The Optimisation of a Single Ambulance Moveup

Lei Zhang, Andrew Mason, Andy Philpott

Report, University of Auckland, Faculty of Engineering. No. 682

ISSN 1179-528x

The Optimisation of A Single Ambulance Move Up

Lei Zhang, Andrew Mason, Andy Philpott
Department of Engineering Science
University of Auckland
New Zealand

Abstract

This paper develops optimal or near-optimal redeployment policies for single-ambulance problems. The first model aims to decide where to move the single ambulance on a network so as to maximize the reward for the next call. A dynamic programming model is formulated. Mathematical properties of optimal solutions are discussed and an efficient solution technique is presented. The second model considers where to move the single ambulance in order to maximize an expected number long run performance measure. To deal with the high-dimensional state space in this model, we formulate a new dynamic programming model with reduced state space. Examples are given to show insights.

1 Introduction

Facility location plays a vital role in strategic planning for a broad range of problems such as deciding where to put bus stops, where to construct new libraries and where to allocate health care facilities. Static location models give a single one-off solution such as might be required when locating a building. Dynamic relocation models attempt to determine how to relocate facilities to best react to changes in the environment such as demand or traffic congestion. The relocation of facilities is not possible for many problems. However, if the facility can be relocated for a reasonable cost and within a reasonable time, then dynamic relocation can be beneficial. In particular, emergency services such as police, fire, repair and emergency medical services (EMS) have attracted much interest from operations research (see [1, 2]) because of the mobility of their servers.

This paper considers relocating ambulances in order to better respond to emergency calls. However, we expect that the research presented here is useful for other problems involving mobile servers. Examples include taxi services and towing services where a driver must decide where to wait for future customers in order to maximize the profits or minimize the time to get to a breakdown. Relocation can also be applied to elevator group control to minimize the average waiting time for each passenger [3].

The ambulance problem we consider is characterised by the response process summarised in Figure 1. When an emergency call is received, a dispatcher chooses an available ambulance to dispatch. A typical dispatch policy will look at those vehicles waiting idle at a base or returning to their base and find the vehicle that is closest to the accident scene. (The time required for this dispatching process is typically small, and so will be ignored here.) This dispatched ambulance travels to the scene of the emergency call. Once the ambulance reaches the scene, the ambulance officers 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 returns to its base. More typically, however, transportation is required to a hospital and the ambulance then becomes free at the hospital after completing a patient hand-over. This response process can be complicated by the presence of calls of different priorities and vehicles with different capabilities, the need to dispatch multiple vehicles 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. However, we ignore these complexities at this stage.

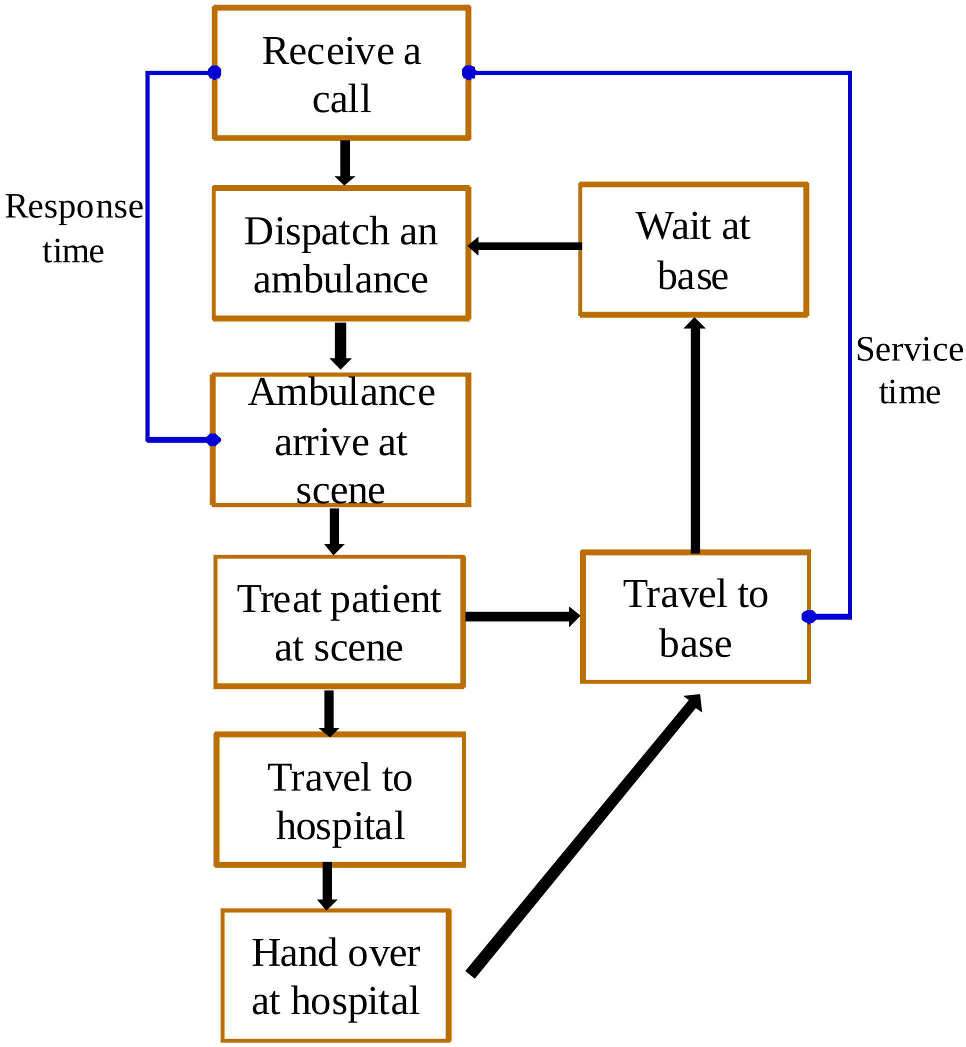

Figure 1 – A typical response process for an emergency call

The elapsed time between the receipt of the call and the vehicle arriving at the scene is termed the *response time*. An ambulance organisation's performance will often be measured by the percentage of calls having a response time no greater than some target time W. When trying to maximise their performance, ambulance operators typically refer to their readiness to respond to the next call in terms of *coverage*, where a suburb is considered covered if its centroid is no further than W minutes drive from the closest available ambulance. They also refer to *call coverage* which is the probability that the location of the next call is no further than W minutes drive from the closest available ambulance.

In many ambulance organisations, each ambulance is assigned a base to which the vehicle returns after each call; determining the best base for each vehicle gives us a static location problem. These problems are typically solved used using Integer Programming (IP) models which seek a vehicle-to-base assignment that maximises some simple coverage-based model of expected system performance. In these IP models, the region of interest is partitioned into suburbs or 'zones' each of which has an associated weight giving the expected number of calls in that zone. Early models such as set covering location problem (SCLP) by Torgas et al. [4] and the maximal covering location problem (MCLP) by Church and ReVelle [5] maximise the weighted sum of zones covered by one or more vehicles. This objective implicitly assumes each vehicle will always be available at its base, and that no extra benefit accrues from covering a zone by multiple vehicles. To help correct for these simplifying assumptions, models which consider the benefits of multiple coverage for a zone have been proposed; see Hogan and ReVelle [6], Gendreau et al. [7] and Andersson et al. [8].

