Emergency Medical Service Allocation in Response to Large Scale Events [1]

Gabriel Zayas-Cabán[2]

Center for Applied Mathematics

657 Rhodes Hall

Cornell University

Ithaca, NY 14850-2488

Mark E. Lewis[3]

Operations Research and Information Engineering

226 Rhodes Hall

Cornell University

Ithaca, NY 14853

Matthew Olson[4] Samuel Schmitz [5]

MITRE Corporation

7515 Colshire Drive

McLean, VA 22102-7539

December 21, 2012

[2] gz63@cornell.edu
[3] mark.lewis@cornell.edu
[4] meolson@mitre.org
[5] sschmitz@mitre.org

Abstract

In the event of a catastrophic or large scale event, demand for Emergency Medical Service (EMS) vehicles will almost certainly overwhelm the available supply. In such cases, it is necessary for cities to request aid (in the form of added capacity) from neighboring municipalities in order to bring the affected region back to its day-to-day levels of operation. In particular, we consider a region consisting of several cities, where each city is in charge of managing its own EMS vehicles. We propose that a centralized or statewide decision-maker coordinate the temporary transfer of resources (EMS vehicles) from cities in the unaffected region into the cities in the affected region. The control of each city's EMS vehicles is modeled as a multi-server queueing system and classical results are used to estimate the number of vehicles available at each city. We then develop a deterministic resource allocation model to guide the allocation of available vehicles from the donor area into the affected one and a clearing system model to dynamically control the added resources.

As the dimension of the problem is large, a heuristic we call the *buddy system* is proposed where cities are paired to form city groups. Within the city groups the clearing system model is solved by Markov decision processes. The performance of our heuristic is compared to several other reasonable heuristics via a detailed numerical study. Results show that the buddy system exhibits significant cost and time savings, and is generally robust to varying parameters.

1 Introduction

After a large scale or catastrophic event it may be the case that no amount of savvy on the part of local emergency medical service (EMS) dispatchers can direct enough resources to alleviate the problem in a timely fashion. In this case other cities in the region may be asked to lend some of their resources until the original city has recovered from the event. The problem can be viewed from several lenses.

- From the perspective of the lending municipality,
  - How many vehicles are available to lend?
  - How long will it be before they are returned?
- From the perspective of the affected region, how should the extra capacity be allocated?

In this paper, we provide a systematic approach for resource allocation when the demand for EMS vehicles exceeds what is typically experienced. Instead of suggesting that the two (or more) city managers/dispatchers make decisions without coordination, we recommend that a centralized or statewide administrator advise several municipalities to send some of their EMS vehicles to the affected region. These vehicles would then be allocated within the affected region until a return to normalcy is achieved. We note that except under special circumstances, for example in situations where the federal government takes over governance of a particular area, this is not the way large scale emergencies are handled. Thus, our research examines the potential of a centrally controlled approach and how this approach should be implemented.

To complete the analysis, prior to the event, we assume that the control of each city's EMS vehicles can be modeled by a multi-server queueing system. By relaxing the measure of quality of service slightly, we estimate the number of vehicles available for lending to other cities. We then provide a heuristic for static allocation from lending cities to the affected region and another dynamic allocation heuristic for allocation within the affected region. Taken together these questions constitute a macroscopic view of resource allocation from outside the affected region and a microscopic dynamic control problem from within.

After the modeling is complete, it becomes clear that the dimension of the state and action spaces are too large to solve the problem in reasonable time. We propose a heuristic we call the "buddy system" where cities or municipalities are paired to form city groups. It is left to the decision-maker to decide how these groups are formed (for example by location or need). This reduces the size of the problem and allows us to solve a deterministic resource allocation problem for initially allocating vehicles to city groups. Within the city groups a clearing system queueing model is solved via Markov decision processes. Our heuristic is tested by varying several parameters in an extensive numerical study.

The rest of the paper is organized as follows: Section 2 contains a brief survey of the literature. Section 3 contains a detailed description of how a decision-maker might decide how many vehicles are available from the donor region and how they might be allocated to the affected region. We present the numerical analysis in Section 4 and conclude the paper in Section 5. A dictionary of symbols and variables is in Appendix A.

2 Literature Review

Early work on emergency medical response systems is too plentiful to provide a complete review here. Instead we review only the most recent work that closely relates to the current study. First, we discuss some of the work that considers day-to-day allocation of EMS since some of this research can either serve as a baseline or provide insights into our problem. Secondly, we consider some of the closely related models considering response to large scale events. In each case, the work considers particular parts of the response (locally), while we are considering a larger response as might be undertaken by a statewide controller. The reader interested in the early work should consult the survey papers by Brotcorne et al. [5] and Goldberg [8].

There are several ambulance location models to deal with day-to-day operations. Most often the goal is to design emergency response systems to cover as many calls as possible while maintaining a target average waiting time (see for example [13] and the references therein). Ingolfsson [11] et al. propose a location model that minimizes the number of ambulances needed to guarantee a certain fraction of the calls is reached within a given time. The authors allow response time to be composed of a random delay (prior to travel to the scene) plus a random travel time. Gerolimisnis et al. [7] develop a spatial queueing model for locating emergency vehicles on urban networks. In addition, they incorporate varying service rates depending on incident characteristics. The model is applied using real data for locating freeway service patrol vehicles. In our work, we capture costs for relocating from one region to the next (as opposed to modeling spacial concerns explicitly), but assume that within the affected region the travel costs are negligible. Each case above is significantly different than the current work since we assume that the goal after a large scale event is to provide *best effort* service to return the system to normalcy. We also include a deterministic optimization component that answers the question of regional (pre-)allocations.

One of the major issues with modeling stochastic behavior in EMS vehicle routing is scalability. Maxwell et al. [13] use approximate dynamic programming techniques to approximate optimal ambulance redeployment. That is, once ambulances have been deployed, they may be reassigned to a new home base consistent with current network characteristics. This is somewhat similar to the current work, since we allow EMS vehicles to be reassigned within the affected region. Of course our objective is different. Schmid and Doerner [16] develop a multi-period version of the ambulance location problem that takes into account time-varying coverage areas and allows vehicles to be repositioned in order to maintain a certain coverage standard throughout the planning horizon. It is formulated as a mixed integer program that tries to optimize coverage at various points in time simultaneously. Each of these studies consider the problem on a day-to-day basis and differ significantly from the way we handle the allocation component of the assignment problem. Our approach is to use a hybrid deterministic optimization component on the macro-scale and a heuristic plus dynamic programming component on the micro-scale.

While all of the work mentioned above is moving toward implementation, a few others stand out as handling particular instances of day-to-day operations (whereas we seek a general framework after a large scale event). Doerner et al. [6] propose and implement solution heuristics for an ambulance location model, namely the double coverage ambulance location problem, in Austria. Moroshi [14] compares several location models using actual patient call data from Tokyo metropolitan area in order to improve ambulance service. Iannoni et al. [9] propose methods to optimize the configuration and operation of emergency medical systems on highways and is specifically applied to highway systems in Brazil. Iannoni et al. [10] use a hypercube queueing model (see the surveys cited above for a description of this model) to address location and districting.

