. Therefore, the willingness of performing move-up can vary among different EMS providers. On the other hand, large capital investments may be needed in order to obtain similar performance improvements using a static location strategy.

9

Conclusions and future research

9.1 Conclusions

This thesis devoted three chapters to the study of optimal move-up under the dynamic programming (DP) framework. The DP models are difficult to extend for practical problems. The main use of the models is to gain insights from optimal move-up policies in small settings.

In the third chapter, optimisation of a single-ambulance move-up model to maximise the benefit for the next call was studied. Theoretical results on the characteristics of such optimal move-up policies and value functions were established. In the fourth chapter, a move-up model for a single ambulance to maximise the average benefit over an infinite horizon was presented. Numerical studies were performed for the two single-ambulance move-up models; the static model was used to provide benchmarks. The results showed that the single-ambulance infinite-horizon move-up model always performed at least as well as the static model, while it was difficult to draw a conclusion about the performance of the 'simpler' single-ambulance next-call move-up model. Nevertheless, it was possible to observe that, in some conditions, the next-call model could outperform the static model and give results similar to those of the infinite-horizon model.

The fifth chapter was an extension of the fourth chapter, considering two ambulances for move-up in order to maximise the average benefit over an infinite horizon. Simplified numerical experiments were analysed. We observed that the optimal move-up policies outperformed the optimal static policies in all test instances. Furthermore, the results provided some interesting move-up insights which were, to our knowledge, never discussed in the literature. A principal finding was that to decide the stand-by location for each free ambulance, many factors such as hospital locations, arrival rate, the spatial distribution of demand, status of busy ambulances, and rewards along a move-up path need to be taken into account; this finding reflected the complexity of designing high-performance move-up policies.

Chapter 6 was a transition chapter taking us to the study of three move-up models for realistic-sized EMS operations. In this chapter, we first introduced the simulation software Optima Predict which we used for policy evaluations. We then discussed a simulation environment to compare and test ambulance location models. Furthermore, three scenarios based on the simulation environment were established and results for the corresponding static policies derived from a simulation-based local search algorithm were presented. The static policies were used to benchmark the performance of the move-up models presented in the subsequent chapters.

In Chapter 7, we presented a ranked-base free-ambulance move-up model and a ranked-base all-ambulance move-up model. As the names suggest, both models are based on a priority list that ranks stand-by locations (ambulance bases). A simulation-based local search algorithm was proposed to optimise the list (rankings) used to form move-up policies based on the two models. The results for the optimised move-up policies and the benchmark static policies were compared. We observed that the ranked-base all-ambulance move-up policies significantly improved the system performance, i.e. the percentage of calls reached on time (which is the objective to maximise) and the average response time, while the ranked-base free-ambulance move-up policies performed marginally better than the static policies[1]. However, it was difficult to conclude that the ranked-base all-ambulance move-up policies were the best options, as they were associated with higher move-up 'costs' (such as extra driving distances, attempted base-to-base ambulance moves and on-road-ambulance redirections) than the static policies and the ranked-base free-ambulance move-up policies.

In Chapter 8, we proposed an IP move-up model. A simulation-based initialisation procedure was proposed to determine the initial values for some of the tunable input parameters in the IP model and numerical optimisation (Nelder-Mead) was used for tuning

[1] Maxwell [34] found a more significant improvement using the ranked-base free-ambulance move-up model for data based on Edmonton.

the parameters. Computational results showed that the optimised IP move-up policies gave similar service levels to the ranked-base all-ambulance move-up policies. However, the optimised IP move-up policies were associated with lower move-up costs. Therefore, the optimised IP move-up policies were more effective. More importantly, we showed that the optimised IP move-up policies use multiple configurations for a given number of free ambulances, instead of a unique configuration under a ranked-base all-ambulance move-up policy. This result supports our belief based on our insights gained from our small-scale DP models that there are multiple good configurations for a given number of ambulances and the target configuration should depend on not just the number of free ambulances, but also other factors such as locations of free and busy ambulances.