Ambulance vehicles are often busy for a high proportion of their time. To explicitly model this, Daskin [9] extended earlier integer programming models to form the maximal expected covering location model (MCLP) which assumes a system-wide busy probability for each vehicle. A natural enhancement of this approach is to allow the busy probabilities to be station specific. Budge [10] developed such a model that incorporates the hypercube approximation originally developed by Larson [11, 12]. The Budge model also incorporated randomness in travel times and pre-delays (time elapsed until a vehicle is dispatched). Although the model's objective is to find the minimum number and locations of ambulances needed for a given system-wide coverage (average coverage of all zones), with slight changes, it can also be used to maximize the expected coverage for a fixed number of ambulances. These models are still limited in their ability to model complex real-world ambulance behavior, and so simulation is often used to better predict the system performance, e.g. see [13]

In an attempt to improve their response times, some ambulance operators operate a *redeployment* policy in which they move idle ambulances from one base to another, or even to street corners, as they seek to improve their call coverage. This vehicle movement is an example of a *move-up*. A common redeployment approach is System Status Management (SSM) which, for any given number of free vehicles n_{free}, specifies a pre-defined vehicle *configuration* $C(n_{\text{free}})$ that gives a standby location (i.e. a base or a street corner) for each of the idle vehicles; e.g. see [14], [15]. Whenever the number of free vehicles changes, the dispatchers are required to determine a set of moveups that efficiently move vehicles into the appropriate SSM configuration.

Gendreau et al. [16] propose an integer programming model that can be used to create System Status Management plans. Their objective is to maximize the expected coverage weighted by the probability $p(n_{\text{free}})$ of having n_{free} idle vehicles. They restrict the number of different waiting sites that can be used when going from configuration $C(n_{\text{free}})$ with n_{free} idle vehicles to an 'adjacent' configuration $C(n_{\text{free}}+1)$ with $n_{\text{free}}+1$ idle vehicles. This gives configurations that are similar in that adjacent configurations share common standby locations. Simulations were performed to show performance improvements over the 'return-to-base' policy in which vehicles always return to their original bases. They showed that applying a relocation policy is better than using a static approach when considering system performance such as the average response time and the percentage of calls covered within 8 minutes. The computation time varied from one minute for 3 vehicles to slightly more than three hours for 6 vehicles.

An alternative approach, which is the focus of our work, is to adopt dynamic relocation models to determine optimal or near optimal moveups for the available ambulances. Unlike System Status Management plans, these solutions do not enforce a single configuration $C(n_{\text{free}})$ for each $n_{\text{free}} = 1, 2, 3, ...$, but instead allow the target configuration to depend on the current vehicle locations.

A natural approach when considering relocation is to extend the integer programming coverage models used for the static vehicle location problems. These extended models take as input some current vehicle configuration (i.e. set of locations) and produce a set of moveups for the vehicles. They maximise an objective formed by some coverage-based performance measure of the final vehicle configuration less some cost function of the travel required to achieve that configuration. For example, Gendreau, Laporte, and Semet [2] propose the dynamic double standard model (DDSM) which chooses a set of vehicle moves that maximizes the total demand-weighted *backup coverage* for all zones less the move up costs, where a zone has backup coverage if it is covered by at least two ambulances. The model is solved whenever vehicles become busy or free. The formulation is 'single-shot' in that it considers only the next set of moveups. However, recent vehicle history is used to impose constraints such as avoiding round trips, avoiding repeated movements of the same ambulance, etc. A parallel tabu search heuristic algorithm is developed to speed up the solution process. This tabu search continuously computes the best possible standby configuration for every possible future scenario in which one of the idle ambulances is dispatched to the next call. No results are given on the performance improvements generated by this relocation approach.

Richards [17] proposed a similar model with a different objective for calculating the benefits from move up. Each additional ambulance, up to a target number for a zone, contributes to a concave increasing performance function. Moreover, under the assumption of perfect information, the current busy ambulances which are likely to be available at some 'look-ahead' time also contribute coverage to the particular zone in which they will become free. Andersson et al. [18] proposed a dynamic ambulance relocation model based on a heuristic static model. They propose a score function which is then used in constraints that ensure each zone achieves some minimum score after relocation. The objective is to minimize the maximum travel time of the vehicles being relocated.

The integer programming models all use some linearised approximation of system performance. The approximations are limited in that they must approximate (or typically ignore) vehicle busy probabilities, they assume all vehicles are idle at a standby location when they are dispatched, and more importantly, they do not incorporate the benefits and costs associated with future moveups. Dynamic Programming approaches can avoid these limitations. Berman [1, 19, 20] developed a Dynamic Programming model to explicitly capture the impact on the long-term performance due to moveups. The objective is to minimize the average response time in the long-run. A set of bases is defined on the network, where each base can have up to one vehicle waiting at it. Move-ups can be used to move vehicles between bases; no other waiting locations are permitted for the vehicles. The system state is defined as the number of idle vehicles at each base. Vehicles provide at-scene service (but no hospital transport) and become free at the nearest vacant base location after an exponentially distributed service time. Moveup of up to one vehicle is allowed to occur at any instant that the system undergoes a service-oriented transition (being an initiation or completion of service, but not the completion of a move-up). It is assumed that during move-up, only one event (a new call or a completion of service) is possible. Therefore given a state and a move-up decision, possible service-oriented transitions include those to a state in which: (1) a vehicle becomes free at a vacant base and the move-up vehicle is at its new base; (2) one of the stationary vehicles becomes busy during or after move-up and the move-up vehicle is at its new base; or (3) the move-up vehicle becomes busy during or after move-up. The immediate cost of the move-up action is the expected response time for the next call which may arrive during or after the move-up. A numerical example is given by Berman to give some insights. For low vehicle utilization factors and/or for small travel times along the move-up paths, a vehicle tends to move from a 'weak' base (one giving giving a poor expected response time) to a 'strong' base (with better response time). When the utilization factor increases, a vehicle may continually move between bases if the call arrival rate on the path between the bases is greater than that at either base. If the utilization factor is very high, then move-up occurs very rarely if the call arrival rate is low along the moveup path.

The state space of a dynamic program becomes intractable for realistically-sized problems. Henderson et al. [21] have recently proposed an approximate Dynamic Programming (ADP) approach for making real-time ambulance relocation decisions. The objective of their model is to minimize the discounted number of urgent calls that are not responded to on time. They use an approximate value function that eliminates the need to store a function value for every state and thus can model a greater complexity of ambulance operations such as an ambulance going to the scene, treating at the scene and transporting to a hospital. A weighted combination of six basis functions is used to approximate the value function. The weights used in this function are tuned using simulation-based policy iterations. The CPU time for each iteration of this policy iteration algorithm was 22 minutes. Their ADP model was compared with two benchmark strategies. The first benchmark was a greedy policy that resulted from setting the parameters of their approximate value functions to zero. The second benchmark was the best 'return-to-base' policy found through a simulation-based enumerative search. Two data sets were tested using simulation, with the ADP approach outperforming both benchmark strategies. The best performance improvements for the first data set were 4.7% and 4.0%, respectively, in the percentage of calls reached on time. The best improvements in performance for the second data set were 2.4% and 2.0%, respectively. The CPU time for generating the optimal decision for real-time operation was about 45 milliseconds.

The main contribution of this paper is to develop optimal or near-optimal redeployment policies for single-ambulance problems. Our models are both more realistic than the Berman model [1] and more accurate than the integer programming models detailed above. They provide valuable insights into the form and benefits of optimal move up that can help direct the development of the approximate models needed for larger problems. The reminder of this paper is organized as follows. In Section 2, we study the move up of a single ambulance on a network when considering just the next call. A Dynamic Programming model is developed, and worked examples are used to provide insights. An efficient solution technique is presented for this problem. In Section 3, we extend our model to maximise an expected long run performance measure. This is an extension to the model proposed by Berman [1]. In Berman's model, a vehicle becomes free at the nearest vacant base. In our Dynamic Programming model, a vehicle can become free at any site on the network. State aggregation is then used in order to solve large problems using a modified form of Policy Iteration [22]. Finally Section 4 contains discussions of the results of the models and the insights they provide for models with more ambulances.