Propelled by the events of hurricane Katrina, the attacks on September 11, and the looming possibility of a pandemic, work to respond to large scale events is more recent than some of the models described above. For example, Jain and Mclean [12] propose a framework with which to integrate modeling and simulation for emergency response while Aaby et al. [1, 2] used simulation (among several other operations research techniques) to address vaccine dispensing. Banomyong and Sopadang [3] developed a Monte-Carlo simulation model to improve emergency logistics in current response models while Bjarnason et al. [4] used simulation-based optimization for resource placement and emergency response. In each case above simulation is used along with either analysis (to see how well a proposed policy works) or deterministic optimization to try to improve on a policy online. In the current work, the assumption is that the decision-maker has access to service rate information and can develop relative weights to assess the importance of assigning vehicles to different locations. To our knowledge none of this work considers the assignment and dynamic allocation of resources after a large scale event and certainly does not attempt to provide a general framework from which to begin this allocation.

3 Modeling Approach

Before explaining our approach we make several remarks. First, we have generically defined a large scale event as one where the affected region might call for (or simply require) significant aid in the form of extra capacity. While we do not formally define these events, we mention that clearly catastrophic events like hurricanes, major floods or terrorist attacks would fall in this category. Other events that occur more frequently may also be categorized as large scale events (depending perhaps on the resources of the affected region) like large fires, power outages, or some large automobile accidents. Second, we will not formally define what is meant by a "return to normalcy". Instead, in our modeling approach we assume that there is a zero state. This depends heavily on the type of event and the municipality or region under consideration. What is crucial is that the decision-maker has knowledge of (or a good estimate of) the number of jobs or outstanding calls that need to be serviced at each location before the system is back to the nominal state. Finally, there is the obvious assumption that the central decision-maker has final authority over who gets sent to each municipality. Currently, in practice decisions are made by local authorities in a decentralized manner. Given that it is quite often the case that some municipalities are too generous we feel that this is a reasonable assumption. Indeed, one of the recommendations we would make is that a centralized control can be beneficial.

3.1 Preprocessing

After a catastrophic event, a decision-maker needs to assess which cities should act as donors and which should be candidates for added capacity. For the donor cities, the nominal quality of service measures are slightly relaxed in preparation for lost capacity. For the affected region, the same measures are relaxed so that capacity can be allocated to the most needy portion of the region. The lost (or actually lent) capacity from the donor region becomes added temporary capacity to the cities in the affected region. In this section we propose a method for estimating the amount of available capacity. Indeed, this is simply one method. Others could be explored and used as inputs to the allocation steps in Section 3.2.

Suppose there is a set of N nodes (in total), where each node represents the operation of an (independent) EMS system for a particular city or municipality. Assume that node k has a fixed and known capacity n_k for $k = 1, 2, \ldots, N$. After the large scale event, we will assume the time to complete a service at a particular node k is exponential, but for now we allow the service times to follow a general distribution with rate μ_k. We make the assumption that under nominal conditions the time between EMS calls at a particular municipality are exponential with rate λ_k. Note that by making these assumptions we are ignoring spatial considerations (like those that might guide which server is assigned to each job) at a particular node. Since demands that are met leave the system, the above description is that of N parallel queues each of which acts as an $M/G/n_k/\infty$ queue.

3.1.1 General Service Times

When the time to complete calls is also exponential we provide an explicit estimate of the number of available vehicles based on the average waiting time (see below). Unfortunately, there are no closed form expressions for the average waiting time for the $M/G/n_k/\infty$ system. We suggest a systematic method to estimate the number of available vehicles that follows closely the analysis suggested by [15].

Fix a node k. To guarantee stability assume that $\frac{\lambda_k}{n_k \mu_k} < 1$. Instead of the average waiting time as a quality of service measure, we make the tacit assumption that a vehicle that is immediately assigned to a call can complete the call in a timely fashion. If this holds, then only those calls that find all vehicles currently busy are made to wait longer than usual. As an approximation for the probability with which calls made to dispatch take longer than usual we use the probability that a call is blocked in a system without buffer space. In this case, calls that do not find a vehicle immediately available are lost. Computing the blocking probability is done by the well-known *Erlang* blocking formula for an $M/G/n_k/n_k$ queueing system,

$$\pi_{n_k} = \frac{\frac{\left(\frac{\lambda_k}{\mu_k}\right)^{n_k}}{n_k!}}{\sum_{i=0}^{n_k} \frac{\left(\frac{\lambda_k}{\mu_k}\right)^{i}}{i!}}.$$

During normal day-to-day operations the local dispatcher sets the targeted quality of service (QoS) as the proportion of "blocked" calls, say $\tilde{\pi}_{n_k}$. After a large scale event at a nearby municipality, this measure is relaxed to $\alpha > \tilde{\pi}_{n_k}$ (for QoS level α). Since the blocking probability is non-increasing in n_k, exhaustive search can then be used to obtain the minimum $n'_k < n_k$ such that $\pi_{n'_k} > \alpha$. The number of vehicles available for reallocation is $n_k - n'_k$. Repeating the argument for each city in the donor region yields the total number of vehicles available for lending.

Since the actual system calls are allowed to wait for a vehicle to become available, this is an underestimate of the actual blocking probability; the estimates are aggressive. On the other hand, since the same algorithm is used to estimate the blocking probability before and after the large scale event, the estimates are consistent.

3.1.2 Exponential Service Times

Continue to consider a single node in the network with a Poisson arrival process. Suppose now, however, that the time to complete each call to dispatch can be modeled as an exponential random variable with rate μ_k. Under this assumption, the system at a fixed node can be modeled as an $M/M/n_k/\infty$ queueing system.

Let $\rho = \frac{\lambda_k}{n_k \mu_k}$ and $r = \frac{\lambda_k}{\mu_k}$. It is well-known that for this system the long-run average number of jobs waiting for a vehicle to be assigned and the average number of total jobs in the system can be computed, respectively by

$$L_q^{n_k} = C \frac{r^{n_k}(\rho)}{n_k!(1-\rho)^2} \qquad\qquad L^{n_k} = r + C \frac{r^{n_k}(\rho)}{n_k!(1-\rho)^2},$$

where C is the normalizing constant such that $C^{-1} = \sum_{n=0}^{n_k} \frac{\lambda_k^n}{n!\mu_k^n} + \sum_{n=n_k+1}^{\infty} \frac{\lambda_k^n}{n_k! n_k^{n-n_k} \mu_k^n}$. Using *Little's Law*, $L = \lambda W$ in each case yields the average wait before being assigned a vehicle is $W_q^{n_k} = \frac{L_q^{n_k}}{\lambda_k}$ and the average wait until service is complete is $W^{n_k} = \frac{L^{n_k}}{\lambda_k}$. In an analogous manner to the previous section, suppose the quality of service measure is the average waiting time. In the normal day-to-day operations, the average wait is W^{n_k}, but after a large scale event, the decision-maker decides to relax this to $\beta > W^{n_k}$. Since the average wait is non-increasing in n_k, exhaustive search yields the minimum n_k' such that $W^{n_k'} < \beta$. The difference $n_k' - n_k$ represents the capacity available to be reallocated.