The Nelder-Mead algorithm used for the tuning process led to heavy computation burdens but small improvements for the solution quality. Additional experiments were conducted to show that the small improvements were caused by our high-quality initialisation procedure for the tunable parameters and Nelder-Mead was able to improve the solution quality significantly when starting from a poor solution. Finally, we also reported, for each of the three scenarios, the additional number of ambulances required in a re-optimised static policy to meet/exceed the performance of the optimised IP move-up policy. As putting even one additional ambulance in service involves a large financial commitment, this information provides a useful reference to measure the value of move-up.

9.2 Future Research Directions

The three move-up models developed in Chapters 7 and 8, which are aimed for large-scale EMS operations, are tested in simplified systems, as discussed in Section 6.2. A natural extension would be to test them in more complicated systems. For example, in practice, the arrival rate and spatial distribution of call demand can vary significantly over time. Calls often have different priorities and travel times are different between off-peak and peak traffic hours. Moreover, in the simplified systems, only ambulance bases are considered as candidate stand-by locations for move-up, while in the real-world EMS operations, some street corners can also be used as stand-by locations.

When calls with different priorities are considered, different EMS providers may have different perspectives for measuring the overall performance. For example, only maximising the performance for calls with the highest priority may no longer be appropriate and a multi-objective optimisation problem may arise. Because we use simulation-based optimisation for our move-up models, an advantage is that it is easy to evaluate the performance with respect to objectives that can be difficult to approximate using the mathematical modelling approach.

When street corners are considered for move-up, the performance of a move-up model may increase compared to the case in which only ambulance bases are considered for move-up, as street corners in high-demand areas can lead to smaller response times. However, standing by at a street corner is probably more 'costly' in terms of crew frustrations than at a base where crews do not have to sit in the vehicle. Therefore, taking street corners into account for a move-up policy may reduce the policy's practicality.

Regarding the IP move-up model developed in Chapter 8, a fruitful area for future research would be to employ the move-up insights obtained from our small-scale DP models in a more sophisticated way. The model, as discussed, can also be viewed as an ADP model. From the ADP perspective, more careful approximation architectures deserve to be explored. At this stage, the cost of moving an ambulance to a base in the IP model is mainly determined by the associated travel time. Recall that in Chapter 5, we showed that the 'quality' of a move-up path can be as important as that of a final stand-by location. A possible extension of the IP model to address this problem is to consider a multi-stage IP model in which each stage corresponds to some time interval, so that the rewards associated with an ambulance's move-up path can be taken into account. However, the computation burden for solving such a multi-stage IP model is expected to increase. Heuristic solution techniques may be required, as the time taken to make a real-time move-up decision should be very short.

We also envision that a possible use of the IP move-up model is to construct multi-configuration compliance-table move-up policies, i.e. multiple target configurations for a given number of free ambulances, which has not been studied in the literature. One possible mechanism for creating a multi-configuration compliance-table move-up policy is to extract some of the frequently chosen configurations based on the IP model, and design an algorithm (which may involve simulation-based tuning) for deciding which target configuration is the optimal one whenever a move-up decision is required.

Finally, the performance of other numerical optimisation methods for tuning parameters in the IP model should be investigated and compared with the Nelder-Mead method used in this work.

Appendices

.1 Proof of Correctness of Algorithm 1

In this appendix, we prove the validity of Algorithm 1.

We proceed by induction. Let $V(k)$ denote the value currently assigned to node k by Algorithm 1, and $V(k)^*$ denote the correct value for node k. Let us renumber the n nodes in decreasing order of their true $V(k)^*$ values, giving $V(1)^* \geq V(2)^* \geq ... \geq V(n)^*$, where nodes are ordered by increasing index for breaking ties. Assume nodes $P = \{1, 2, ..., p-1\}$ currently have permanent labels, and nodes $T = \{p, p+1, ..., n\}$ have temporary labels.

(i) If all permanently labelled nodes have their correct $V(k) = V(k)^*$ values, then the next step of Algorithm 1 will permanently label node p with its correct $V(p) = V(p)^*$ value.

