Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

EXPLORING BOUNDS ON AMBULANCE DEPLOYMENT POLICY PERFORMANCE

Eric Cao Ni
Susan R. Hunter
Shane G. Henderson
Huseyin Topaloglu

School of Operations Research and Information Engineering
Cornell University
Ithaca, NY 14853, U.S.A.

ABSTRACT

Ambulance deployment involves controlling a fleet of ambulances, often in real time, in an attempt to keep response times small. Simulation has been used to devise redeployment policies, and bounds have been obtained from a combination of comparison methods for queues (coupling) and simulation. These techniques yield varying results on two realistic examples. In an attempt to understand the varying results, we explore the performance of the policies and bounds on artificial models.

1 INTRODUCTION

Ambulance deployment is the practice of positioning ambulances around a city, perhaps using real-time information. Positions are chosen to attempt to keep response times – the time from when a call is received until an ambulance arrives at the scene of the call – small. Response times are usually summarized by the fraction of calls with response times under some time threshold that is typically taken to be 8 or 9 minutes. We describe a call as "late" if its response time exceeds the time threshold. An important practical goal, then, is to choose a deployment policy that minimizes the fraction of calls that are late.

Recently, methods have been proposed for computing bounds on what can be achieved with *any* deployment policy, whether the policy uses real-time information or not (Maxwell et al. 2012). Such bounds can be used to

1. help identify when a given policy is close to optimal, and
2. help determine whether deployment strategies have the potential to improve performance to within some target, or to firmly establish that some other approach, such as increasing resources, is needed.

The computational results in Maxwell et al. (2012) showed that the difference in performance between an implementable policy and the bound (henceforth called the *gap*) was small (about 2%) in one realistic example of a city, and large (about 10%) in another. A gap of 2% indicates that the given policy is near optimal, but the gap of 10% suggests that either the policy or the bound, or both, can be greatly improved.

We want to understand why the gap varies in this way. In particular, what characteristics of a city will lead to small gaps, and in cases where the gap can be expected to be large, are there other bounds that can be derived? In this paper we provide a simulation study that attempts to shed light on this question. We generate a number of artificial "cities" that have varying characteristics that we hypothesize might have an impact on the gap. We then use a simulation study to compute the gaps for each artificial city and look for important factors.

The factors we consider are

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

1. the number of modes in the two dimensional probability distribution for the location of incoming calls,
2. the degree to which the location distribution is concentrated around the modes, and
3. the degree to which the locations where ambulances can park (called *bases* here, even though they may be as simple as a vacant carpark) are concentrated around the modes of the location distribution.

The remainder of this paper is organized as follows. Section 2 discusses methods for choosing a redeployment policy and reviews one such method that we use in our computational study. Section 3 reviews a method for obtaining performance bounds that we use in our computational study. The computational study itself is described in Section 4. Section 5 discusses the results and concludes the paper.

2 CHOOSING A POLICY

There are a large number of methods for determining how to position ambulances in a city. These methods can be divided into *static* policies, wherein each ambulance operates out of a given location throughout its shift, returning to that location when it is not engaged with a call, and *dynamic* policies that relocate ambulances throughout their shifts. Dynamic methods may involve real-time control, making decisions on ambulance locations in real time using real-time information on ambulance location and status. Policies using real-time information are often called "system-status management" policies or "move-up" policies.

Techniques for designing static policies date back to the 1960s, e.g., Bell and Allen (1969), with Church and ReVelle (1974) and Daskin (1983) being particularly important references. Two excellent surveys are Swersey (1994) for work up to the early 1990s, and Brotcorne, Laporte, and Semet (2003) for more recent work.

Dynamic methods that do not involve real-time control include Rajagopalan, Saydam, and Xiao (2008), Schmid and Doerner (2010).

If one has the added benefit of real-time information on the location and status of ambulances, then one can use real-time methods that exploit this information. Real-time methods may be constructed by solving integer programs in real time (Gendreau, Laporte, and Semet 2001; Brotcorne, Laporte, and Semet 2003; Richards 2007; Nair and Miller-Hooks 2009). Alternatively, one can construct a look-up table, called a compliance table, giving desirable locations of ambulances as a function of the number of available ambulances. Dispatchers then attempt to dispatch ambulances to adhere to the compliance table, so the policies depend not just on the table but also on dispatching. Compliance tables can be constructed by presolving integer programs (Gendreau, Laporte, and Semet 2006), or by screening potential tables with fast approximate models and then checking the top contenders through simulation (Alanis, Ingolfsson, and Kolfal 2012). Real-time policies can also be obtained by solving or approximately solving stochastic dynamic-programming formulations. Exact formulations that yield insight include Berman (1981a), Berman (1981c), Berman (1981b), Zhang, Mason, and Philpott (2010), Zhang (2010). Approximate formulations that scale to realistic problems include Maxwell et al. (2010), Maxwell, Henderson, and Topaloglu (2011), Schmid (2012). One can also develop policies through heuristic approaches such as the preparedness concept (Andersson 2005; Andersson and Vaerband 2007). In the remainder of this section we go into more detail on the method for designing real-time policies in Maxwell, Henderson, and Topaloglu (2011), because it is used to obtain a feasible policy, and therefore an upper bound on the fraction of late calls, in the computational study to follow.