Remark. *After this analysis is complete, the decision-maker has a vector of "extra" capacity at each city. For the donor cities, this is the capacity that can be reallocated to the affected region, and for cities in the affected region this is capacity that can be used to help return the city to its nominal state.*

3.2 Allocation from Donor Cities to the Affected Region

As has been alluded to, before beginning the allocation a decision-maker needs to decide which city(ies) will be designated as *donor* cities and which cities are within the *affected* region. The decision-maker will then take the following steps.

1. For each of the cities use one of the results from Section 3.1.1 (if the nominal case

assumes general services) or Section 3.1.2 (if the nominal case assumes exponential services)

(a) For the cities in the affected region (the number of which is denoted L), the available vehicles are put into the vector $\vec{\ell}$. That is, the available vehicles at city k are ℓ_k, $k = 1, 2, \ldots, L$.

(b) The total number of vehicles available from all donor cities is obtained and denoted N_D.

2. Suppose $v_{N_k}(\vec{m}, \vec{\nu})$ ($w_{N_k}(\vec{m}, \vec{\nu})$) denotes the total cost (time) of returning the system to normalcy starting with the initial vector of $\vec{m} = \{m_1, m_2, \ldots, m_L\}$ calls above normalcy in the affected region and the current allocation captured in the vector $\vec{\nu} = \{\nu_1, \nu_2, \ldots, \nu_L\}$.

   (a) Solve the following optimization problem (MP) if objective is to minimize the total cost of returning the system to normalcy:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{L} v_{N_k}(\vec{m}, \vec{\nu}) \\ \text{subject to} \quad & \sum_{k=1}^{L} N_k = N_D, \\ & N_k \geq 0. \end{aligned} \tag{1}$$

   (b) Solve the following optimization problem (MP') if objective is to minimize the time of returning the system to normalcy:

$$\begin{aligned} \text{minimize} \quad & \max_{k=1,\ldots,L} w_{N_k}(\vec{m}, \vec{\nu}) \\ \text{subject to} \quad & \sum_{k=1}^{L} N_k = N_D, \\ & N_k \geq 0. \end{aligned} \tag{2}$$

Note that constraints can be easily added to (MP) ((MP')). For example, if the decision-maker decides that city k must have M_k vehicles from the excess capacity, adding the constraint $N_k = M_k$ achieves the goal. Suppose that there is a budget, b and that sending vehicles to the location associated with city k is c_k. The constraint $\sum_{k=1}^{\ell} c_k N_k \leq b$ can be added.

The difficulty in implementing this algorithm is in the assumption that $v_{N_k}(\vec{m}, \vec{\nu})$ $(w_{N_k}(\vec{m}, \vec{\nu}))$ can be obtained quickly and easily. In the next section, we provide a general case that quickly becomes intractable due to the curse of dimensionality. We then show how the computational issue can be alleviated, but at the cost of an oversimplification. Our heuristic is a compromise between these two extremes.

3.3 Allocation within the Affected Region

Consider the reallocation of available vehicles between the cities in the affected region. Mathematically, our decision-making scenario is modeled as a clearing system. That is, each arrival rate is set to zero ($\lambda_k = 0$ for all $k = 1, 2, \ldots, N$), the nominal state is referred to as the zero state and we seek the control policy that empties or "clears" the system at minimal cost. To model the decision-maker's need to prioritize certain areas more than others, the system is charged holding costs h_k per unit time for each job waiting at station k. From the standpoint of the decision-maker, this is not a value judgement on the importance of meeting one service call or another, but it might (for example) capture the severity of various service calls. We assume that the time to move vehicles from one locale to the next is negligible. This is done for tractability and may not directly reflect reality. On the other hand, we model the cost of assigning vehicles from node k to node m as $c_{k,m}$, or more generally, define the function $c(\cdot)$ to model the action of assigning vehicles from one location to another.

Another concern of the decision-makers may be that within the affected region, vehicles are moved around too often. Thus far, the cost function is a function of the allocation to be used in the coming decision epoch, and not of the current decision. Recall, $\vec{\nu} = \{\nu_1, \nu_2, \ldots, \nu_L\}$ is the current allocation of vehicles to cities and let $d(\vec{\nu}, a)$ be the cost of changing the allocation from $\vec{\nu}$ to a. As an example of a possible function d, one might consider the total number of vehicles moved by the allocation $\vec{n}$,

$$d(\vec{\nu}, \vec{n}) = \sum_{k=1}^{L} |\nu_k - n_k|.$$

Thus, once the donor city vehicles have been assigned to the affected region the decision-making scenario is as follows: The decision-maker views the state of the system in the state space $\mathbb{X} := \{(\vec{m}, \vec{\nu}) | \vec{m} \in (\mathbb{Z}^+)^L, \sum_{k=1}^{L} \nu_k = M\}$, where M is the total number of vehicles available for assignment. The vector $\vec{m} = \{m_1, m_2, \ldots, m_L\}$ is the current number of jobs (patients for example) at each municipality. After each service completion, the decision maker chooses from the set of available (re-)assignments; a subset of $A(M) := \{(n_1, n_2, \ldots, n_L) | \sum_{k=1}^{L} n_k = M\}$. Note that the model is robust enough to restrict the set of available assignments to reflect the decision-maker's desire to model reality. For example, we may only allow reassignment of vehicles from one subset of the cities to another. In (MP) we seek a control policy that minimizes the cost to empty the system while in (MP') we find the policy that returns the system to normalcy in the minimum time.

3.3.1 Independent Allocation Heuristic

To search for an optimal control we begin by assuming that cities in the affected region operate independently. That is, once the allocation from the donor region to each city in the affected region in completed, vehicles stay at their assigned location until the affected city has returned to normal after which they return to their respective donors. In this case there is no need to keep track of how many vehicles are currently assigned to each city in the state space. The total cost for city k is computed

$$v_{N_k}(i) = c(N_k) + \frac{i h_k}{\min\{N_k + \ell_k, i\}\mu_k} + v_{N_k}(i-1). \tag{3}$$

Analogously, the total time until city k returns to normal is computed

$$w_{N_k}(i) = \frac{1}{\min\{N_k + \ell_k, i\}\mu_k} + w_{N_k}(i-1). \tag{4}$$

The recursions above are easy to solve making the allocation problems in (1) and (2) also simple to solve. This simplified version of the problem has the advantage that the decision-maker does not need to be in constant communication with each municipality (and continually update the state space). The down side is that this is quite the rudimentary heuristic. After a large scale event, it is possible that within the affected region EMS vehicles may be reallocated dynamically to alleviate difficulties in more highly affected areas. The next section discusses the adjustment to the model above to capture these decisions.