Proof: The value $V(k)$ of any temporarily labelled node k is calculated by Algorithm 1 as:

$$V(k) = \max_{k' \in N(k)} \left(r(k), (1 - e^{-\lambda \Delta t_{(k,k')}}) r(k) + e^{-\lambda \Delta t_{(k,k')}} \max_{j \in N_k \cap P} V(j)^* \right) \quad (1)$$

Comparing this with the optimality equation, (3.3), we see that this equation excludes all neighbouring nodes of k that have temporary labels. Thus, we must have $V(k) \leq V(k)^* \; \forall \; k \in T$.

Consider now node p. By our ordering assumption, any neighbouring node k excluded in (1) for node p has $V(k)^* \leq V(p)^*$, and so excluding this node in (3.3) would not alter the calculated value $V(p)^*$, making (3.3) and (1) equivalent. Therefore, $V(p) = V(p)^*$, and so node p is ready to be permanently labelled. Furthermore, we have $V(p) = V(p)^* \geq V(k)^* \geq V(k)$, $\forall \; k \in T$, and so node p will be the next node selected for a permanent label by Algorithm 1

(ii) It remains to show that the first permanent label $V(1)$ is calculated correctly.

Proof: Algorithm 1 puts $V(1) = r(1) = \max_{k=1..n} r(k)$, and so V_1 is the reward gained by waiting at node 1, being the node with the largest $r(k)$. Clearly, moving away from this node to wait at some node with a lower $r(k)$ cannot be a better policy, and so $V(1) = V(1)^*$.

This completes our proof.

.2 Optimality Equations for the Two-Ambulance Infinite-Horizon Move-up Model

In this appendix, we provide the optimality equations for all the states in the two-ambulance infinite-horizon move-up model discussed in Chapter 5.

First we consider states $((k_1, \text{Free}),(k_2, \text{Free}))$, $\forall k_1, k_2 \in N$ – one ambulance is free at node k_1 and the other ambulance is free at node k_2. Each ambulance can move to any node $k'_i \in N_{k_i}, i = 1, 2$ in which case the non-zero transition probabilities associated with each of these states are:

$$P\{((k'_1, \text{Free}), (k'_2, \text{Free}))|((k_1, \text{Free}), (k_2, \text{Free}))\} = e^{-\lambda},$$

$$P\{((k_1, i), (k'_2, \text{Free})|((k_1, \text{Free}), (k_2, \text{Free}))\} = (1 - e^{-\lambda})p(x) \quad x \in Q_{k_1},$$

$$P\{((k'_1, \text{Free}), (k_2, i)|((k_1, \text{Free}), (k_2, \text{Free}))\} = (1 - e^{-\lambda})p(x) \quad x \in Q_{k_2}.$$

The first transition corresponds to the state given no call-arrival during the wait interval – ambulance i jumps from node k_i to k'_i, $i = 1, 2$, respectively, after one time-step. The second and third transitions correspond to the states where a call arrives during the wait interval and the closest ambulance gets dispatched. The term $Q_{k_i}, i = 1, 2$, denotes a subset of nodes that are closest to node k_i. If there is a tie for node x, each ambulance has an equal chance of 0.5 to get dispatched to node x.

The immediate reward is independent of which node each ambulance moves to since we use the 'wait-then-jump' scheme and it is the probability of one call arriving during the wait interval multiplied by the reward function $r(k_1, r_2)$ as shown below:

$$(1 - e^{-\lambda})r(k_1, k_2).$$

Therefore, Equation 4.2 gives:

$$\begin{aligned} V((k_1,\text{Free}),(k_2,\text{Free}))+g = \max_{k_i'\in N_i, i=1,2} \Big[&(1-e^{-\lambda})r(k_1,k_2) \\ &+ (1-e^{-\lambda_{(k,k')}})\sum_{i\in N} p(i)V(k,i) \\ &+ e^{-\lambda} V((k_1',\text{Free}),(k_2',\text{Free}))\Big]. \end{aligned}$$

Next we consider states $((k_1,\text{Free}),(k_2,j))$, $\forall k_1,k_2 \in N, \forall j \in N \setminus k_2$ and $((k_1,\text{Free}),(k_2,\text{H}))$, $\forall k_1 \in N, \forall k_2 \in N \setminus h(k_2)$ – one ambulance is free at node k_1 and the other ambulance is traveling from node k_2 to (but not yet reach) a call location j or the closest hospital. In each of these states, the free ambulance can move to any node $k_1' \in N_{k_1}$ and the other ambulance just moves to the next node towards call location j or the closest hospital along the shortest path.