2 The Single-Ambulance Next-Call Model

Ambulance performance is typically measured and reported over periods of weeks or months, and so any move-up policy should perhaps maximise an expected long run performance measure. However, in some situations it may be acceptable to focus on the performance in the very near future, perhaps over just the next one or two calls. For example, in a system with low vehicle utilization, vehicles typically have time to return to their standby locations before becoming busy, and so focusing on just the next call may give a near optimal policy. In this section, we consider the move-up problem for a single ambulance where we maximise performance for just the single next call. We assume all patients are treated at the scene, and so we ignore any hospital effects.

2.1 The Single-Ambulance Next-Call Formulation

Consider a single ambulance operating on a network G consisting of a set of nodes N and a set of undirected links L, where $(i, j) \in L$ is the undirected link joining nodes i and j, $i, j \in N$. We define $N_k = \{j : (j, k) \in L\}$ to be the set containing those nodes adjacent to node k. The spacing of the nodes is such that each link requires a constant drive time Δt to traverse where Δt is small enough that the probability of more than one call occurring during Δt is insignificant. Call arrivals follow a Poisson process with a total arrival rate λ. The probability that the next call occurs at node k is p_k with $\sum_{k \in N} p_k = 1$.

Ambulance responses are given a reward of 1 if the ambulance can reach the call within the specified target time W and 0 otherwise. The expected reward r_k for the next call when the ambulance is waiting at node k is given by:

$$r_k = \sum_{j \in N : d_{k,j} \leq W} p_j \quad \forall k \in N$$

where $d_{k,j}$ is the travel time along the shortest path from node k to node j, and W is the target response time. (We assume d_{kj} and W are both expressed as multiples of the time step δt, and so d_{kj} is also the number of nodes along the shortest path.)

We aim to find a move up policy that determines where to move the ambulance from any initial node in order to maximize the expected reward for the next call. To model ambulance movement, we use a 'wait-and-jump' discretisation in which the ambulance waits for time Δt at its current node k, then moves instantaneously to an adjacent node $k' \in N_k$. If a call occurs during this waiting time, the ambulance is dispatched from node k giving expected reward r_k; otherwise the moving ambulance jumps to node k'.

In order to find the maximum expected reward V_k for an ambulance initially at node k, we formulate a Dynamic Programming model. The formulation requires the following additional notation:

V_k = maximum expected reward for the next call under an optimal move up policy when the vehicle is initially idle at node k

$\pi(k)$ = move up policy that specifies the successor node when at node k; $\pi(k) \in N_k \cup \{k\}$.

For the problem as defined, the optimality equation can be written as follows:

$$V_k = (1 - e^{-\lambda \Delta t}) r_k + \max_{k' \in N_k \cup \{k\}} e^{-\lambda \Delta t} V_{k'}. \tag{1}$$

This can be interpreted as follows. The ambulance waits for time Δt at its current location k. If the next call occurs during this interval, the ambulance is dispatched giving an expected reward r_k. If no call arrives, the ambulance jumps to node k', where k' is an adjacent node ($k' \in N_k$) or k itself ($k' = k$). Note that to ensure a policy is unique, we break ties in (1) by assuming the vehicle 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 [23] if we reformulate the problem with the objective of minimising the probability of not getting to the next call on time. The 'cost' for going to a neighboring node at node k is 1-r_k in this case. The termination node starting from any node k in this shortest path problem is the 'absorbing node' at which the next call occurs.

2.2 Worked Examples and Insights

We use two examples to gain insights into the benefits of move up under this model. The first example considers moveup on a single road. The second example considers a general network.

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

The horizontal axis in Figure 2 represents a network consisting of 30 nodes located along a line at one minute spacings. Using a target response time of $W = 4$ minutes, each node k has an expected reward of r_k as shown on the vertical axis. Solving (1) with $\lambda = 0.5$ call per hour gives an optimal policy in which the idle vehicle always travels to and waits at node 20. The expected reward V_k under this policy is shown on the plot.

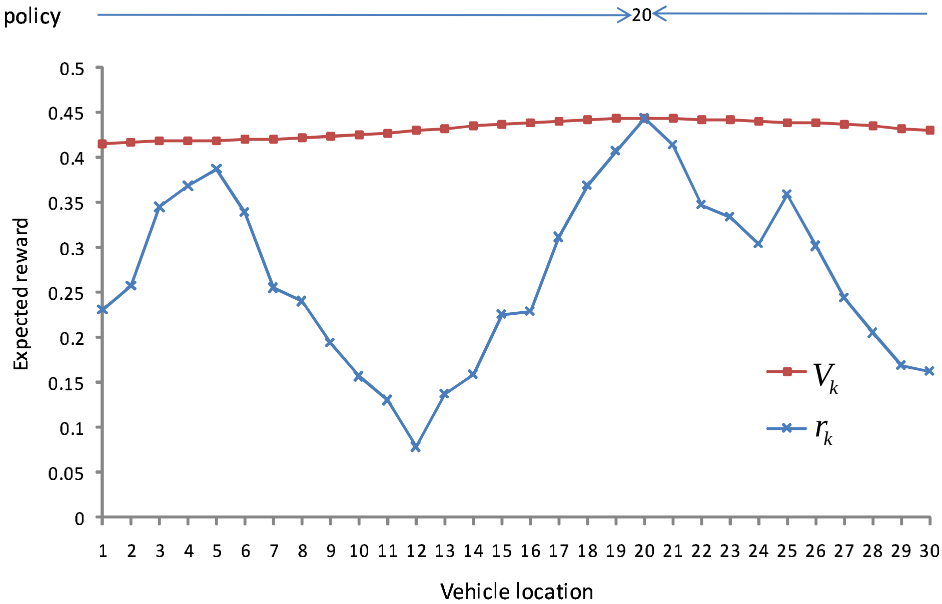

Figure 2 – Plot of reward function r_k, value function V_k and the optimal move-up policy for Example 1 with $\lambda = 0.5$ calls per hour. There is one optimal standby location, node 20. Note that r_k gives the expected reward without move-up, while V_k gives the maximum expected reward under the moveup policy.

Figure 3 shows the optimal policy and associated value function when the call arrival rate increases to $\lambda = 3$ calls per hour. We can see that a new move-up policy is formed in which there are two standby locations, nodes 5 and 20. If the ambulance is initially located between node 1 and node 10, it moves to standby location 5; otherwise the ambulance moves to standby location 20.

We can see that different optimal move up policies arise from the two call arrival rates. With $\lambda = 3$, 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 standby location (node 5) is optimal. This occurs because, for the higher call arrival rate, there is too great a chance of the next call occurring during the move-up while the vehicle is at a location with a low expected reward r_k.