3.3.2 The L-station Clearing System

In this section we present the other extreme when compared to the independent allocation of Section 3.3.1. Suppose that we allow the decision-maker to reallocate vehicles within the affected region. Assume that the cost of assigning vehicles depends on the city the vehicles are going and not on the initial donor city. Recall the total number of vehicles available for assignment is M and that the set of available assignments is then $A(M) := \{(n_1, n_2, \ldots, n_L) | \sum_{k=1}^{L} n_k = M\}$. Let e_i be the vector of all zeros except for a 1 in the i^{th} element; $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$. Including the cost of the current allocation and the (smoothing) cost to change allocations the cost optimality equations are now

$$v(\vec{m}, \vec{\nu}) = \min_{\vec{n} \in A(M)} \Big\{ c(\vec{n}) + d(\vec{\nu}, \vec{n}) + \sum_{i=1}^{L} \frac{h_i m_i}{\sum_{k=1}^{L} \min\{n_k + \ell_k, m_k\} \mu_k} + \sum_{i=1}^{L} \frac{\min\{n_i + \ell_i, m_i\} \mu_i v(\vec{m} - e_i, \vec{n})}{\sum_{k=1}^{L} \min\{n_k + \ell_k, m_k\} \mu_k} \Big\}. \tag{5}$$

while the time optimality equations are

$$w(\vec{m}, \vec{\nu}) = \min_{\vec{n} \in A(M)} \Big\{ \frac{1}{\sum_{k=1}^{L} \min\{n_k + \ell_k, m_k\} \mu_k} + \sum_{i=1}^{L} \frac{\min\{n_i + \ell_i, m_i\} \mu_i w(\vec{m} - e_i, \vec{n})}{\sum_{k=1}^{L} \min\{n_k + \ell_k, m_k\} \mu_k} \Big\}. \tag{6}$$

The value function $v(\vec{m}, \vec{\nu})$ $(w(\vec{m}, \vec{\nu}))$ must be computed for each element of the state space. The difficulty (of course) in computing v (w) is the fact that $(\vec{m}, \vec{\nu})$ could be high dimensional. In each state, the set of actions is the number of ways to allocate M vehicles to L cities. Using the (multi-choose) identity,

$$\left(\!\!\binom{L}{M}\!\!\right) = \binom{L+M-1}{M},$$

shows that the number of actions becomes intractable as the number of vehicles and/or cities gets large. Moreover, even when the recursion can be computed, implementation requires that a centralized decision-maker monitor each city and make decisions dynamically.

3.3.3 The Buddy System

Unfortunately, the recursion in (5) ((6)) is computationally infeasible as the number of affected cities grows. It may also suffer from a lack of implementability. Suppose we can solve the recursion in (5) ((6)) for city **groups** of size B.

1. Divide the cities into $r := \lceil \frac{L}{B} \rceil$ groups.
2. Parameterize the optimal cost when a city cluster has N_k movable vehicles added as a single node with zero vehicles as $v_{N_k}(\vec{m}, \vec{\nu})$ $(w_{N_k}(\vec{m}, \vec{\nu}))$.
3. (a) If the objective is to minimize the total cost of returning the system to normalcy, then solve the following optimization problem (MP$_r$):

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{r} v_{N_k}(\vec{m}, \vec{\nu}) \\ \text{subject to} \quad & \sum_{k=1}^{r} N_k = N_D, \qquad\qquad N_k \geq 0. \end{aligned}$$

   (b) If the objective is to minimize the time of returning the system to normalcy, then solve the following optimization problem (MP'_r):

$$\begin{aligned} \text{minimize} \quad & \max_{k=1,\ldots,r} w_{N_k}(\vec{m},\vec{\nu}) \\ \text{subject to} \quad & \sum_{k=1}^{r} N_k = N_D, \qquad\qquad N_k \geq 0. \end{aligned}$$

4. Allocate the optimal N_k to city cluster k and use $v_{N_k}(\vec{m},\vec{\nu})$ $(w_{N_k}(\vec{m},\vec{\nu}))$ to allocate the vehicles in that cluster dynamically.

It should clear that the models of Sections 3.3.1 and 3.3.2 are the cases of this heuristic with $B = 1$ and $B = L$, respectively. As an alternative, suppose instead that the decision-maker has paired cities in the affected region making groups of size 2. That is to say, that each city has a "buddy" city which it will borrow from or lend to depending on the current number of jobs to be completed at the city. The above description can still be used to discern how much excess capacity is added to each 2-city sub group in the affected region. Another benefit in using the buddy system is that the central decision-maker really only needs to know the state of the system immediately after the large scale event. Once the allocation has been made, (s)he can pass control to the city group for local communication.

4 Numerical Analysis

In this section we discuss how well the buddy system described in Section 3.3.3 performs when compared to several heuristics. Accordingly, we calculated the percentage savings in expected cost and time using the optimization formulation (MP_r) in Section 3.3.3 coupled with the buddy system with $B = 2$ when compared to four alternative heuristics. In addition, we calculated the percentage savings in expected time using the optimization formulation (MP'_r) in Section 3.3.3 coupled with the buddy system with $B = 2$ when compared to the same four alternative heuristics. The heuristics are described below:

Heuristic 1: The centralized dispatcher allocates available vehicles evenly between the cities. Once vehicles are allocated, they remain in the city until system has returned to the nominal state.

- This heuristic reflects the centralized dispatcher's preference for equity. Each city is treated the same without regard to the number of incidences, processing rates or an assessment of variable severity.

Heuristic 2: Cities are ordered such that the vector of initial jobs above normalcy is such that $m_1 \geq m_2 \geq \cdots \geq m_L$. The centralized dispatcher allocates $a_1 := \min\{N_D, m_1\}$ to city 1, $a_2 := \min\{N_D - a_1, m_2\}$ to city 2, etc., until all available vehicles have been exhausted. Vehicles remain in each city until all jobs are completed.

- The cities are ordered from the one with the most jobs to the one with the least. The decision-maker allocates enough vehicles so that as few as possible jobs in city 1 are made to wait. If there is a surplus, the same is done for city 2, etc.
- In this heuristic, the centralized dispatcher prioritizes cities with the most emergencies.

Heuristic 3: The centralized dispatcher allocates available vehicles evenly amongst city groups of size 2. Once available vehicles have been evenly distributed, vehicles within each city group are allocated dynamically according to the 2-station clearing system.

- This heuristic is similar to the first with the exception that once the macroscopic allocation is made, the microscopic allocation is dynamic.
- We also remark that Heuristic 1 is **not** always a sub-case of Heuristic 3. Indeed, if there 4 total cities and 10 available vehicles, Heuristic 1 divides the first 8 vehicles evenly $(2, 2, 2, 2)$ and then starts from city 1, then city 2, etc. until all vehicles are exhausted (ending with $(3, 3, 2, 2)$ in this case). Heuristic 3 divides the vehicles into two city groups of size 2, each having 5 vehicles. Assuming that cities 1 and 2 form a group and 3 and 4 form a group, the assignment $(3, 3, 2, 2)$ is not achievable by Heuristic 3.