2.1 Dynamic Programming Formulation

In the system we consider, arriving patient calls are served in first-come-first-serve fashion. To handle a call, an ambulance moves to the scene of the call and provides service. After completion of service, the ambulance may or may not have to transport the patient to a hospital. After the patient is transported to a hospital, if necessary, the ambulance is available to serve another call. If there is one such call, then the ambulance moves to the scene of the next call and a similar cycle is repeated. Otherwise, we need to decide where to reposition the available ambulance to serve future calls in the most effective fashion. Thus, the crucial decision in the problem is where to reposition an ambulance once it becomes available after serving a call.

Our approach in this paper is based on formulating the ambulance deployment problem as a stochastic dynamic program and using tractable approximations of the value functions. To represent the state of the system, we assume that there are N ambulances and keep track of the state of the ambulances by using an N-tuple whose components give information relevant to the state of each ambulance. The state of ambulance i is given by $a_i = (\sigma_i, \ell_i, d_i, t_i)$, where σ_i is the status of the ambulance, ℓ_i and d_i are respectively the origin and destination locations of the ambulance and t_i is the starting time of any ambulance movement. The status of an ambulance can take values such as "idle at base," "going to call," etc. If ambulance i is moving towards a certain location, then t_i corresponds to the time this movement began. Otherwise, t_i represents the starting time of the current phase in the service cycle, which, for example, may correspond to the time at which the ambulance began serving a call.

We assume that we can keep at most M calls in the waiting call list. The state of call j in the waiting call list is given by $c_j = (\delta_j, p_j, \zeta_j)$, where δ_j is the status of the call, p_j is the location of the call and ζ_j is the time at which the call arrived into the system. The status of a call can take values such as "assigned to ambulance i," "waiting for service," etc. The system is driven by events and we keep track of the state of the system at these event times only. The possible events in the system are "call arrives and is placed in the jth position," "ambulance i departs for scene of call j," "ambulance i arrives at scene of call j," "ambulance i leaves scene of call j for hospital," "ambulance i arrives at hospital," "ambulance i finishes at hospital" and "ambulance i arrives at base." We make decisions only at the times of these events.

Given that we make decisions only at event times, we represent the state of the system at a time point by $s = (\tau, e, A, C)$, where τ is the current time, e is the current event, $A = (a_1, \ldots, a_N)$ captures the state of the ambulances and $C = (c_1, \ldots, c_M)$ captures the state of the calls in the waiting call list. This setup still allows us to make decisions at other times, if we simply define those times as event times.

We use $\mathscr{X}(s)$ to denote the set of feasible actions in state s. Given that we are in state s, if the current event corresponds to completing a call, meaning that the ambulance serving a call becomes available, and there are no other calls in the waiting call list, then $\mathscr{X}(s)$ includes all possible locations that the ambulance can be repositioned to. We assume the existence of a systems dynamics equation $f(\cdot)$ such that $f(s,x,U)$ gives the next state of the system as a function of the current state s, current action x and the random noise captured by the vector of uniform random variables U. We use $c(s,x,U)$ to denote the cost of our decisions, capturing the cost incurred when we take action x in state s and the random noise turns out to be U. For our application, the cost function tracks the late calls. Thus, if the action x involves assigning an ambulance to a call and the ambulance cannot reach the call on time given that the state of the system is s, then we incur a cost of one, and otherwise the cost is zero. In this case, we can formulate the problem as a dynamic program as

$$V(s) = \min_{x \in \mathscr{X}(s)} \mathbb{E}\Big\{ c(s,x,U) + V(f(s,x,U)) \Big\}. \tag{1}$$

Throughout, we assume that the cardinality of the set of feasible actions $\mathscr{X}(s)$ is small enough that we can solve the minimization problem on the right side above by enumerating all elements of $\mathscr{X}(s)$. However, even if the cardinality of the set of feasible actions is small, solving the optimality equation above requires computing an expectation. Due to the need to compute this expectation, even following the policy induced by a value function can be computationally intractable. In the next section, we describe an approach that sidesteps the difficulty associated with the expectation.

2.2 Post-Decision State Variable and Search for Good Policies

Given that we are in state s and we apply the action x, we use $s^+(x)$ to denote the state of the system immediately after applying the action x but before passage of any time. For example, if the decision x

is to send an ambulance to a particular location, then the status and the origin and destination locations of the ambulance are updated accordingly to obtain the state $s^+(x)$, but the states of all other ambulances remain the same. The state $s^+(x)$ is the post-decision state of s after taking action x; see Powell (2011) for a detailed discussion of post-decision states in approximate dynamic programming. We observe that the cost function we use can be written as a function of $s^+(x)$ and U because once an ambulance starts moving towards a call, only the random noise will determine whether the ambulance will be late or not. Thus, we write the immediate cost function as $c(s^+(x),U)$. Similarly, the state transition can be captured by $f(s^+(x),U)$. In this case, the expected cost to go from being in post-decision state $s^+(x)$ is given by

$$\mathbb{E}\Big\{c(s^+(x),U)+V(f(s^+(x),U))\Big\}. \tag{2}$$

If we use $J(s^+(x))$ to denote the expression above, then (1) and (2) imply that given that we are in state s, we can find the optimal action by solving

$$\min_{x\in\mathscr{X}(x)} J(s^+(x)).$$

The key observation is that if we have a good approximation $\tilde{J}(\cdot)$ to $J(\cdot)$, then we can find a good action to take in state s simply by solving the problem $\min_{x\in\mathscr{X}(s)}\tilde{J}(s^+(x))$. Furthermore, solving the last optimization problem does not require dealing with expectations at all.