Figures 2 and 3 suggest the following propositions:

Proposition 1 *For an optimal policy,*

(i) $\pi(k) \neq k \Rightarrow V_{\pi(k)} > V_k > r_k \quad \forall\ k \in N.$

(ii) $\pi(k) = k \Rightarrow V_k = r_k \quad \forall\ k \in N.$

Proof: From (1) we have the optimality equation

$$V_k = (1 - e^{-\lambda \Delta t})r_k + e^{-\lambda \Delta t} V_{\pi(k)}. \tag{2}$$

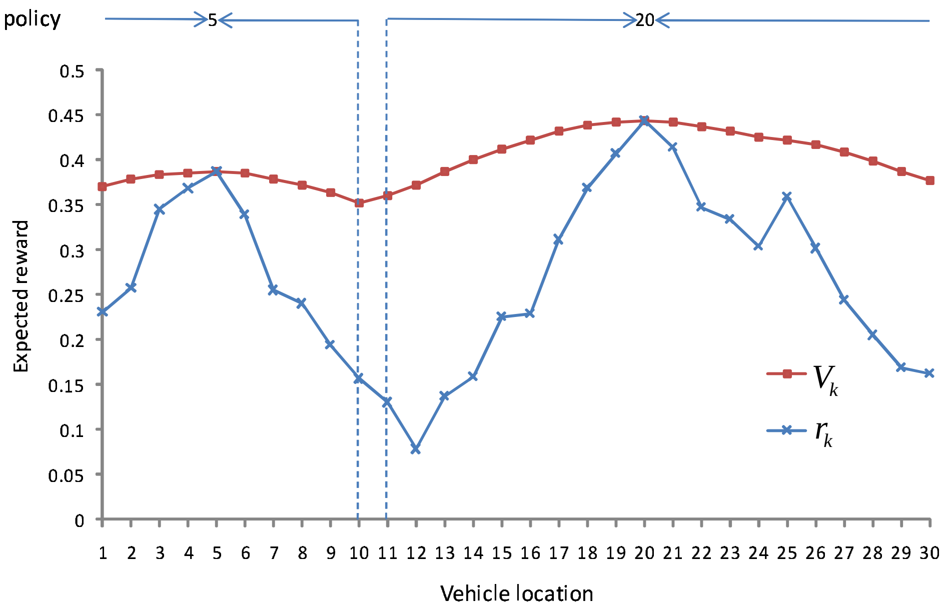

Figure 3 – Plot of reward function r_k, value function V_k and the optimal move-up policy for Example 1 with $\lambda = 3$ calls per hour. There are two optimal standby locations, nodes 5 and 20.

To prove (i), we simply note that in (2), V_k is a convex combination of r_k and $V_{\pi(k)}$. To obtain the strict inequality we require both $0 < e^{-\lambda \Delta t} < 1$ and $V_{\pi(k)} > r_k$. The former is true for any finite non-zero λ. If the latter were not true, we would have $V_k \leq r_k$, which violates our assumption of moving only if this gives a strict improvement. The proof of (ii) follows by putting $\pi(k) = k$ in (2).

Proposition 2 *Under an optimal policy, the expected reward r_k at a standby 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 $(1 - e^{-\lambda \Delta t})r_k + e^{-\lambda \Delta t} r_{k'} > r_k$. Hence we have a contradiction.

Proposition 3 *Consider now a possible moveup in which a vehicle 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 moveup path, and let $J_{c,n},\ c = 0, 1, ..., n-1$ denote the expected reward when a vehicle following this moveup is at node c. The following conditions are necessary for this moveup to occur in an optimal policy.*

(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 moveup 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(i). 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 4 *For any moveup path satisfying the necessary conditions in Proposition 3,* $J_{c,n},\ c = 0, 1, ..., n-1$ *is a non-increasing function of* λ*.*

Proof. We proceed by induction. Note that, by definition, we have

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

We first show $\frac{dJ_{n-1,n}}{d\lambda} < 0$. Put $c = n-1$ and differentiating $J_{c,n}$ gives

$$\frac{dJ_{n-1,n}}{d\lambda} = \Delta t\ e^{-\lambda \Delta t}(r_{n-1} - r_n)$$

Using Proposition 3(ii), we have $r_n > r_{n-1}$, therefore $\frac{dJ_{n-1,n}}{d\lambda} < 0$.

We now show if $\frac{dJ_{c+1,n}}{d\lambda} < 0$ for all $c+1 : 1 \leq c+1 \leq n-1$, then we have $\frac{dJ_{c,n}}{d\lambda} < 0$. Differentiating $J_{c,n}$ with $\frac{dJ_{c+1,n}}{d\lambda} < 0$ and using $r_c < J_{c+1,n}$ (Proposition 3(i)) gives

$$\frac{dJ_{c,n}}{d\lambda} = \Delta t e^{-\lambda \Delta t}(r_c - J_{c+1,n}) + e^{-\lambda \Delta t} \frac{dJ_{c+1,n}}{d\lambda} < 0$$

Thus $\frac{dJ_{0,n}}{d\lambda} < 0$ follows by induction.

Proposition 5 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 set of destination nodes for all paths $C(k, \lambda)$ starting from node k which satisfy the necessary conditions in Proposition 3. Modifying our notation to explictly show the dependence on λ, we note that Proposition 4 shows $\lambda_1 < \lambda_2 \Rightarrow J_{k,n}(\lambda_1) \geq J_{k,n}(\lambda_2)$ 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 $\max_{n \in D(k,\lambda_1)} J_{k,n}(\lambda_1) \geq \max_{n \in D(k,\lambda_2)} J_{k,n}(\lambda_2)$. 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 3. Only the first condition, $J_{c+1,n} > r_c$, depends on λ. Proposition 4 shows that $J_{c+1,n}$ is non-increasing in λ. Given that r_c is constant, our result follows immediately. Thus the proof is complete.

Proposition 6 *Let* $\pi^\lambda(k)$ *be an optimal policy for a call arrival rate* λ*. An optimal standby location* $k : k = \pi^{\lambda_1}(k)$ *for arrival rate* λ_1 *is also an optimal standby 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. This proposition follows by considering the term $\max_{k' \in N_k \cup \{k\}} e^{-\lambda \Delta t} V_{k'}$ in the optimality equation 1. 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 5). Thus the optimal decision will not change, and so the result follows.

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 1 – The expected reward r_k at each node for the next call for Example 2.

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

We now apply the model to an undirected network of 35 nodes with a call arrival rate $\lambda = 6$ calls per hour. The target response time is assumed to be 2 minutes. The probability p_k that the next call occurs at node k is given in Appendix A. The expected reward r_k at each node for the next call is shown in Table 1.

Figure 4 illustrates the optimal move-up policy. The two solid circles at nodes 24 and 33 represent two optimal standby locations. We make the following observations.

Proposition 7 *The move-up policy divides the network into separate trees with each optimal stand-by node forming the root of a tree.*

Proposition 8 *An optimal move up path may not be a shortest path.*

For example, for an ambulance at node 13, the shortest path 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 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.

2.3 Solution Techniques

We can use the standard value iteration approach [8] to solve our Dynamic Programming model. Alternatively, Algorithm 1 presents a faster solution technique which solves the model in linear time. This algorithm finds the objective value node by node in descending order of V_k. The validity of this solution technique is proven in Appendix A.

We tested the standard value iteration and our improved algorithm on a 20000-node problem in which the probability of the next call occurring at each node and the edges on the network were randomly generated. The value iteration took 0.67 seconds to find the optimal policy while our algorithm took 0.11 seconds.

3 The Single-Ambulance Infinite-Horizon model

The Single Ambulance Next Call model discussed in the previous section focused on the response time for just the next call. In this section, we consider instead the long run benefit of move up

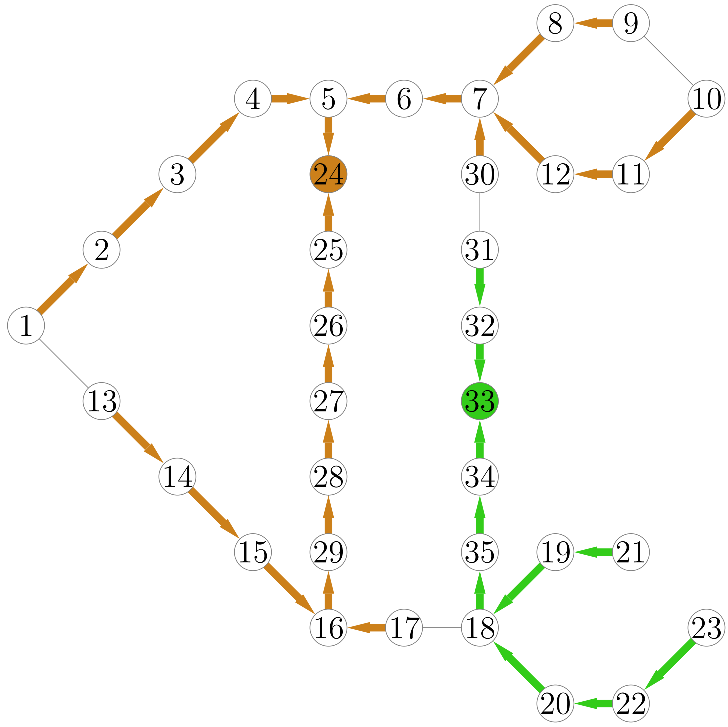

Figure 4 – Single vehicle, next call move-up solution on a network. Nodes 24 and 33, shown as solid circles, represent optimal standby locations. The optimal move-up path from any node is shown by the arrows.

when a single ambulance responds to an infinite sequence of calls. Clearly this problem falls into the category of infinite horizon problems. This section is organized as follows. Section 3.1 presents assumptions for this problem. Section 3.2 describes the state space and control. Section 3.3 presents a Dynamic Programming model and the set of linear equations used to solve the problem using Policy Iteration. Then in Section 3.4, state aggregation is used to allow larger problems to be solved. Examples and insights are presented.