Heuristic 4: The centralized dispatcher determines the city (or cities) with the highest number of jobs. If there is (only) one, allocate a vehicle to that city. If not, among the cities having the highest number of jobs, determine the city with the lowest processing rate and allocate a vehicle there arbitrarily choosing one if there are several. Decrease the number of jobs by one for the recipient city and repeat this process (with the updated "number of jobs" vector) until the available vehicles have been exhausted.

- This heuristic reflects the centralized dispatcher's desire to provide aid to cities that are in most need; those cities with either the largest number of accidents or slowest processing rates.

There are several scenarios in which a policy may not allocate any vehicles to some cities. We have assumed that the vector of excess capacity within the affected region is $\ell = \{1, 1, \ldots, 1\}$. Thus no matter what the assignment of donor vehicles, work is still being completed at a minimum rate. Throughout the numerical analysis we assume the cost of moving vehicles between cities as well as the smoothing cost is zero. We list the parameters tested in Table 1 below.

Parameter Symbol	**Definition**
r	number of city groups
N_D	number of vehicles available for (re-)allocation
$i_{n,m}$	number of jobs in city n of city group m
$h_{n,m}$	per unit holding cost in city n of city group m
$\mu_{n,m}$	processing rate for city n of city group m

Table 1: List of Parameters

Two cases are considered. In Case I, there are 4 cities in the affected region with a total of 144 instances. In Case II there are 8 cities and a total of 300 different instances. The parameter values for each case are displayed in Tables 2a and 2b. We also remark that throughout the study in each case the policies are implemented in a system where cities are paired as in the buddy system. In this system, the buddy system uses the optimal control while for example Heuristic 1 splits the extra capacity evenly amongst the paired sub-groups. This is done for consistency and alleviates the subtle difference between the maximum of expectations and the expectation of maximums. The main

Parameters	**Values/Levels**
N_D	20, 50
$i_{n,m}(n = 1; m = 1, 2)$	8
$i_{n,m}(n = 2; m = 1, 2)$	8, 20, 50
$h_{1,1}, h_{2,1}, h_{1,2}$	2
$h_{2,2}$	2, 8
$\mu_{n,m}(n = 1; m = 1, 2)$	1
$\mu_{n,m}(n = 2; m = 1, 2)$	1, 5

(a) Case I: 4 cities ($r = 2$)

Parameters	**Values/Levels**
N_D	15
$i_{n,m}(n = 1, 2; m = 1, 2)$	8
$i_{1,3}, i_{2,3}, i_{1,4}, i_{2,4}$	16, 16, 24, 24
$h_{1,1}, h_{2,1}$	2
$h_{n,m}(n = 1, 2; m = 2, 3, 4)$	1, 8
$\mu_{1,1}, \mu_{2,1}, \mu_{1,3}, \mu_{2,3}$	1
$\mu_{n,m}(n = 1, 2; m = 2, 4)$	1, 5

(b) Case II: 8 cities ($r = 4$)

Table 2: List of Parameter Values

takeaways from the numerical study are as follows:

- The buddy system performs well in each case and does so consistently. However, since implementation of the system is not without cost one may consider one of the heuristics as an alternative.

- If the donor cities provide enough capacity to cover the number of jobs to be completed, the benefit of the buddy system is significantly reduced.

- There are two components of the decision process; the macroscopic allocation from donors to affected cities and the microscopic (potentially dynamic) allocation of vehicles within the region. Both components can have a significant effect on cost savings.

- Each of these observations are displayed in the 4 city example and confirmed in the 8 city example.

- Finally, we note that the buddy system also performs well under the minimal completion time criterion, even when the control is obtained by minimizing the total cost. So a decision-maker can prioritize more severely injured patients while still guaranteeing a rapid return to normalcy.

4.1 Case I: 4 cities ($r = 2$)

4.1.1 Expected Cost

Results for the comparison of the total expected cost between the buddy system and the 4 heuristics described above for Case I are summarized in Table 3. The percentage savings in expected cost from implementing the buddy system versus Heuristics 1-4 is nonnegative for all instances. In fact, the buddy system is significantly better than Heuristics 1, 2, and 4 in a large proportion of the cases. In each case, the percentage difference was more than 30% more than 35% of the time. Heuristic 3, which was on average more than 7% more costly than the buddy system was more than 10% more costly in 30% of the cases tested.

Compared to the buddy system, Heuristic 1 fared well in the instances where the number of available vehicles was at least the total number of jobs. Indeed when the initial number of jobs vector is $(8, 20, 8, 8)$ the average percent savings when N_D is 50 is only 3.4%. On the other hand, when the initial jobs vector is $(8, 20, 8, 20)$ or $(8, 20, 8, 50)$, with $N_D = 20$, the average cost savings is 32.1%. Heuristic 1 also fared well in those instances where all of the cities had the same number of jobs. Out of the remaining instances, Heuristic 1 fared well when there were high service rates and low holding costs corresponding to cities with a high job count. This stands to reason since the concern of the high job count is mitigated by the high service rates and lower costs.

Similarly, Heuristic 2 fared well in the instances when enough vehicles were available to cover all of the jobs. However, its performance worsened dramatically as the difference between the total number jobs and the number of available vehicles increased. In particular, when the initial vectors were $(8, 20, 8, 20)$ and $(8, 20, 8, 50)$ and $N_D = 20$ the average cost savings were 77.73% and 53.86%, respectively. In general, Heuristic 2 fared the poorest and we obtained significant savings from implementing our heuristic.

Heuristic 3 fared well in all of the same instances as Heuristic 1. It turns out that in these cases the macroscopic allocation decision is the same for the two heuristics. Heuristic 3 then allocates according to what is optimal while Heuristic 1 does no further allocation. Heuristic 3 also performed well when exactly two cities had either 20 or 50 jobs to complete, regardless of the number of vehicles available for reallocation. Upon examining the macroscopic allocation for Heuristic 3 we note that it did not differ significantly from that of our heuristic while the microscopic allocation for the two is in essence the same. When restricted to these cases, the average difference in the costs is 1.3%. The instances in which it performed poorly correspond to those in which there was only one city where the number of jobs was either 20 or 50 and the total number of jobs exceeded the number of available vehicles. This is because the macroscopic allocation decision for Heuristic 3 differs significantly from that of our heuristic; at the macroscopic level, Heuristic 3 does not take into consideration the severity of one city with high incidences while the buddy system does. When restricted to these cases the average cost difference is 18.37%.

We also note that Heuristic 3 fared better than Heuristic 1 in all instances. This suggests that significant benefits can be obtained by incorporating system dynamics at the microscopic decision-making level even when the macroscopic decision is egalitarian.