The possible transitions for each of these states after one time-step are as follows: (1) no call arrives during the wait interval, meaning that the free ambulance jumps to node k_1' and the other ambulance jumps to the next node along the shortest path, and (2) a call arrives during the wait interval, meaning that the free ambulance is dispatched to the call with reward $r(k_1)$, and the other ambulance jumps to the next node along the shortest path. Therefore Equation 4.2 gives:

$$\begin{aligned} V((k_1,\text{Free}),(k_2,j))+g = \max_{k_1'\in N_1} (&(1-e^{-\lambda})r(k_1) \\ &+ (1-e^{-\lambda})\sum_{x\in N} p(x)V((k_1,x),(\text{next}(k_2,j),j)) \\ &+ e^{-\lambda}V((k_1',\text{Free}),(\text{next}(k_2,j),j))), \quad k_2 \neq j, \end{aligned}$$

and

$$
\begin{aligned}
V((k_1, \text{Free}), (k_2, \text{H})) + g = \max_{k_1' \in N_1} &((1 - e^{-\lambda}) r(k_1) \\
&+ (1 - e^{-\lambda}) \sum_{x \in N} p(x) V((k_1, x), (\text{next}(k_2, h(k_2)), \text{H})) \\
&+ e^{-\lambda} V((k_1', \text{Free}), (\text{next}(k_2, h(k_2)), \text{H})), \quad k_2 \neq h(k_2).
\end{aligned}
$$

Next we consider states ((k_1, Free),(j,j)), $\forall k_1 \in N, \forall j \in N$ – one ambulance is free at node k_1 and the other ambulance is providing on-site treatment. The free ambulance can move to any node $k_1' \in N_{k_1}$. As we mentioned, we assume the 'wait' interval is small enough such that there is only one event that can occur during this interval.
The possible transitions for each of these states after one time-step are as follows: (1) neither a call arrives or the on-site treatment finishes during the wait interval, meaning that the free ambulance reaches node k_1' and the other ambulance is still busy on site, (2) a call arrives during the wait interval, meaning that the free ambulance is dispatched from node k_1 with reward $r(k_1$, and the other ambulance is still busy on site, (3) the on-site service finishes during the wait interval and transport is needed, meaning that the free ambulance reaches node k_1' and the other ambulance starts travelling to the closest hospital, and (4) the on-site service finishes during the wait interval and transport is not required, meaning that the free ambulance reaches node k_1' and the other ambulance becomes free at node j. Therefore, Equation 4.2 gives:

$$
\begin{aligned}
V((k_1, \text{Free}), (j, j)) + g = \max_{k_1' \in N_1} &((1 - e^{-(\lambda+\mu)}) r(k_1) \\
&+ \frac{\lambda}{\lambda + \mu} (1 - e^{-(\lambda+\mu)}) \sum_{x \in N} p(x) V((k_1, x), (j, j)) \\
&+ \frac{\mu}{\lambda + \mu} (1 - e^{-(\lambda+\mu)}) p_{\text{transport}} V((k_1', \text{Free}), (j, \text{H})) \\
&+ \frac{\mu}{\lambda + \mu} (1 - e^{-(\lambda+\mu)}) (1 - p_{\text{transport}}) V((k_1', \text{Free}), (j, \text{Free})) \\
&+ e^{-(\lambda+\mu)} V((k_1', \text{Free}), (j, j))), \quad k_2 \neq j.
\end{aligned}
$$

Next we consider states $((k_1, \text{Free}),(\text{h,H}))$, $\forall k_1 \in N$— one ambulance is free at node k_1 and the other ambulance is handing over a patient at a hospital node h. In each of these states, the free ambulance can move to any node $k_1' \in N_{k_1}$.