3.1 Problem Assumptions

The call arrival and vehicle travel models are the same as those used for the next-call model. The at-scene service time is assumed to follow a negative exponential distribution with mean rate u_p^{-1} per hour. We assume that following the at-scene treatment, 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 the hospital is also assumed to follow a negative exponential distribution with rate u_h. Finally we assume there is no queuing, and so calls that arrive while the ambulance is busy are lost to the system.

3.2 State Space and Control

Consider now the states required in our model. These states, as illustrated in Figure 5, track the steps in the typical response process described earlier. State (k, Free) indicates that the ambulance is idle at node k. This is a decision state in that we must determine if the ambulance stays at node k or moves to an adjacent node $k' \in N_k$. In either case, the ambulance is

Algorithm 1 An improved algorithm for the single-vehicle next-call 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 $T = N$.

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 neighbour node $h \in T \cap N_k$ and update V_h:

$$V_h \leftarrow \max(V_h, (1 - e^{-\lambda \Delta t})r_h + e^{-\lambda \Delta t} V_u)$$

Until every node is permanently labeled

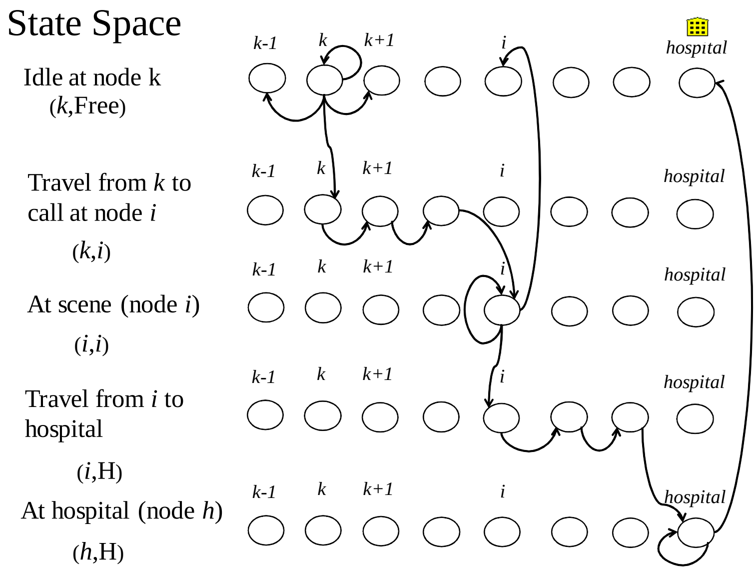

Figure 5 – Example of the State Space for the Single-Ambulance Infinite-Horizon problem.

considered to be at k for time Δt, during which time a call may arrive with probability $1-e^{-\lambda \Delta t}$; this call generates an expected reward r_k. If a call arrives, then it can occur at any node i with probability p_i. When a vehicle is dispatched to the call, the system enters state (k, i) meaning the vehicle is at node k traveling on a shortest path to a call at location i. After each Δt time step, the ambulance moves to the next node along this path. 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 time step Δt, the system will still be in state (i, i) with probability $e^{-u_p \Delta t}$, or the treatment will have been completed. If the treatment completes, the system transitions either to state (i, Free), indicating the ambulance is now free at the scene, or to state (i, H). The state (k, H) indicates that the ambulance is currently at node k transporting a patient to the closest hospital. Assuming the closest hospital is at node $h = h(k)$, then at each time step Δt, the ambulance takes one step along the shortest path to node h. When the ambulance arrives at the hospital, we enter a new state (h, H) indicating that the ambulance is at node h handing over the patient. Based on the assumption of an exponential service time at the hospital, after time Δt, the system may still be in state

(h, H) with probability $e^{-u_h \Delta t}$, or have transitioned to the free state (h, Free).

3.3 Single-Ambulance Infinite-Horizon Formulation

For an infinite horizon problem, two common objectives are to maximize the discounted total reward and to maximize the average reward in the long run. Ambulance organisations report their performance over all calls, and so we use an un-discounted objective. In general, the average reward $g(s)$ [24] under a policy π when starting in some state s can be written as:

$$g^\pi(s) = \lim_{T\to\infty} \frac{E(\sum_{t=0}^{t=T-1} r_t^\pi(s))}{T},$$

where T represents the infinite time horizon and $r_t^\pi(s)$ is the reward at time step t when starting in state s under the policy π. For our problem, the whole state space forms a single recurrent class, and so the average reward is independent of the starting state [24], giving $g^\pi = g^\pi(s)\ \forall s \in S$ where S is the set of all states.

For our problem, the average reward g is the average number of calls reached on time per time step. The problem of maximising g can be formulated as a Dynamic Programming model. The standard average reward Bellman equation [24] is

$$v_s + g = \max_{a\in A_s}(r_{s,a} + \sum_{y\in S} p^a_{sy} v_y) \quad \forall s \in S \tag{3}$$

where v_s is the relative value at state s by setting the value of a reference state $v_{s_1} = 0$, S is the set of all states, A_s is the set of feasible actions at state s, $r_{s,a}$ is the immediate reward from choosing action a in state s and p^a_{sy} is the transition probability from state s to state y given action a.

For our problem, we have a state space $S = \{(k, \mathrm{Free}), (k, i), (k, \mathrm{H}) : i, k \in N\}$. A set of hospitals H are located at nodes of the network G. The move-up decisions we have to determine are given by $\pi(k)$, $k \in \{(k, \mathrm{Free}) : k \in N\}$. To simplify our notation, we shall simply use $\pi(k)$, $k \in N\}$ to denote the neighbouring node we move to from node k under our moveup policy. The non-zero transition probabilities associated with these free states are:

$$P\{(\pi(k), \mathrm{Free})|(k, \mathrm{Free})\} = e^{-\lambda\Delta t},$$
$$P\{(\pi(k), i)|(k, \mathrm{Free})\} = (1 - e^{-\lambda\Delta t})p_i \quad i \in N.$$

where $e^{-\lambda\Delta t}$ is the probability of no call arriving during time interval Δt. The expected immediate reward when in state (k, Free) is

$$(1 - e^{-\lambda\Delta t})r_k$$

which is the expected reward at node k multiplied by the probability $1 - e^{-\lambda\Delta t}$ of a call occurring during Δt. Therefore (3) gives:

$$v_{k,\mathrm{Free}} + g = (1 - e^{-\lambda\Delta t})r_k + (1 - e^{-\lambda\Delta t})\sum_{i\in N} p_i v_{k,i} + \max_{k'\in N'} e^{-\lambda\Delta t} v_{k',\mathrm{Free}}, \quad \forall k \in N \tag{4}$$

Consider next states $(k, i), k \neq i$ and $(k, \mathrm{H}), k \neq h(k)$ in which the ambulance is traveling from node k to i to serve a call, or traveling to (but 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 reward for either of these states. Thus (3) gives

$$v_{k,i} + g = v_{\mathrm{next}(k,i),i} \quad k \neq i \tag{5}$$

$$v_{k,\mathrm{H}} + g = v_{\mathrm{next}(k,h(k)),\mathrm{H}} \quad k \neq h(k) \tag{6}$$

where $\mathrm{next}(k, j)$ represents the successor of node k along the shortest path to j.