Lastly, Heuristic 4 fared well in the instances where enough vehicles were available to cover all of the jobs. In particular, when the initial number of jobs vector is either $(8, 8, 8, 8)$ or $(8, 20, 8, 8)$ the average percent savings when $N_D = 50$ is negligible. Furthermore, when the initial number of jobs vector is $(8, 8, 8, 8)$ the average percent savings when $N_D = 20$ is 3.1% and when the initial number of jobs vector is $(8, 20, 8, 20)$ the average percent savings when $N_D = 50$ is only 0.51%. If the processing rates and costs corresponding to cities with 20 or 50 jobs took on high and low values, respectively, then Heuristic 4 also performed well. It performed dramatically worse otherwise. This is due to the fact that in the remaining instances, when the initial number of jobs vector are $(8, 8, 8, 50)$, $(8, 20, 8, 50)$, and $(8, 50, 8, 50)$, or when it is $(8, 20, 8, 8)$ with $N_D = 20$, Heuristic 4 typically ignores the processing rates and always ignores the costs. In these cases the average cost savings using the buddy system is 34.2%.

4.1.2 Expected Completion Time for Expected Cost Optimal Policies

In this section, we use the policies derived using the buddy system under the minimal cost criterion (solving (MP)) and compare the results to Heuristics 1-4 under the expected completion time criterion. The results are summarized in Table 4. The expected completion time for Heuristic 1 is lower than or equal to the buddy system in those instances

Percentage Savings	**Heuristic 1**	**Heuristic 2**	**Heuristic 3**	**Heuristic 4**
<= 1%	10	24	53	32
Between 1% and 5%	21	2	22	10
Between 5% and 10%	16	4	26	5
Between 10% and 20%	15	6	30	20
Between 20% and 30%	25	9	10	26
Between 30% and 40%	31	7	3	17
Between 40% and 50%	18	8	0	9
Between 50% and 60%	7	23	0	13
> 60%	1	61	0	12
Average	23.66%	47.99%	7.13%	24.71%
Standard Deviation	17.12%	30.68%	8.40%	22.13%
Max	61.78%	95.17%	35.50%	84.28%
Min	0%	0%	0%	0%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 3: Percentage savings in expected cost. Case I: The Buddy System v. Heuristics 1-4

where the number of available vehicles is at least the total number of jobs. If we consider only the cases where the initial jobs vector is $(8, 20, 8, 8)$ and $N_D = 50$ the average times savings is 0.76%. When the total number of jobs exceeds the number of vehicles available, Heuristic 1 outperforms the buddy system when high processing rates and/or high cost corresponded to at least one city with a high number of jobs. For example, when the initial jobs vector is $(8, 20, 8, 50)$, $\vec{h} = (2, 2, 2, 8)$, and either $\vec{\mu} = (1, 1, 1, 5)$ or $\vec{\mu} = (1, 5, 1, 5)$ the average times savings is -6.24%. It also fared well in those instances where all the cities had the same number of jobs. This makes sense since the buddy system is taking into account different holding costs while the criterion does not call for such consideration. For instance, when the initial jobs vector is $(8, 8, 8, 8)$ and $N_D = 20$ the average times savings is 0.82%. For all other cases, the expected completion time for the buddy heuristic heuristic is lower, with an average times savings of 22.98%.

The expected completion time for Heuristic 2 is better than that of the buddy system when the number of vehicles available exceeds the total number of jobs. It also fared well in those instances where all the cities had the same number of jobs. Note that in the latter case, the initial allocation for Heuristic 2 is the same as that for Heuristic 1.

Lastly, Heuristic 2 fared well in the instances where $N_D = 50$, the initial jobs vector is $(8, 20, 8, 20)$, and $\vec{\mu} = (1, 1, 1, 5)$ with average times savings is 0.05%. Otherwise, the buddy system had a lower expected time of completion by an average of 57.08%.

As in the expected cost section, the instances where the expected completion time for Heuristic 3 is comparable or better than the buddy system correspond to the same instances where the expected completion time for Heuristic 1 also performs well. It also performed comparably to the buddy instances when the number of jobs exceeded the total number of vehicles available for reallocation, the initial jobs vector is one of $(8, 20, 8, 20)$, $(8, 20, 8, 50)$, or $(8, 50, 8, 50)$ and $\vec{\mu} = (1, 1, 1, 1)$ with average times savings is 1.52%. In the remaining cases, where the buddy system outperforms Heuristic 3, the average time savings is 8.1%.

Lastly, the expected completion time for Heuristic 4 is comparable to that of the buddy system in those instances where the number of available vehicles is at least the total number of jobs, when there is at least one high processing rates to offset the effects of high job count, and when the number of jobs in each city are the same. To be specific, these instances include the cases when the initial jobs vectors is $(8, 20, 8, 8)$ with $N_D = 50$ and when the initial jobs vectors are $(8, 50, 8, 8)$ and $(8, 20, 8, 20)$ with $\vec{\mu} = (1, 5, 1, 1)$ or $\vec{\mu} = (1, 5, 1, 5)$, and $N_D = 50$. In the former case, the average times savings is 0% and in the latter cases, it is 6.32%. In the remaining cases, where the buddy system outperforms Heuristic 4, the average time savings is 35.8%.

4.1.3 Expected Completion Time for ($\mathbf{MP}'_r$)

While the last section used the policies tuned to the cost criterion and showed that the buddy system did well when compared to the other heuristics, in this section we tune the buddy system to minimize the total expected completion time. Results for the expected completion time are summarized in Table 5. The percentage savings in expected time from implementing the minimax buddy system versus Heuristics 1-4 is nonnegative for all instances.

The expected completion time for Heuristic 1 is comparable to that of the minimax buddy system in those instances where the number of available vehicles is at least the total number of jobs. If we consider only the cases where the initial jobs vector is $(8, 20, 8, 8)$

Percentage Savings	Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
Between -30% and -20%	1	0	0	0
Between -20% and -10%	6	0	3	0
Between -10% and -5%	10	0	5	0
Between -5% and -1%	14	0	6	0
Between -1% and 0%	20	28	38	24
Between 0% and 1%	10	0	17	10
Between 1% and 5%	23	0	25	22
Between 5% and 10%	3	6	33	1
Between 10% and 20%	24	0	13	10
Between 20% and 30%	14	3	5	17
Between 30% and 40%	7	9	0	14
$> 40\%$	12	98	0	54
Average	9.25%	46.02%	3.39%	27.70%
Standard Deviation	16.00%	28.63%	6.93%	25.05%
Max	44.73%	90.52%	22.07%	78.76%
Min	-27.12%	-0.41%	-19.79%	-0.00%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 4: Percentage savings in expected completion time. Case I: The Buddy System v. Heuristics 1-4 (dynamic control computed using expected cost)

and $N_D = 50$ the average times savings is 0.01%. Out of the remaining instances, the expected completion time for Heuristic 1 is comparable to the minimax buddy system when high processing rates correspond to cities with a high number of jobs. For example, when the initial jobs vector is $(8, 20, 8, 50)$ and $\vec{\mu} = (1, 5, 1, 5)$ the average times savings is 3.49%. That is, when the rates offset the negative effects of a high jobs count. For all other cases, the average times savings is 26.15%.