Motivated by this observation, we approximate $J(\cdot)$ with a function of the form $\tilde{J}(\cdot,r)$, where r represents a vector of adjustable parameters. The exact form of the function $\tilde{J}(\cdot,r)$ is specific to the application on hand. In the ambulance redeployment setting, we view each base as a separate Erlang loss system with call arrival rate equal to the rate of calls arriving within the vicinity of the base. Then, using a reasonable approximation to the service rate and the number of ambulances assigned to the base, we can estimate the rate of calls that are “lost” without receiving service from a particular base. (Calls are not actually lost, but we use this approach as an approximation for the fraction of late calls.) In this case, $\tilde{J}(\cdot,r)$ is a linear combination of the rate of lost calls from each of the bases, and the adjustable parameters r are the multipliers in the linear combination. For precise details of the approximation $\tilde{J}(\cdot,r)$, we refer the reader to Maxwell, Henderson, and Topaloglu (2011). Thus, for each set of adjustable parameters r, we have an approximation $\tilde{J}(\cdot,r)$ to $J(\cdot)$. This approximation defines a policy, where the action we take in state s is given by

$$\min_{x\in\mathscr{X}(s)} \tilde{J}(s^+(x),r). \tag{3}$$

The goal is to adjust the parameters r so that $\tilde{J}(\cdot,r)$ is a good approximation to $J(\cdot)$, in which case, we hope that the policy obtained through (3) is a good policy.

There are a number of methods to adjust the parameters r to ensure that $\tilde{J}(\cdot,r)$ is a good approximation to $J(\cdot)$. Many of these methods can roughly be visualized as regression based methods where one chooses r so that $\tilde{J}(\cdot,r)$ is a good fit to the sampled cost trajectories of the system. In this paper, we follow a more direct approach. In particular, a value of r induces a policy through (3). Let $\Pi(r)$ be the expected cost of the policy induced by the adjustable parameters r. Then, we can try to find a good set of adjustable parameters r by solving the problem

$$\min_{r}\ \Pi(r). \tag{4}$$

The problem above tries to find a set of adjustable parameters r so that the expected cost of the policy induced by these parameters is as small as possible. Using Problem (4) to find a good set of adjustable parameters, rather than fitting $\tilde{J}(\cdot,r)$ to $J(\cdot)$, is a more direct approach because $\Pi(r)$ gives a clear indication of how well the policy induced by the parameters r will work in practice.

The objective function of problem (4) is difficult to compute analytically as it corresponds to the expected cost incurred by a particular ambulance redeployment policy. Nevertheless, it is straightforward to obtain estimates of this expected cost by simulating the policy characterized by the adjustable parameters r. Thus, we can use derivative-free simulation optimization methods to try to solve problem (4). Maxwell et al. (2012) use the Nelder-Mead method to identify a solution r^* to this problem. Given such an r we can come up with a policy by solving problem (3) whenever we need to make a decision. It is conceivable that the function Π is multimodal, so that one may wish to apply multiple local searches with the Nelder-Mead algorithm from multiple randomly chosen starting points. In this paper we do not use simulation optimization to identify r^* but rather use a set of coefficients that yields good performance in a range of numerical experiments. This saves us a great deal of computational time because simulation optimization is computationally expensive. We do hope to extend our computational experiments in future to include this optimization step.

3 BOUNDING PERFORMANCE

In the previous section we described how we can obtain a policy, and therefore an upper bound on the fraction of late calls. We now briefly review one of two methods, derived in Maxwell et al. (2012), for obtaining *lower* bounds on this quantity. For full details and a more precise derivation of the bounds, see Maxwell et al. (2012). We review just one of the bounds, the *cover* bound, here because both bounds gave similar results when tested on two realistic examples in Maxwell et al. (2012), and the cover bound requires fewer assumptions. We only report the cover bound in our computational results.

The cover bound is based on a "comparison of queues," also known as a *coupling* of queues or, in the simulation community, a careful use of common random numbers. Consider the ambulance system as a queueing system in which ambulances are servers and the service time consists of the time from when an ambulance is assigned to a call till the time when the ambulance completes the call, either at the scene or at a hospital. The cover bound relies on two key ideas:

1. The distribution function of a service time for a call depends on the number and location of available ambulances at the time the call is received. For each number of available ambulances, we find a stochastic lower bound on this family of distribution functions.
2. Assume that at all times the available ambulances (i.e., those not currently engaged in a call) are located at sites that minimize the probability that the next call will be late.

The cover bound is then computed by simulating a queueing system with the same call arrival process as the real ambulance system, and service times that are computed based on Idea 1 above. This ensures that there are always more ambulances available than in reality in this bounding system. Idea 2 is exploited in the statistics collected: Instead of recording a "1" if a call is late and "0" otherwise, we instead record the conditional probability that the call is late, assuming that the available ambulances at the time of the call are distributed as in Idea 2.