The possible transitions after one time-step are as follows: (1) neither a call arrives or at-hospital hand-over finishes during the wait interval, meaning that the free ambulance reaches node k_1' and the busy ambulance remains busy, (2) a call arrives during the wait interval, meaning that the free ambulance gets dispatched with reward $r(k_1)$ while the busy ambulance remains busy, and (3) the hand-over finishes during the wait interval, meaning that the free ambulance reaches node k_1' and the busy ambulance becomes free at the hospital. Therefore, Equation 4.2 gives:

$$
\begin{aligned}
V((k_1, \text{Free}), (h, \text{H})) + g = \max_{k_1' \in N_1} &((1 - e^{-(\lambda+\mu_h)})r(k_1) \\
&+ (\frac{\lambda}{\lambda + \mu_h}(1 - e^{-(\lambda+\mu_h)}) \sum_{x \in N} p(x) V((k_1, x), (h, \text{H})) \\
&+ \frac{\mu_h}{\lambda + \mu_h}(1 - e^{-(\lambda+\mu_h)}) V((k_1', \text{Free}), (h, \text{Free})) \\
&+ e^{-(\lambda+\mu)} V((k_1', \text{Free}), (h, \text{H})), \quad k_2 \neq j.
\end{aligned}
$$

Next we show the optimality equations for states where both ambulances are busy with a call respectively. For these 'busy' states, there is zero immediate reward since there are no free ambulances. Note we assume that (1) only one event, i.e. a call-arrival or a completion of on-site/at-hospital service may occur during a wait interval, and (2) when both ambulances are providing an on-site/at-hospital service, if an event of freeing up an ambulance occurs, it may happen to each ambulance with probability 0.5. This is because the two ambulances are assumed to be identical.

Consider states $((k_1, k_1),(k_2, k_2))$,$\forall k_1,\ k_2 \in N$ – both ambulances are providing on-site treatment at nodes k_1 and k_2, respectively. The possible transitions after one-time step are: (1) neither one of the two ambulances finishes at-scene treatment, meaning that they are in the same state as before; (2) one of the ambulance finishes on-site service and transport is required, meaning that one ambulance remains providing on-site treatment and the other ambulance starts travelling to the closest hospital; (3) one of the ambulance finishes on-site service and transport is not required, meaning that one ambulance remains providing on-site treatment and the other ambulance becomes free on site. Therefore, Equation 4.2 gives:

$$\begin{aligned}
V((k_1, k_1), (k_2, k_2))) + g = &\ 0.5(1 - e^{-2\mu})p_{\text{transport}} V((k_1, \text{H}), (k_2, k_2)) \\
&+ 0.5(1 - e^{-2\mu})(1 - p_{\text{transport}}) V((k_1, \text{Free}), (k_2, k_2))) \\
&+ 0.5(1 - e^{-2\mu})p_{\text{transport}} V((k_1, k_1), (k_2, \text{H})) \\
&+ 0.5(1 - e^{-2\mu})(1 - p_{\text{transport}}) V((k_1, k_1), (k_2, \text{Free}))) \\
&+ e^{-2\mu} V((k_1, k_1), (k_2, k_2))|((k_1, k_1), (k_2, k_2)).
\end{aligned}$$

Next we consider states $((h_1, \text{H}),(h_2, \text{H})),\forall h_1 \in M, h_2 \in M$ – both ambulances are handing over a patient at hospital node h_1 and h_2 respectively. Similar to the states we just discussed, the possible transitions are as follows: (1) neither one of the two ambulances finishes hand-over, (2) the ambulance at node h_1 finishes hand-over and becomes free at node h_1, and (3) the ambulance at node h_2 finishes hand-over and becomes free at node h_2. Equation 4.2 gives:

$$\begin{aligned}
V((h_1, \text{H}), (h_2, \text{H})) + g = &\ 0.5(1 - e^{-2\mu_h}) V((h_1, \text{Free}), (h_2, \text{H})) \\
&+ 0.5(1 - e^{-2\mu_h}) V((h_1, \text{H}), (h_2, \text{Free})) \\
&+ e^{-2\mu}) V((h_1, \text{H}), (h_2, \text{H})).
\end{aligned}$$