The expected completion time for Heuristic 2 is comparable to that of the minimax buddy system only when the number of vehicles available exceeds the total number of jobs, when high processing rates correspond to cities with 20 or 50 jobs, and when the number of jobs at each city is the same. Otherwise, our heuristic had a lower expected time of completion by an average of 62.92%.

The expected completion time for Heuristic 3 is no greater than that of the minimax buddy system. This stands to reason since allocating vehicles according to what Heuristic 3 prescribes serves as a feasible allocation/solution to the minimax buddy system heuristic. Consequently, the expected completion time obtained from Heuristic 3 for any given instance is an upper bound to the expected completion time of the minimax buddy system heuristic.

Lastly, the expected completion time for Heuristic 4 is comparable to that of the buddy system only in those instances where the number of available vehicles was at least the total number of jobs or if at least one high processing rate corresponds to a city with a high job count. In these instances, the average times savings is 0.09%. Otherwise, our heuristic had a lower expected time of completion by an average of 43.47%.

Percentage Savings	**Heuristic 1**	**Heuristic 2**	**Heuristic 3**	**Heuristic 4**
Between 0% and 1%	26	20	44	22
Between 1% and 5%	26	0	18	10
Between 5% and 10%	10	4	12	4
Between 10% and 20%	18	6	18	6
Between 20% and 30%	18	2	22	16
Between 30% and 40%	10	8	14	20
Between 40% and 50%	20	8	8	12
Between 50% and 60%	8	12	2	8
Between 60% and 70%	2	46	6	36
> 70%	6	38	0	10
Average	21.46%	52.66%	15.21%	36.84%
Standard Deviation	21.37%	28.61%	17.34%	26.65%
Max	72.21%	94.55%	60.81%	86.35%
Min	0.00%	0.00%	0.00%	0.00%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 5: Percentage savings in expected completion time. Case I: The Minimax Buddy System v. Heuristics 1-4 (dynamic control computed using expected cost)

4.2 Case II: 8 cities ($r = 2$)

4.2.1 Expected Cost

The results from Case I support the hypothesis that larger savings in expected cost can be obtained when implementing the buddy system if the large scale event leaves the total number of jobs significantly higher than the total number of vehicles available for reallocation. In particular, the buddy system is useful when resources are scarce relative to the need. To check the scalability of this observation the parameters in Case II are chosen so that resources are scarce in a slightly larger model. The remaining parameters are varied. The results for Case II are summarized in Table 6.

As in the previous case, Heuristics 2 and 4 performed poorly compared to the buddy system and were worse than Heuristics 1 and 3. In general, Heuristic 2 performs worse than Heuristic 4. We obtain savings of at least 4% from implementing the buddy system versus Heuristic 1 and 3. In line with our hypothesis, higher processing rates improved the performance of Heuristics 1 and 3 relative to the buddy system. This stands to reason since higher processing rates help reduce the need for scarce resources. For example, in the case where the processing rates in city groups 1-4 are all 1, the average cost savings is 36.27%. On the other hand, when the processing rates for city groups 1,2, and 3 are all 1 but the processing rate for city group 4 is 5, the average cost savings is 30.67%. Similarly, an increase in cost increases the importance of allocating resources effectively. For example, in the case where the cost vector is $\vec{h} = (2, 2, 1, 1, 1, 1, 1, 1)$, the average cost savings is 21% but when the cost vector is $\vec{h} = (2, 2, 8, 8, 8, 8, 8, 8)$ the average cost savings is 30.81%.

Furthermore, similar to earlier results, Heuristic 3 fared better than Heuristic 1 in all of the instances. This is further support for considering dynamics when making the reallocation decisions.

Percentage Savings	**Heuristic 1**	**Heuristic 2**	**Heuristic 3**	**Heuristic 4**
<= 1%	0	0	0	0
Between 1% and 5%	0	0	3	0
Between 5% and 10%	1	0	9	6
Between 10% and 20%	13	0	57	4
Between 20% and 30%	38	6	97	6
> 30%	248	294	134	284
Average	39.70%	68.51%	27.97%	57.26%
Standard Deviation	10.79%	10.23%	10.42%	15.87%
Max	66.51%	82.82%	55.25%	79.76%
Min	9.08%	28.02%	4.03%	7.55%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 6: Percentage savings in expected cost. Case II: The Buddy System v. Heuristics 1-4

4.2.2 Expected Completion Time for Expected Cost Optimal Policies

Percentage Savings	Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
Between -50% and -40%	0	0	19	0
Between -40% and -30%	21	0	24	0
Between -30% and -20%	0	0	39	0
Between -20% and -10%	3	0	7	0
Between -10% and -5%	65	0	12	0
Between -5% and -1%	0	0	2	0
Between -1% and 0%	0	0	15	0
Between 0% and 1%	0	0	0	0
Between 1% and 5%	9	0	6	0
Between 5% and 10%	6	0	40	0
Between 10% and 20%	66	0	74	0
Between 20% and 30%	94	0	45	0
Between 30% and 40%	24	0	17	4
Between 40% and 50%	10	82	0	141
Between 50% and 60%	0	33	0	46
Between 60% and 70%	0	168	0	102
$> 70\%$	0	17	0	7
Average	11.41%	59.01%	1.00%	55.44%
Standard Deviation	20.02%	7.99%	22.04%	8.64%
Max	46.99%	72.11%	38.18%	72.11%
Min	-34.89%	47.45%	-41.10%	36.51%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 7: Percentage savings in expected completion time. Case II: The Buddy System v. Heuristics 1-4

The results for Case II are summarized in Table 7. As can been seen from Table 7, the expected completion time for the buddy system is lower than that of Heuristics 2 and 4 in all of the instances. It is lower than that of Heuristic 1 in approximately 70% of the cases, and it is lower than that of Heuristic 3 in approximately 66% of the cases.

Heuristic 1 outperforms the buddy system heuristic when the processing rate for at least one city in city group 4 is high (5), and either the holding costs for at least one city in city group 4 is high (8) and the holding costs for at least two additional cities is also high (8), or, the holding costs for at least 4 cities are high (8). For instance, when

$\vec{\mu} = (1, 1, 1, 1, 1, 1, 1, 5)$ and either $\vec{h} = (2, 2, 8, 8, 8, 8, 1, 1)$ or $\vec{h} = (2, 2, 1, 1, 1, 8, 8, 1)$, the average time savings is -8.43%.

Heuristic 3 outperforms the buddy system heuristic when the processing rate for at least one city in city group 4 is 5 and either the holding cost for at least one city in city group 4 is 8 or the holding cost for at least 2 other cities is also 8. For example, when $\vec{\mu} = (1, 1, 1, 1, 1, 1, 1, 5)$ and $\vec{h} = (2, 2, 1, 1, 1, 1, 8, 1)$ or $\vec{h} = (2, 2, 1, 1, 8, 8, 1, 1)$ the average time savings is -4.15%.