To be more precise, let $G_k(\cdot;c,t)$ be the distribution function of the service time for a call that arrives at time t, when there are k ambulances available, and where c gives the configuration (locations) of the k available ambulances at time t. (In the simulation of the ambulance service, we never actually compute this distribution function, but it is needed implicitly for computing the cover bound.) If we assume that the distribution of call locations does not depend on time, then $G_k(\cdot;c,t)$ does not depend on t, and we can then use the notation $G_k(\cdot;c)$. We will explain below how we can find a stochastic lower bound on $G_k(\cdot;c)$ for all possible configurations c, separately for each $k = 1,2,\ldots,a$, where a is the number of ambulances (available or not). Let $\tilde{G}_k(\cdot)$ be this stochastic lower bound, so that $\tilde{G}_k(\cdot) \geq G_k(\cdot;c)$ for all c, for each $k = 1,2,\ldots,a$. Now suppose that, conceptually speaking, in simulating the ambulance service we generate service times using inversion from the distribution function $G_k(\cdot;c)$ whenever there are k ambulances available in configuration c when a call is received. This yields sample paths with the correct distribution.

Let U_j be the uniform random variable used to generate the jth service time. Using common random numbers, we can simulate a *bounding* queueing system where the arrival process of calls is the same as in the real system, but the jth service time is computed as $\tilde{G}^{-1}_{\tilde{k}(j)}(U_j)$, assuming there are $\tilde{k}(j)$ ambulances available in the bounding queueing system at the time of the jth call. Then, under the additional assumption that $\tilde{G}_1(\cdot) \leq \tilde{G}_2(\cdot) \leq \cdots \leq \tilde{G}_a(\cdot)$, i.e., that the service-time distribution functions in the bounding system are stochastically decreasing in $\tilde{k}$, it follows from induction on the sequence of arriving customers that the service times in the bounding system will always be at most equal to those in the real system, and therefore that the number of available ambulances in the bounding system will always be at least equal to the number of available ambulances in the real system. In other words, we have defined a coupling that ensures that the bounding system dominates the real system in terms of the number of available ambulances.

The bounding system has more available ambulances than the real system at any time, and certainly at the times of call arrivals. But if these ambulances are poorly positioned, or if the call location is, by chance, close to an ambulance in the real system, then the response time to the call could be smaller in the real system than in the bounding system. To avoid this, we now employ Idea 2, assuming that in the bounding system the ambulances are always in the position that minimizes the probability of being late on the next call. This ensures that we obtain a lower bound on the probability of being late on the next call in the bounding system, and by averaging this lower bound over all calls we obtain a lower bound on the expected number of calls that are late. On any sample path, it is possible that the response time for any call will be smaller in the simulated system than in the bounding system, but in expectation, and therefore for sufficiently long simulation runs, this cannot happen.

This then completes the explanation of the cover bound, except for how we compute the stochastic lower bound on the service time distribution, and for how we compute the positions that minimize the probability that the next call will be late. These quantities are obtained by solving a number of integer programs.

First consider the lower bound, $v(k)$ say, on the probability that a call will be late when it arrives and there are k ambulances available, as required for Idea 2. These k ambulances can be in any location throughout the city, so if we solve an optimization problem that minimizes the probability that the call is late over possible ambulance locations, then we obtain the desired lower bound. We take the set of possible ambulance locations to be the discrete and finite set $1,2,\ldots,J$ to make the optimization problem tractable. For $1 \leq k \leq a$, let $v(k)$ be the optimal objective value of the integer program (Church and ReVelle 1974)

$$\begin{aligned}
\min \quad & \sum_{j=1}^{J} d_j(1-w_j) & & \\
\text{s.t.} \quad & \sum_{i=1}^{J} x_i \leq k & & \\
& w_j \leq \sum_{i=1}^{J} \delta(i,j) x_i & & \forall j = 1,2,\ldots,J \\
& x_i \in \{0,1\} & & \forall i = 1,2,\ldots,J \\
& w_j \in \{0,1\} & & \forall j = 1,2,\ldots,J.
\end{aligned} \tag{5}$$

Here d_j is the probability that the call will arise at location j, $\delta(i,j)$ is 1 if Location j can be reached within the time threshold by an ambulance originating at Location i, and the decision variables x_i equal 1 if an ambulance is placed at location i and 0 otherwise, and w_j equals 1 if Location j can be reached on time by some ambulance, and equals 0 otherwise.

The process for obtaining the stochastic lower bounds $\tilde{G}_k(\cdot)$ also involves solving certain integer programs. To obtain $\tilde{G}_k(r)$ for any fixed k and r we solve an integer program. This integer program selects k ambulance locations that maximize the probability that the service time will be completed within a time of length r. The important decision variables specify the locations of the ambulances and the location i from which calls at j should be responded from, for each i and j. These integer programs are known as p-median problems, and can be difficult to solve. Fortunately, in cases where the integer program is difficult to solve, the objective function of the linear-programming relaxation yields a bound that can be used in computing our overall lower bound, and appears to be tight enough for our purposes.

We solve these integer programs for each $k = 1, 2, \ldots, a$, and for each r on a lattice of time values. The resulting objective values specify $\tilde{G}_k(\cdot)$ on that same lattice of time values, and it remains to define $\tilde{G}_k(\cdot)$ at non-lattice values. The details, including the integer-programming formulations, are somewhat involved, so we refer the reader to Maxwell et al. (2012).

4 COMPUTATIONAL STUDY

In the previous sections, we discussed the design of ambulance redeployment policies and the computation of a lower bound. Maxwell et al. (2012) observed a large difference in the size of the "optimality gap" between the performance of a particular policy and the lower bound in two realistic examples. In this section, we study the causes of such gaps by running a $3^2 \times 2$ full-factorial experiment.