State (i, i), in which the ambulance is treating at the scene, has the following non-zero transition probabilities:

$$P\{(i,i)|(i,i)\} = e^{-u_p \Delta t}$$
$$P\{((i,\mathrm{Free})|(i,i)\} = (1 - e^{-u_p \Delta t})((1 - p_{\mathrm{transport}})$$
$$P\{((i,\mathrm{H})|(i,i)\} = (1 - e^{-u_p \Delta t})p_{\mathrm{transport}}$$

There is no immediate reward for this state and so (3) gives

$$v_{i,i} + g = e^{-u_p \Delta t} v_{i,i} + (1 - e^{-u_p \Delta t})((1 - p_{\mathrm{transport}})v_{i,Free} + p_{\mathrm{transport}} v_{i,\mathrm{H}}) \tag{7}$$

The last state (h, H), in which the ambulance is at the hospital node h handing over the patient, has the following associated non-zero transition probabilities:

$$P\{(h,\mathrm{H})|(h,\mathrm{H})\} = e^{-u_h \Delta t}, \ h \forall H$$
$$P\{((h,\mathrm{Free})|(h,\mathrm{H})\} = 1 - e^{-u_h \Delta t}, \ h \forall H$$

There is no immediate reward for this state and so (3) gives:

$$v_{h,\mathrm{H}} + g = e^{-u_h \Delta t} v_{h,\mathrm{H}} + (1 - e^{-u_h \Delta t}) v_{h,\mathrm{Free}} \tag{8}$$

To find the optimal policy, we use these average reward Bellman equations in relative value iteration proposed by White [25]. Let s_1 denote an arbitrary reference state. Let $T(v)(s)$ denote the mapping obtained by applying the right hand side of Bellman's equation:

$$T(v)(s) = \max_{a \in A_s} (r_{x,a} + \sum_{yinS} p^a_{sy} v_y).$$

We also define the span of a vector v as follows

$$sp(v) = \max_{s \in S} v_s - \min_{s \in S} v_s.$$

The relative value iteration algorithm is listed in Algorithm 2.

Algorithm 2 The relative value iteration

1 Initilize $v^0_s = 0$ for all states s, and select an $\epsilon > 0$. Set $m = 0$.

2 Set $v^{m+1}_s = T(v^m)(s) - T(v^m)(s_1)$ over all $s \in S$.

3 if $sp(v^{m+1} - v^m) > \epsilon$, increment m and go to step 2

4 For each $k \in N$, choose $\pi(k) = \arg\max_{k' \in N'}((1 - e^{-\lambda \Delta t})r_k + (1 - e^{-\lambda \Delta t}) \sum_{i \in N} p_i v_{k,i} + e^{-\lambda \Delta t} v_{k',\mathrm{Free}})$

3.4 State Space Reduction

The state space for the above Dynamic Programming model 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 $|N|^2$, making it intractable for large networks. This state space can be reduced by carefully observing equations (4)-(8) which show that if the ambulance gets dispatched to the next call, it follows a deterministic path to the call location and then becomes idle again either at call location or at hospital after some expected service time. Therefore we can rewrite $v_{k,i}$ in terms of the average reward g, the values $v_{i,\text{Free}}$, and the value $v_{h(i),\text{H}}$ corresponding to the hospital location $h(i)$ closest to node i:

$$\begin{aligned} v_{k,i} &= (1 - p_{\text{transport}})v_{i,\text{Free}} + p_{\text{transport}} v_{h(i),\text{Free}} \\ &\quad - (d_{k,i} + \frac{1}{1 - e^{-u_P \Delta t}})g \\ &\quad - p_{\text{transport}}(d_{i,h(i)} + \frac{1}{1 - e^{-u_{h(i)} \Delta t}})g \end{aligned} \tag{9}$$

where $d_{k,i}$ is the number of time steps required to travel from node k to i. This equation can be interpreted as follows. The first two terms show possible transitions from the point of being dispatched to becoming free upon the completion of a request. 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. This loss of reward is the product of service time and the average reward per time step. The third term shows the loss of reward due to traveling to the call location and the expected service time at the scene. The fourth term gives the further losses that occur as a result of any travel to and expected service time at the hospital.

If we substitute (9) in (4)-(8), we obtain a new system of average reward Bellman equations:

$$v_{k,\text{Free}} + g = (1 - e^{-\lambda \Delta t})(r_k + v_{k,\text{Busy}}) + \max_{k' \in N'} e^{-\lambda \Delta t} v_{k',\text{Free}} \quad \forall k \in N \tag{10}$$

$$\begin{aligned} v_{k,\text{Busy}} &= (1 - p_{\text{transport}}) \sum_{i \in N} p_i v_{i,\text{Free}} + p_{\text{transport}} \sum_{i \in N} p_i v_{h(i),\text{Free}} \\ &\quad -(\sum_{i \in N} p_i d_{k,i} + \frac{1}{1 - e^{-u_p \Delta t}})g \\ &\quad -p_{\text{transport}}(\sum_{i \in N} p_i (d_{i,h(i)} + \frac{1}{1 - e^{-u_{h(i))} \Delta t}}))g \end{aligned} \tag{11}$$

where $v_{k,\text{Busy}} = \sum_{i \in N} p_i v_{k,i}$ corresponds to a new aggregated busy state (k, Busy).

We further substitute equation (11) into equation (10), we have

$$\begin{aligned} v_{k,\text{Free}} + g = (1 - e^{-\lambda \Delta t})(r_k + (1 - p_{\text{transport}}) \sum_{i \in N} p_i v_{i,\text{Free}} + p_{\text{transport}} \sum_{i \in N} p_i v_{h(i),\text{Free}} \\ - (\sum_{i \in N} p_i d_{k,i} + \frac{1}{1 - e^{-u_p \Delta t}})g \\ - p_{\text{transport}}(\sum_{i \in N} p_i (d_{i,h(i)} + \frac{1}{1 - e^{-u_{h(i))} \Delta t}}))g + \max_{k' \in N'} e^{-\lambda \Delta t} v_{k',\text{Free}} \quad \forall k \in N \end{aligned}$$

This gives a much reduced state space $S = \{(k, \text{Free}), k \in N\}$ for which we have $|S| = |N|$. A problem arises when we use the relative value iteration with this formulation. Note that the average reward g appears on the right hand side, which is unknown until we find the optimal policy. We address this problem by using a modified relative value iteration. In this modified algorithm, we use an approximate average reward g^m at each iteration, which then is updated using the current average increase rate over all states. The modified relative value iteration is listed in Algorithm 3. The computational gains are obvious as we solved the standard DP using

---

Algorithm 3 The modified relative value iteration

---

1 Initialization:

Initilize $g^m = 0$ and $v^m_{k,\text{Free}} = 0, \forall k \in N$. Select a reference state (k_1, Free).

Select an $\epsilon > 0$.

Set iteration count $m = 0$.