Now consider states $((i, i),(k_2, j))$, $\forall i \in N, \forall k_2 \in N, j \in N \setminus k_2$ and states $((i, i),(k_2, \text{H}))$, $\forall i \in N, \forall k_2 \in N \setminus h(k_2)$ – one ambulance is providing on-site treatment, and the other ambulance is travelling from node k_2 to (but not yet reach) call location j or the closest hospital.The possible transitions after one-time step are as follows: (1) the on-site ambulance remains busy and the other ambulance jumps to the next node along the shortest path to the call/hospital location, (2) the on-site ambulance finishes the treatment and transport is required, meaning that the on-site ambulance starts travelling to the closest hospital and the other ambulance jumps to the next node along the shortest path to the call/hospital location, and (3) the on-site ambulance finishes the treatment and transport is not required, meaning that on-site ambulance becomes free and the other ambulance jumps to the next node along the shortest path to the call/hospital location. Therefore, for a state $((i,i),(k_2,\mathrm{H}))$, Equation 4.2 gives:

$$\begin{aligned} V((i,i),(k_2,j)) + g = {} & (1-e^{-\mu})p_{\text{transport}}V((i,\mathrm{H}),(\text{next}(k_2,j),j)) \\ & + (1-e^{-\mu})(1-p_{\text{transport}})V((i,\text{Free}),(\text{next}(k_2,j),j)) \\ & + e^{-\mu}V((i,\text{Free}),(\text{next}(k_2,j),j)). \end{aligned} \tag{2}$$

Similarly, for a state $(i,i),(k_2,\mathrm{H})$, we have

$$\begin{aligned} V((i,i),(k_2,\mathrm{H})) + g = {} & (1-e^{-\mu})p_{\text{transport}}V(i,\mathrm{H}),(\text{next}(k_2,h(k_2)),\mathrm{H}) \\ & + (1-e^{-\mu})(1-p_{\text{transport}})V((i,\text{Free}),(\text{next}(k_2,j),\mathrm{H})) \\ & + e^{-\mu}V((i,\text{Free}),(\text{next}(k_2,j),\mathrm{H})) \end{aligned} \tag{3}$$

Next we consider states $((h_1,\mathrm{H}),(k_2,j))$, $\forall h_1 \in M, \forall k_2 \in N \setminus j$ and $((h_1,\mathrm{H}),(k_2,\mathrm{H}))$, $\forall h_1 \in M, \forall k_2 \in N \setminus h(k_2)$ – the first ambulance is handing over a patient and the second ambulance is travelling from node k_2 (but not yet reach) to call location j or the closest hospital. There are two possible transitions: (1) the first ambulance still remains busy and the second ambulance jumps to the next node along the shortest path, and (2) the first ambulance becomes free and the second ambulance jumps to the next node along the shortest path. Therefore, Equation 4.2 gives:

$$\begin{aligned} V((h_1,\mathrm{H}),(k_2,j)) + g = {} & e^{-\mu_h}V((h_1,\mathrm{H}),(\text{next}(j),j)) \\ & + (1-e^{-\mu_h})V((h_1,\mathrm{H}),(\text{next}(j),j)) \end{aligned}$$

and

$$
\begin{aligned}
V(((h_1, \mathrm{H}), (k_2, \mathrm{H})) + g = e^{-\mu_h} V(((h_1, \mathrm{H}), (\mathrm{next}(j), \mathrm{H})) \\
+ (1 - e^{-\mu_h}) V((h_1, \mathrm{Free}), (\mathrm{next}(j), \mathrm{H}))
\end{aligned}
$$

Lastly, we consider states $((i,i),(h,\mathrm{H}))$, $\forall i \in N, \forall h \in M$ – one ambulance is providing on-site treatment and the other ambulance is handing over a patient at a hospital. There are four possible transitions: (1) the on-site ambulance remains busy treating the patient and the at-hospital ambulance remains busy handing over the patient, (2) the on-site ambulance transports the patient from node i to the closest hospital and the at-hospital ambulance remains busy handing over the patient, (3) the on-site ambulance becomes free and the at-hospital ambulance remains busy handing over the patient, and (4) the on-site ambulance remains busy treating the patient and the at-hospital ambulance becomes free. Therefore, Equation 4.2 gives:

$$
\begin{aligned}
V((i,i),(h,\mathrm{H})) + g = e^{-(\mu+\mu_h)} V((i,i),(h,\mathrm{H})) \\
+ \frac{\mu}{\mu+\mu_h}(1 - e^{-(\mu+\mu_h)}) p_{\mathrm{transport}} V((i, h(i)), (h, \mathrm{H})) \\
+ \frac{\mu}{\mu+\mu_h}(1 - e^{-(\mu+\mu_h)})(1 - p_{\mathrm{transport}}) V((i, \mathrm{Free}), (h, \mathrm{H})) \\
+ \frac{\mu_h}{\mu+\mu_h}(1 - e^{-(\mu+\mu_h)}) V((i,i),(h, \mathrm{Free})).
\end{aligned}
$$

References

[1] St John Annual Report 2011, November 2011. URL `http://www.stjohn.org.nz/news/publications/annualreports/`. 1, 5, 73, 74

[2] R. Alanis, A. Ingolfsson, and B. Kolfal. A Markov chain model for an EMS system with repositioning, 2010. URL `http://apps.business.ualberta.ca/aingolfsson/documents/PDF/Repositioning.pdf`. 4, 13, 14, 96, 98, 111

[3] T. Andersson and P. Vaerband. Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, 28(2):195–201, 2007. 11, 18, 131

[4] R. R. Barton and J.S. Ivey Jr. Nelder-mead simplex modifications for simulation optimisation. *Management Science*, 42:954–973, 1996. 121

[5] O. Berman. Dynamic repositioning of indistinguishable service units on transportation networks. *Transportation Science*, 15(2):115–136, 1981. 9, 10, 19, 53, 54

[6] O. Berman. Repositioning of distinguishable urban service units on networks. *Computers and Operations Research*, 8(2):105–118, 1981. 9

[7] O. Berman. Repositioning of two distinguishable urban service vehicles on networks. *IEEE Transctions on Systems, Man, and Cybernetics*, SMC-11(3):187–193, 1981. 9

[8] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, 2005. 15, 23

[9] D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16:580–595, 1991. 24, 35

[10] L. Brotcorne, G. Laport, and F. Semet. Ambulance location and relocation models. *European Journal of Operational Research*, 147:451–463, 2003. 3

[11] S. Budge, A. Inglofsson, and E. Erkut. Approximating vehicle dispatch probabilities for emergency service systems with location-specific service times and multiple units per location. *Operations Research*, 57:251–255, 2009. 78

[12] G. Cady. 2001 JEMS 200 city survey, JEMS 2001 annual report on EMS operational & clinical trends in large, urban areas. *JEMS : A Journal of Emergency Medical Services*, 27(2):46–71, 2002. 4

[13] K.H. Chang. Stochastic Nelder-Mead simplex method: A new globally convergent direct search method for simulation optimisation. *European Journal of Operational Research*, 220:685–694, 2012. 121

[14] R.L. Church and C.S. ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32:101–118, 1974. 3

[15] Byatt D., I. Coope, and C. Price. 40 years of the Nelder-Mead algorithm. Technical report, University of Canterbury, New Zealand, 2003. 121

[16] M. Delasay, B. Kolfal, and Ingolfsson. Maximizing throughput in finite-source parallel queue systems. *European Journal of Operational Research*, 217(3):554–559, 2012. 9

[17] G. Erdoğan, E. Erkut, A. Ingolfsson, and Laport G. Scheduling ambulance crews for maximum coverage. *Journal of the Operational Research Society*, 61(4):543–550, 2010. 76, 78

[18] J. Fitch. Response times: Myths, measurement & management. *JEMS : A Journal of Emergency Medical Services*, 30(1):46–56, 2005. 1

[19] M. Gendreau, G. Laporte, and F. Semet. Solving an ambulance location model by tabu search. *Location Science*, 5:75–88, 1997. 10

[20] M. Gendreau, G. Laporte, and F. Semet. A dynamic model and parallel tabu search heuristic for real time ambulance relocation. *Parallel Computing*, 27(12):1641–1653, 2001. 4, 10, 11, 18, 131