Our experiment is carried out on fictional cities that are 15 miles by 15 miles large, with 7 ambulances each traveling at 24 miles per hour. Distances from point to point are computed using the Manhattan metric. Each city has 25 ambulance bases and 2 hospitals. The word "base" here simply means a location where ambulances might be asked to wait for their next call, and is often referred to as a "post" in the industry. The arrival of emergency calls is Poisson with a rate of 3 calls per hour, and call locations are chosen independently from a probability distribution in two dimensions. The density of the location distribution is piecewise constant on a 5×5 grid on the city.

The factors we consider in our $3^2 \times 2$ full-factorial design include nine different demand distributions (three levels each of two demand distribution factors), and two levels of locations for the ambulances bases. We model two key factors in the call location distribution. First, our fictional cities are designed to have 1, 2 or 5 peaks in the demand density. One peak represents a city with one city center, two peaks represents a city with two city centers as in a "twin city," and five peaks represents a city with a main city center and four suburban areas, as in a "hub and spoke" design. Second, the concentration of each peak varies across different cities. We chose three different levels of demand concentrations to represent cities with varying degrees of sprawl. Once the demand distribution is determined, we use two different strategies to locate ambulance bases. In the first strategy, bases are located uniformly across the city, with one base at the center of each 3 mile-by-3 mile square in the grid. In the second strategy, we choose base locations close to the peaks in the demand. Figure 1 shows the maps of the nine cities, with each panel depicting the demand density (color coded so that regions with high demand intensities are red), the hospital locations, and the base locations in the second strategy. The base locations for the first (uniform) strategy are found at the centers of the 3 mile-by-3 mile squares of the grid. As indicated in Figure 1, each fictional city has two hospitals, where the hospital locations do not vary by city. The name of a fictional city is given by XYZ, where X is the number of peaks, 1, 2, or 5; Y is the peakedness, H for high, M for medium, L for low; and Z is the configuration of the ambulance bases, C for close to demand and U for uniformly distributed.

In the simulation model, we assume zero turnout time for ambulances responding to a new call. When an ambulance reaches the emergency scene, the time it spends there is exponentially distributed with a mean of 12 minutes. Subsequently, with probability 0.75 the patient will be transported to a hospital, and the choice between the two hospitals is random with probabilities 0.4 and 0.6 respectively. Upon reaching the hospital, the time required to transfer the patient into the care of the hospital is Weibull distributed with shape parameter (alpha) 2.5 and mean 30.4 minutes. After this, the ambulance is free to be redeployed to one of the bases, unless calls have queued up, in which case the ambulance is deployed to the first call received.