2 Solve Bellman's Equations:

$$\begin{aligned} v^{m+1}_{k,\text{Free}} = &(1 - e^{-\lambda \Delta t})(r_k + (1 - p_{\text{transport}}) \sum_{i \in N} p_i v^m_{i,\text{Free}} + p_{\text{transport}} \sum_{i \in N} p_i v^m_{h(i),\text{Free}} \\ &- (\sum_{i \in N} p_i d_{k,i} + \frac{1}{1 - e^{-u_p \Delta t}})g^m - p_{\text{transport}}(\sum_{i \in N} p_i(d_{i,h(i)} + \frac{1}{1 - e^{-u_{h(i)} \Delta t}}))g^m) \\ &+ \max_{k' \in N'} e^{-\lambda \Delta t} v^m_{k',\text{Free}} \quad \forall k \in N. \end{aligned}$$

3 If $sp(v^{m+1} - v^m) < \epsilon$, go to step 4; otherwise set

$$\begin{aligned} m &= m + 1, \\ g^m &= (1 - \alpha) g^{m-1} + \alpha \frac{\sum_{k \in N}(v^m_{k,\text{Free}} - v^{m-1}_{k_1,\text{Free}})}{|N|}, \\ v^m_{k,\text{Free}} &= v^m_{k,\text{Free}} - v^{m-1}_{k_1,\text{Free}} \quad \forall k \in N. \end{aligned}$$

and go to step 2.

4 Set the optimal policy as:

$$\begin{aligned} \pi(k) = \arg\max_{k' \in N'} &((1 - e^{-\lambda \Delta t})(r_k + (1 - p_{\text{transport}}) \sum_{i \in N} p_i v^m_{i,\text{Free}} + p_{\text{transport}} \sum_{i \in N} p_i v^m_{h(i),\text{Free}} \\ &- (\sum_{i \in N} p_i d_{k,i} + \frac{1}{1 - e^{-u_p \Delta t}})g^m - p_{\text{transport}}(\sum_{i \in N} p_i(d_{i,h(i)} + \frac{1}{1 - e^{-u_{h(i)} \Delta t}}))g^m) \\ &+ e^{-\lambda \Delta t} v^m_{k',\text{Free}}) \quad \forall k \in N. \end{aligned}$$

---

the relative value iteration and the reduced DP using the modified relative value iteration on a line of 30 nodes. The probability of the next call occurring at each node is randomly generated. The response time target is 4 minutes and the hospital is at node 2. The reduced DP took less than 1 second to converge while the standard DP took 7 minutes to converge.

3.5 Examples

In this section, we present two example problems for the single vehicle infinite horizon problem. We also compare move up with a benchmark 'return-to-base' policy.

Consider a network consisting of 30 nodes on a line, a target response time of $W = 3$ minutes and a reward function r_k as shown in Figure 6. The at-scene service rate is 10 calls per hour. The at-hospital service rate is 1.2 calls per hour. There is a 0.2 probability of the vehicle becoming free at scene and a 0.8 probability of becoming free at hospital. There is a single hospital at node 24. In order to compare move up with a benchmark 'return-to-base' policy, we modified the model above to implement this no-moveup policy, and then evaluated all possible base locations to find that giving the best average reward.

Table 2 shows the results for two call arrival rates. Solving our dynamic program for the first $\lambda = 0.5$ case gave a moveup policy with just one optimal standby location at node 6. As we would expect, our exhaustive search under the benchmark return-to-base policy showed that the base should be placed at this location, giving a policy (and hence performance) identical to that with moveup. Solving the $\lambda = 3$ case gave a moveup policy with two optimal standby locations at nodes 6 and 26, with the best return-to-base policy occuring when the base was placed at node 6. With move up, the ambulance is able to respond on time to 0.463 calls per hour, a 4% improvment over the 0.445 calls per hour achieved without moveup.

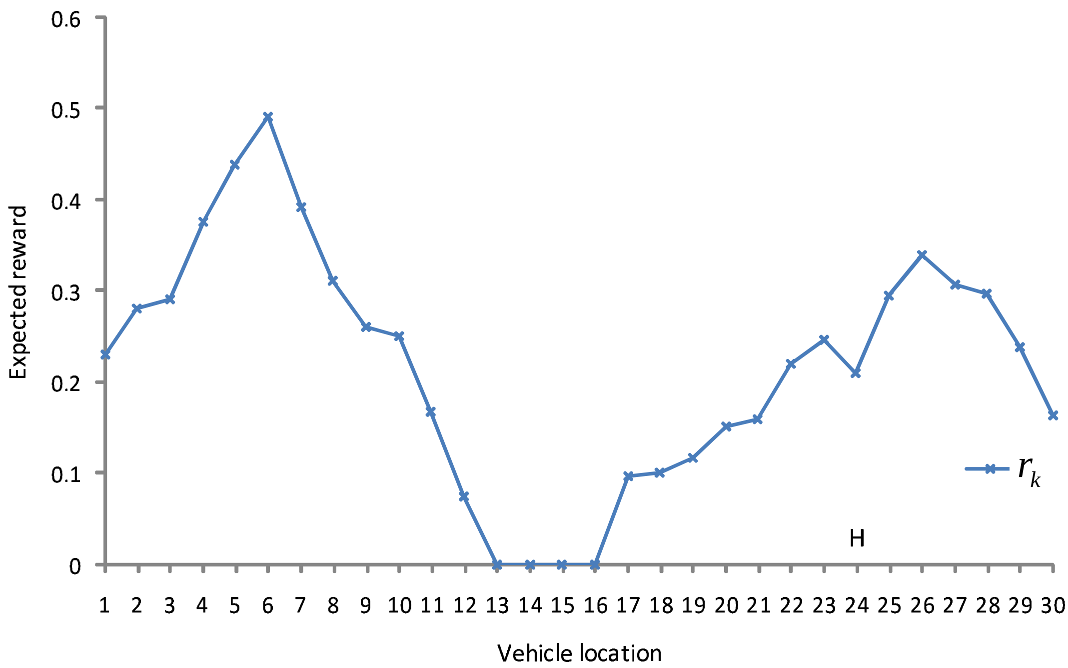

Figure 6 – The reward function r_k for a 30-node problem with a hospital at node 24.

In our earlier single-ambulance next-call model, we showed that standby locations occurred at local maxima of r_k, and that any standby location found in an optimal low-λ solution is also a standby location for a higher arrival rate λ. We now show that these two results do not hold for the infinite horizon model. Figure 7 shows a network consisting of 250 nodes located on a line, a target response time of $W = 8$ minutes and a reward function r_k as shown. The at-scene service rate is 10 calls per hour and the service rate at hospital is 1.2 calls per hour. The hospital is at node 100. We first solved this model using a very small call arrival rate of 0.001 call per hour and found node 242 to be the single optimal standby location. Intuitively,

	No Moveup		Optimised Moveup			
λ	Best base	g^{base}	Vehicle location	Optimal standby node	g^*	Improvement
0.5	6	0.185	1-30	6	0.185	0%
3	6	0.445	1-19 20-30	6 26	0.463	4%

Table 2 – Comparison between moveup and return-to-base for two call arrival rates