4.2.3 Expected Completion Time for ($\mathbf{MP}'_r$)

The results for Case II are summarized in Table 8. As can been seen from Table 8, the expected completion time for the buddy system is lower than all of the heuristics in all of the instances, with savings of at least 50.16%.

Percentage Savings	**Heuristic 1**	**Heuristic 2**	**Heuristic 3**	**Heuristic 4**
<= 1%	0	0	0	0
Between 1% and 5%	0	0	0	0
Between 5% and 10%	0	0	0	0
Between 10% and 20%	0	0	0	0
Between 20% and 30%	0	0	0	0
Between 30% and 40%	0	0	0	0
Between 40% and 50%	0	0	0	0
Between 50% and 60%	90	0	210	0
Between 60% and 70%	120	0	90	0
> 70%	90	300	0	300
Average	62.41%	82.80%	56.67%	81.46%
Standard Deviation	8.17%	2.04%	8.10%	0.00%
Max	73.37%	85.91%	68.94%	81.46%
Min	52.35%	81.46%	50.16%	81.46%

(a) Note: The values in columns 2-5 represent the number of instances in the range in column 1.

Table 8: Percentage savings in expected completion time. Case II: The Minimax Buddy System v. Heuristics 1-4

5 Conclusion

In this paper, we consider how to respond to an emergency scenario in which a centralized dispatcher decides how to allocate EMS resources to affected cities from an unaffected region nearby. Our goal is to aid the decision-maker so that (s)he can bring the affected region back to its day-to-day levels of operations as efficiently as possible. We thus proposed a model in which EMS for individual cities are represented by Markovian queues with finite capacity. This model guides the pre-allocation stage. Additionally, the model is used to make the location decision via a queueing clearing system. It is well known that dynamic programming becomes computationally intractable when the size of the state space increases. We propose an implementable heuristic which at the macroscopic level is analogous to the classic resource allocation problem. At the microscopic level it allocates vehicles dynamically. Our results show that significant advantages in both cost and time savings are obtained when using our proposed heuristic as opposed to some common sense heuristics. The savings are robust over various parameter settings when the total number of jobs exceeds the total number of available vehicles.

There are several avenues for further research. For example, we have presented but one suggested heuristic that works well under the two criterion studied. To be certain there are others that would serve well under this and other criterion. At the very least, we hope that this paper can serve as a further catalyst to adopt systematic methods for resource allocation in the face of large scale emergency events.

6 Acknowledgements

The authors would like to thank Matthew Olson and Samuel Schmitz from the Mitre Corporation for expertise and guidance in formulating the model.

References

[1] K. Aaby, T. Cook, J. W. Herrmann, C. Jordan, and K. Wood. Simulating a mass vaccination clinic. online document. preprint (2005), available at `www.isr.umd.edu/Labs/CIM/projects/clinic/hcms.pdf`. 4

[2] K. Aaby, J. W. Herrmann, C. S. Jordan, M. Treadwell, and K. Wood. Montgomery county's public health service uses operations research to plan emergency mass dispensing and vaccination clinics. *INTERFACES*, 36(6):569–579, 2006. 4

[3] R. Banomyong and A. Sopadang. Using monte carlo simulation to refine emergency logistics response models: A case study. *International Journal of Physical Distribution and Logistics Management*, 40(8):709–721, 2010. 4

[4] R. Bjarnason, P. Tadepalli, A. Fern, and C. Niedner. Simulation-based optimization of resource placement and emergency response. In *Innovative Applications of Artificial Intelligence*, 2009. 4

[5] L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. *European Journal of Operational Research*, pages 451–463, 2003. 2

[6] K. F. Doerner, R. F. Hartl, M. Karall, and M. Reimann. Heuristic solution of an extended double-coverage ambulance location problem for austria. *Central European Journal of Operations Research*, 13(4):325–340, 2005. 3

[7] N. Geroliminis, M. G. Karlaftis, and A. Skabardonis. A spatial queuing model for the emergency vehicle districting and location problem. *Transportation Research Part B: Methodological*, 43(7):798–811, 2009. 3

[8] J. B. Goldberg. Operations research models for the deployment of emergency services vehicles. *EMS Management Journal*, 1(1):20–39, 2004. 2

[9] A. P. Iannoni, R. Morabito, and C. Saydam. An optimization approach for ambulance location and the districting of the response segments on highways. *European Journal of Operational Research*, 195(2):528–542, 2009. 4

[10] A. P. Iannoni, R. Morabito, and C. Saydam. Optimizing large-scale emergency medical system operations on highways using the hypercube queuing model. *Socio-Economic Planning Sciences*, 45(3):102–117, 2010. 4

[11] A. Ingolfsson, S. Budge, and E. Erkut. Optimal ambulance location with random delays and travel times. *Health Care Management Science*, 11:262–274, 2008. 2

[12] S. Jain and C. R. McLean. An architecture for integrated modeling and simulation for emergency response. In *IIE Annual Conference and Exhibition 2004*, pages 1503–1508, 2004. 4

[13] M. S. Maxwell, S. G. Henderson, and H. Topaloglu. Ambulance redeployment: An approximate dynamic programming approach. In *Winter Simulation Conference (WSC), Proceedings of the 2009*. Institute for Operations Research and Management Science, December 2009. 2, 3

[14] H. Morohosi. A case study of optimal ambulance location problems. In *The 7th International Symposium on Operations Research and Its Applications (ISORA 2008)*, pages 125–130. ORSC and APORC, October 2008. 3

[15] M. Restrepo, S. Henderson, and H. Topaloglu. Erlang loss models for the static deployment of ambulances. *Health Care Management Science*, 12(1):67–79, March 2009. 6

[16] V. Schmid and K. F. Doerner. Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operations Research*, 207(3):1293–1303, 2010. 3

A Dictionary of Symbols and Variables

Parameter Symbol	**Definition**
N	number of cities
λ_k, μ_k, n_k	arrival rate, service rate, and capacity at city k
$L_q^{n_k}$	Long-run average number of jobs waiting to be assigned
L^{n_k}	Average number of jobs in system
$W_q^{n_k}$	Average wait before being assigned a vehicle
W_{n_k}	Average wait until service is completed
L	Number of cities in affected region
$\vec{l}$	Vector of available vehicles
$\vec{m}$	Initial vector of calls above normalcy
$\vec{\nu}$	Vector of current vehicle configuration
$v_{N_k}(\vec{m}, \vec{\nu})$	Total cost of returning system to normalcy
h_k	Holding cost per unit time at city k
$c(\cdot)$	Cost function for moving one vehicle
N_D	Number of vehicles available for re-allocation
M	Total number of vehicles available for assignment
$A(M)$	Set of available assignments when total number of vehicles available for assignment is M
B	Number of cities in city group

Table 9: List of Parameters