[21] M. Gendreau, G. Laporte, and F. Semet. The maximal expected coverage relocation problem for emergency vehicles. *Journal of the Operational Research Society*, 57(1): 22–28, 2006. 13, 95, 97

[22] R.A. Howard. *Dynamic Programming and Markov Processes*. M.I.T Press, Cambridge, Mass, 1960. 35, 41

[23] J.P. Jarvis. Approximating the equilibrium behavior of multi-server loss systems. *Management Science*, 31:235–239, 1985. 78

[24] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9:112–147, 1998. 122, 123

[25] R.C. Larson. A hypercube queueing model for facility location and redistricting in urban emergency services. *Computers and Operations Research*, 1:67–75, 1974. 78

[26] R.C. Larson. Approximating the performance of urban emergency service systems. *Operations Research*, 23:845–868, 1975. 78

[27] A.M. Law and W.D. Kelton. *SIMULATION MODELING and ANALYSIS - Third Edition*. McGRAW-HILL, 2000. 83

[28] R.M. Lewis, V. Torczon, and M.W. Trosset. Direct search methods: then and now. *Numerical Analysis*, 124:191–207, 2000. 121

[29] C.S. Lim, R. Mamat, and T. Braunl. Impact of ambulance dispatch policies on performance of emergency medical services. *IEEE Transations on Intelligent Transportation System*, 12:624–632, 2011. 74

[30] Restrepo M. *Computational methods for static allocation and realtime redeployment of ambulances*. PhD thesis, Cornell University, 2008. 14, 16, 95

[31] S. Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22(1-3):159–195, 1996. 40

[32] A. Mason. Simulation and real-time optimized relocation for improving ambulance operations. In Brian Denton, editor, *Healthcare Operations Management: A Handbook of Methods and Applications.* Springer. 71

[33] A. Mason and S. Henderson. Bartsim: A tool for analysing and improving ambulance performance in auckland, new zealand,. In *Proceedings of the The 35th Annual Conference of the Operational Research Society of New Zealand*, 2000. 71, 73

[34] S.M. Maxwell, S.G. Henderson, and H. Topaloglu. *APPROXIMATE DYNAMIC PROGRAMMING POLICIES AND PERFORMANCE BOUNDS FOR AMBULANCE REDEPLOYMENT*. PhD thesis, Cornell Univerisity, 2011. 15, 16, 17, 77, 78, 95, 96, 98, 102, 117, 121, 131, 132, 158, 164

[35] R. Nair and E. Miller-Hooks. Evaluation of relocation strategies for emergency medical service vehicles. *Transportation Research Record: Journal of the Transportation Research Board (2137)*, pages 63–73, 2009. 116

[36] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. 121

[37] M.J.D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–592, 2002. 121

[38] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* John Wiley & Sons, Hoboken, NJ., 2007. 15

[39] D.P. Richards. Optimised ambulance redeployment strategies. Master's thesis, Department of Engineering Science, University of Auckland, Auckland, New Zealand, 2007. 5, 11, 18, 72, 131, 151

[40] J. Savage. Ambulance slower to answer calls. NZ Herald, January, 16, 2012. 5

[41] R. Sutton and A. Barto. *Reinforcement Learning*. The MIT Presss, Cambridge, Massachusetts., 1998. 15

[42] A. J. Swersey. *The deployment of police, fire and emergency medical units.*, volume 6, pages 151–190. Operations Research and the Public Sector, Handbooks in Operations Research and Management Science, North-Holland, 1994. 3

[43] C. Toregas, R. Swain, C. Revelle, and L. Bergmann. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971. 3

[44] Schmid V. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, 219: 611–621, 2012. 16, 17

[45] D.J. White. *MARKOV DECISION PROCESSES*. JOHN WILEY & SONS, 1993. 38, 40, 46, 57

[46] D.M. Williams. Jems 2008 200 city survey: The future is your choice. *JEMS : A Journal of Emergency Medical Services 3*, 34(2):36–51, 2009. 4

[47] Statistics New Zealand. Digital boundary files. URL http://www.stats.govt.nz/browse_for_stats/people_and_communities/Geographic-areas/digital-boundary-files.aspx. 76