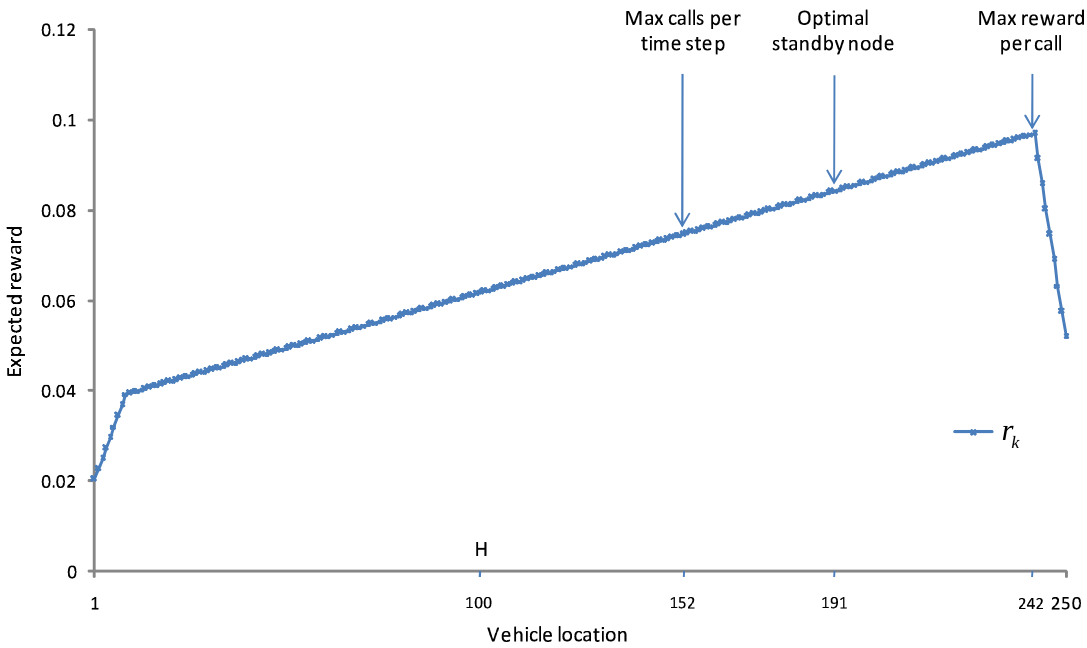

Figure 7 – The reward function r_k and optimal move-up policy for a 250-node problem with $\lambda = 2$ calls per hour.. The optimal standby location (node 191) illustrates the trade-off required between average service time and expected call reward.

the call arrival rate is small enough to allow the vehicle enough time to complete its moveup before the next call arrives. Locating the vehicle at node 242 gives the maximum reward for each arriving call. We then increased the call arrival rate to 2 calls per hour, and found that the optimal standby location changed to node 191. This confirms our two assertions above: the previous standby location, node 242, is no longer a standby location for an increased call arrival rate and the new standby location is not a local maximum of r_k. To understand why, we note that this optimal standby location represents a trade-off between two objectives. The first objective is the reward received from each call responded to. Clearly, to maximise this reward, the ambulance should wait at node 242, being the global maximum of r_k. However, one also needs to consider the time taken to service each call which determines the number of calls that the vehicle responds to per hour. This service time includes the travel time to the scene, which depends on the choice of standby location. In this case, we maximize the number of calls served per hour by locating the vehicle at node 152. The optimal standby location of 191 represents a trade off between these two extremes. The importance of these two factors in practice is likely to depend upon their relative magnitudes for a full scale problem.

4 Discussions

This paper has developed new optimal models for the single vehicle moveup problem and provided insights into the form and structure of optimal moveup policies and the benefit that move up can offer. We have presented examples showing that an optimal move up policy for a single ambulance is influenced by several factors including the call arrival rate λ and the reward function r_k along a potential moveup path.

The models we propose only consider a single ambulance, but are currently being extended to handle multiple vehicles. The state space required to model multiple vehicles inevitably becomes too large to deal with exactly and therefore some candidate approaches such as heuristics, approximate Dynamic Programming and scenario planning are being explored to solve these problems. This work is being guided by the insights from models we have described in this paper.

A Call Arrival Distribution Over the Network for Example 2

p_1	0.0048	p_{10}	0.0335	p_{19}	0.0008	p_{28}	0.0239
p_2	0.0239	p_{11}	0.0093	p_{20}	7.421E-05	p_{29}	0.0287
p_3	0.0359	p_{12}	0.0119	p_{21}	0.0349	p_{30}	0.0007
p_4	0.0431	p_{13}	0.0048	p_{22}	0.0048	p_{31}	0.0682
p_5	0.0321	p_{14}	0.0002	p_{23}	0.0115	p_{32}	0.1053
p_6	0.0074	p_{15}	0.0005	p_{24}	0.0718	p_{33}	0.1341
p_7	0.0005	p_{16}	7.181E-05	p_{25}	0.0958	p_{34}	0.0007
p_8	7.421E-05	p_{17}	0.0192	p_{26}	0.1196	p_{35}	0.0048
p_9	0.0119	p_{18}	0.0048	p_{27}	0.0503		

Table 3 – The call arrival distribution for the next call

B 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 ties are broken in some predictable manner. 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:

$$V_k = \max\left(r_k, (1 - e^{-\lambda\Delta t})r_k + e^{-\lambda\Delta t} \max_{j \in N_k \cap P} V_j^*\right) \tag{12}$$

Comparing this with the optimality equation, (1), 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 excluded in (12) for node p has $V_k^* \leq V_p^*$, and so excluding this node in (1) would not alter the calculated V_p^* value, making (1) and (12) 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 (assuming any ties are broken using the approach adopted when renumbering the nodes).

(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.

References

[1] O. Berman, Dynamic repositioning of indistinguishable service units on transportation networks., Transportation Science 15 (2) (1981) 115–136.

[2] M. Gendreau, L. G., F. Semet, A dynamic model and parallel tabu search heuristic for real time ambulance relocation., Paralell Computing 27 (12) (2001) 1641–1653.

[3] H. Robert, G. Andrew, Elevator group control using multiple reinforcement learning agents, Machine Learning 33 (1998) 235–262.

[4] T. C., R. C. Swain, R., L. Bergmann, The location of emergency service facilities., Operations Research 19 (6) (1971) 1363–1373.

[5] R. Church, C. ReVelle, The maximal covering location problem., papers of the Regional Science Association (1974).

[6] K. Hogan, C. S. ReVelle, Concepts and applications of backup coverage, Management Science 34 (1986) 1434–1444.

[7] M. Gendreau, G. Laporte, F. Semet, Solving an ambulance location model by tabu search., Location Science 5 (1997) 75–88.

[8] T. Andersson, S. Petersson, P. Varbrand, Opal-optimized ambulance logistics, le Gosier, Guadeloupe (2004).

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

[10] A. Ingolfsson, S. Budge, E. Erkut, Optimal ambulance location with random delays and travel times, Health Care Management Science 11 (3) (2008) 262–274.

[11] R. Larson, A hypercube queueing model for facility location and redistricting in urban emergency services., Computers and Operations Research 1 (1974) 67–75.

[12] R. Larson, Approximating the performance of urban emergency service systems, Operations Research 23 (1975) 845–868.

[13] S. Henderson, A. Mason, in brandeau, m.l., sainfort, f., and w.p. pierskalla, eds: Ambulance service planning: Simulation and data visualization 70 (2004) 77–102.

[14] E. Bryan, D. Bledsoe, FACEP, EMT-P, Ems myth 7: System status management lowers response times and enhances patient care, http://publicsafety.com/article/article.jsp?id=2030.

[15] S. Jack, System status management. the fact is, it's everywhere, http://www.stouts.org/jack/EMS/JEMS0489.htm (1989).

[16] M. Gendreau, L. G., S. F., The maximal expected coverage relocation problem for emergency vehicles., Journal of the Operational Research Society 57 (1) (2006) 22–28.

[17] D. P. Richards, Optimised ambulance redeployment strategies, master's thesis, Department of Engineering Science, University of Auckland, Auckland, New Zealand. (2007).

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

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

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

[21] S. G. Henderson, H. Topaloglu, M. M. G., Approximate dynamic programming for ambulance redeployment, Proceedings of the 2009 Winter Simulation Conference.

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

[23] s. D. Bertaseka, J. Tsitsiklis, An analysis of stochastic shortest path problems 16.

[24] M. S., Average reward reinforcement learning, Machine Learning 22 (1-3) (1996) 159–195.

[25] W. D., Dynamic programming, markov chians, and the method of successive approximations, Journal of Mathematical Analysis and Applications 6 (1963) 373–376.