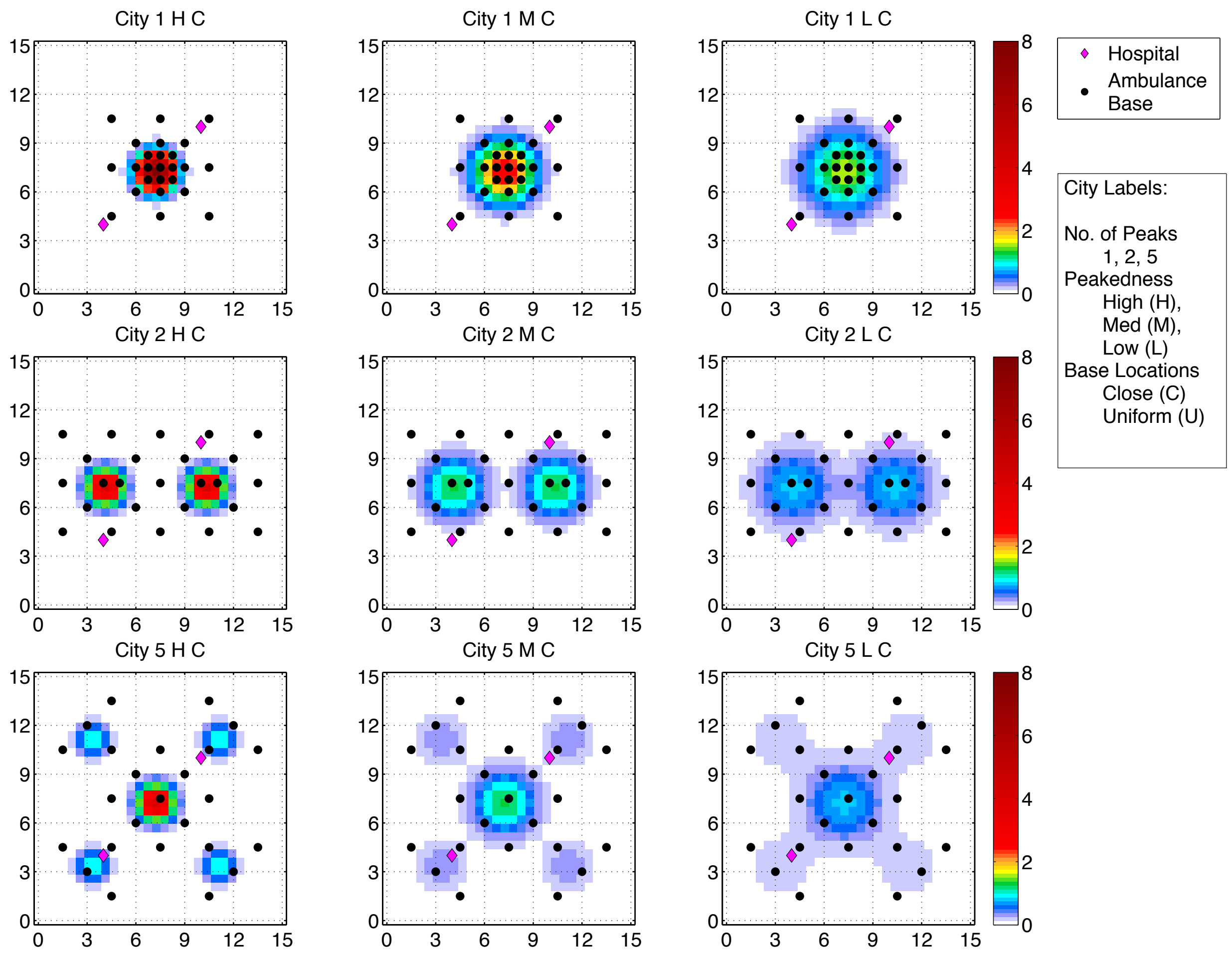

Figure 1: Distribution of demand, bases (strategically located), and hospitals in experimental cities

With three different numbers of peaks, three different level of peakedness and the two base allocation strategies, we have in total eighteen configurations of an experimental city. Given information on travel network, base locations, various time components of an emergency service cycle together with the incoming call distributions for each of these configurations, a cover bound as described in Section 3 can be computed. We compare this bound with the performance of an implementable policy that is described in Section 2. For each of the 18 configurations, we use common random numbers to simulate the bounding system in parallel with the implementable policy. The computation is carried out on an Ubuntu 11.10 platform with an Intel Core 2 CPU running at 2.67GHz with 2.00 GB RAM. It takes approximately 10 hours to solve the integer programs required for the cover bound using the Gurobi solver for C++, and 45 minutes to run both the bounding system and the stylized simulation for 2000 iterations.

The results of the simulation runs are summarized in Figures 2, 3, and 4. Figure 2 shows box plots of the estimated percent of late calls as output from the simulation for each fictional city. A box plot for the lower bound of the estimated percent of late calls is also shown for each city. We are particularly interested in the gap between the simulation performance and the lower bound, and box plots of this gap are plotted for each city in Figure 3. Finally, to better assess trends in the data across the main effects, we show main effects boxplots for the gap between the simulation performance and the lower bound in Figure 4.

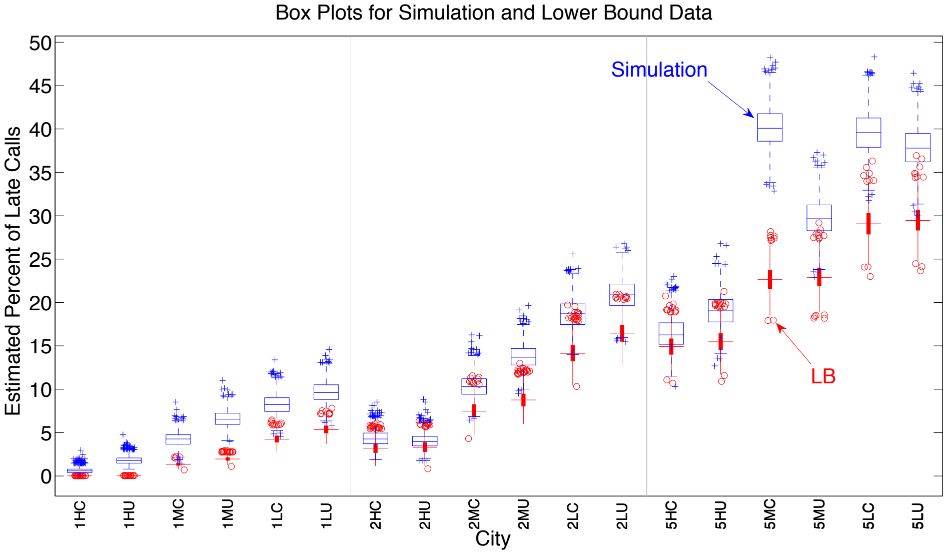

Figure 2: Box plots of the simulation and lower bound results for each fictional city.

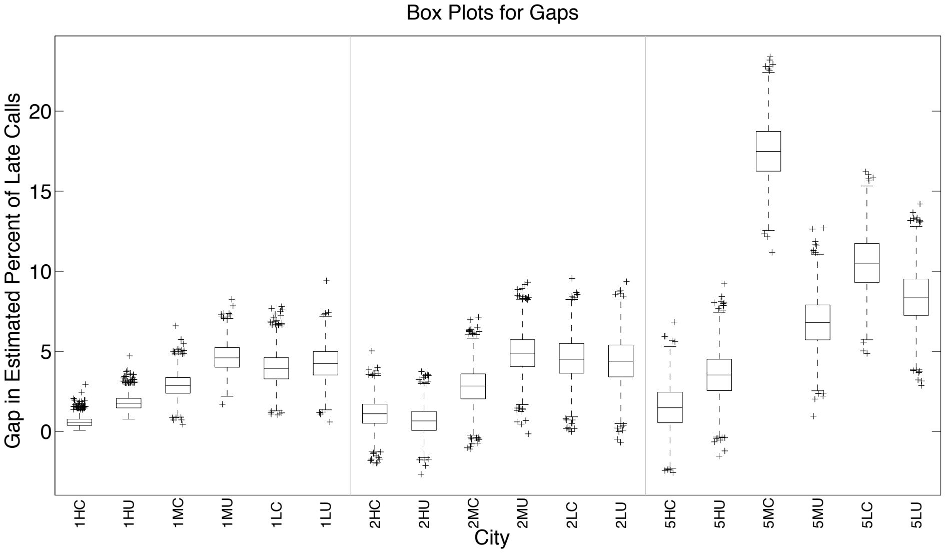

Figure 3: Box plots of the gap between the simulation and lower bound results for each fictional city

5 DISCUSSION

We now make a number of observations from the results of the previous section, and provide our proposed explanation for each of them.

1. As the number of modes increases, the lower bound and the results from the simulated policy both increase. We believe this happens because multi-modal distributions lead to the breaking down of cooperation between ambulances, because each peak needs to be covered somewhat separately from the others. Thus the economies of scale associated with larger queueing systems are lost.
2. As the number of modes increases the gap increases. We do not have an explanation of this observation.

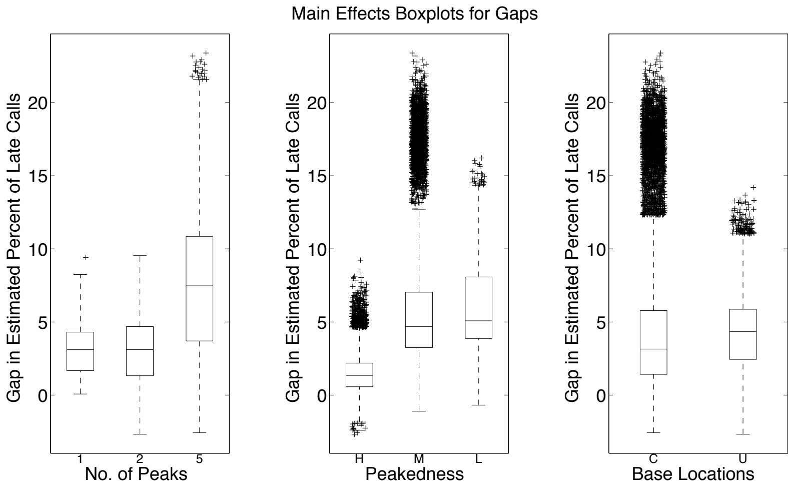

Figure 4: Box plots of the main effects for the gap between the simulation and lower bound results for each fictional city.

3. The lower bound is only modestly affected by the clustering of the bases. We believe this occurs because the lower bound is based on *covering* demand, so that as long as base locations are positioned so that large fractions of the demand can be covered with available ambulances, the fine details of the location of the ambulance are not important.
4. As the location distribution becomes less concentrated, the gap widens, except in the 5-modes case. Figure 1 shows that the least concentrated 5-mode case has a location distribution that is somewhat unimodal. Our results suggest that unimodal cases are easier than multimodal cases, so this may help to explain why the gap does not change monotonically with peakedness in the 5-node case. One might ask why this does not seem to happen with the 2-node cases. We suspect that this issue does not arise because even our least-peaked 2-mode case is still somewhat bimodal.
5. Clustering of bases to match demand improved (or at least did not harm) performance of the simulation policy, except in the 5-mode case, where it had a negative effect particularly for the "medium-peakedness" case. The simulation policy is heavily affected by clustering in this particular case, but the lower bound is not. We therefore hypothesize that the simulation policy needs improvement in this case, and view this as a key observation from this study that will spur future research.

As an aside, in all of our cases the demand can be covered with a full complement of ambulances. This is not typically the case in practice, because the borders of cities tend to contain lower-density populations. If we were to include such areas in our "cities" then we believe that both the lower bound and the simulation policy results would be shifted upwards, thus making the gap proportionately smaller, but disguising the effects we wish to uncover.

The effects we have seen above do not appear to be due to different utilizations of the ambulances in the different cases. Indeed, we estimated the utilizations in all cases to an accuracy of approximately one decimal place, and all estimated utilizations fell between 34% and 39.8%, and furthermore, all estimated utilizations in the 5-mode cases fell between 37.8% and 39.9%.

These observations have led to several conjectures that we hope will lead to improvements in our deployment policies. For example, in the redeployment policies of Maxwell, Henderson, and Topaloglu (2011) the value function approximation relies on partitioning demand between ambulance bases, and cooperation between bases is ignored. However, as the system becomes busy, the partition is not an accurate model of operations. We now conjecture that the partition should not be relative to the bases, but rather to the locations of the available ambulances. It is not yet clear how to implement this change in an ADP setting, but that is a topic of current research.

ACKNOWLEDGMENTS

This work was supported, in part, by National Science Foundation grants CMMI-0758441 and CMMI-0800688.

REFERENCES

Alanis, R., A. Ingolfsson, and B. Kolfal. 2012. “A Markov Chain Model for an EMS System with Repositioning”. *Production and Operations Management*. To appear.

Andersson, T. 2005. *Decision support tools for dynamic fleet management*. Ph. D. thesis, Department of Science and Technology, Linkoepings Universitet, Norrkoeping, Sweden.

Andersson, T., and P. Vaerband. 2007. “Decision support tools for ambulance dispatch and relocation”. *Journal of the Operational Research Society* 58:195–201.

Bell, C. E., and D. Allen. 1969. “Optimal planning of an emergency ambulance service”. *Journal of Socio-Economic Planning Science* 3:95–101.

Berman, O. 1981a, May. “Dynamic repositioning of indistinguishable service units on transportation networks”. *Transportation Science* 15 (2): 115–136.

Berman, O. 1981b. “Repositioning of distinguishable urban service units on networks”. *Computers and Operations Research* 8:105–118.

Berman, O. 1981c, March. “Repositioning of two distinguishable service vehicles on networks”. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-11 (3): 187–193.

Brotcorne, L., G. Laporte, and F. Semet. 2003. “Ambulance location and relocation models”. *European Journal of Operational Research* 147:451–463.

Church, R., and C. ReVelle. 1974. “The maximal covering location problem”. *Papers of the Regional Science Association* 32:101–108.

Daskin, M. 1983. “A maximal expected covering location model: formulation, properties, and heuristic solution”. *Transportation Science* 17:48–70.

Gendreau, M., G. Laporte, and S. Semet. 2001. “A dynamic model and parallel tabu search heuristic for real time ambulance relocation”. *Parallel Computing* 27:1641–1653.

Gendreau, M., G. Laporte, and S. Semet. 2006. “The maximal expected coverage relocation problem for emergency vehicles”. *Journal of the Operational Research Society* 57:22–28.

Maxwell, M. S., S. G. Henderson, and H. Topaloglu. 2011. “Tuning approximate dynamic programming policies for ambulance redeployment via direct search”. *Stochastic Systems* Submitted.

Maxwell, M. S., M. Restrepo, S. G. Henderson, and H. Topaloglu. 2010. “Approximate Dynamic Programming for Ambulance Redeployment”. *INFORMS Journal on Computing* 22 (2): 266–281.

Maxwell, M. S., C. Tong, S. G. Henderson, and H. Topaloglu. 2012. “Bounds on the performance of ambulance redeployment policies”. Submitted.

Nair, R., and E. Miller-Hooks. 2009. “Evaluation of Relocation Strategies for Emergency Medical Service Vehicles”. *Transportation Research Record* 2137:63–73.

Powell, W. B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Second ed. Hoboken, NJ: John Wiley & Sons.

Rajagopalan, H. K., C. Saydam, and J. Xiao. 2008. “A multiperiod set covering location model for dynamic redeployment of ambulances”. *Computers & Operations Research* 35:814–826.

Richards, D. P. 2007. “Optimised ambulance redeployment strategies”. Master’s thesis, Department of Engineering Science, University of Auckland, Auckland, New Zealand.

Schmid, V. 2012. “Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming”. *European Journal of Operational Research* 219:611–621.

Schmid, V., and K. F. Doerner. 2010. "Ambulance location and relocation problems with time-dependent travel times". *European Journal of Operational Research* 207:1293–1303.

Swersey, A. J. 1994. "The deployment of police, fire, and emergency medical units". In *Operations Research and the Public Sector*, edited by S. M. Pollock, M. H. Rothkopf, and A. Barnett. Amsterdam: North Holland.

Zhang, L. 2010, November. "Optimisation of Small-Scale Ambulance Move-up". In *Proceedings of the 45th Annual Conference of the Operations Research Society of New Zealand*, edited by M. Ehrgott and A. Mason, 150–159.

Zhang, L., A. Mason, and A. Philpott. 2010. "Optimization of a Single Ambulance Move up". Technical report, University of Auckland Faculty of Engineering.

AUTHOR BIOGRAPHIES

ERIC CAO NI is a Ph.D. student in the School of Operations Research and Information Engineering at Cornell University. He received a B.Eng. in Industrial and Systems Engineering and a B.Soc.Sci. in Economics from the National University of Singapore in 2010. His research interests include simulation optimization, emergency services and financial engineering.

SUSAN R. HUNTER is a postdoctoral associate in the School of Operations Research and Information Engineering at Cornell University. Her research interests include Monte Carlo methods and simulation optimization. Her email address is hunter@cornell.edu, and her webpage is http://people.orie.cornell.edu/srh227/.

SHANE G. HENDERSON is a professor in the School of Operations Research and Information Engineering at Cornell University. He holds a B.Sc. (Hons) from the University of Auckland and an M.S. (Statistics) and Ph.D. (Operations Research) from Stanford University. His research interests include discrete-event simulation and simulation optimization, and he has worked for some time with emergency services. He is the current chair of the INFORMS Applied Probability Society, and an associate editor for both *Management Science* and *Stochastic Systems*. He co-edited the Proceedings of the 2007 Winter Simulation Conference. His web page is http://people.orie.cornell.edu/∼shane.

HUSEYIN TOPALOGLU is an associate professor in the School of Operations Research and Information Engineering at Cornell University. He holds a B.Sc. in Industrial Engineering from Bogazici University in Turkey, and a Ph.D. in Operations Research and Financial Engineering from Princeton University. His research interests include stochastic programming and approximate dynamic programming with applications in transportation logistics, revenue management and supply chain management. He teaches courses on dynamic programming, simulation modeling, systems engineering and revenue management. His webpage is http://people.orie.cornell.edu/∼huseyin.