Simulation-based Optimization of Resource Placement and Emergency Response

Ronald Bjarnason and **Prasad Tadepalli** and **Alan Fern**
{ronny, tadepall, afern}@eecs.oregonstate.edu
Oregon State University
Corvallis, OR, USA

Carl Niedner
cdn@coelo.com
Coelo Company of Design
Corvallis, OR, USA

Abstract

Many city governments are under pressure to optimize the utilization of their resources to respond to fire, rescue and medical emergencies. In this paper we describe a simulation-based software system that learns from a history of emergency requests to optimize the placement of resources and response policies. We describe a two-level random-restart hill climbing approach that yields policies which are better than the current practice and are sensitive to optimization metrics and population changes. Some of the policies learned by the system give insight into response practices that would otherwise be counterintuitive.

Introduction

Responding to fires and medical emergencies is one of the critical functions of city governments. Many cities face the problem of optimal placement of their limited resources and adopting appropriate response policies to meet the desired quality of service goals. The problem is compounded by the lack of knowledge of concrete future demands, the need to balance success across multiple criteria, and the sheer complexity of managing several pieces of equipment, people, and tasks at the same time. The fire chiefs and the city planners could also use help in evaluating the effects of new housing developments, population changes, and resource changes on the optimal placement and response policies and the resulting changes to the quality of service.

Simulation is often used to evaluate specific policies or policy changes in the domain of emergency response. For example, extensive simulations revealed that doubling the staff at a hospital in Taipei would reduce the response time by 50% even if the demand raised 10-fold (Su and Shih 2003). Another study showed that not requiring ambulances to be stationed at the fire stations improved the ambulance coverage without hurting the fire coverage (ReVelle and Snyder 1995).

What is missing in all these studies and others like them is the ability to automatically improve the policies through search and optimization. Such methods have been shown to be effective in other domains optimizing across multiple metrics, such as school redistricting (desJardins et al. 2006). Isolated subproblems of our domain such as vehicle placement have been extensively studied in Operations Research through mathematical programming and other optimization methods (Brotcorne, Laporte, and Semet 2003; Goldberg 2004; Fitzsimmons 1973; ReVelle and Snyder 1995). However, these studies are not based on simulations of real-world data nor do they study the problem comprehensively to include both resource placement and emergency response.

One of the challenges of this domain is that the policies produced by the any optimization method must be implementable by people. This usually means that the response policies should be simple to describe and follow prespecified constraints. Many cities use a "running order" to define a response policy for geographic response districts within the city. Each running order assigns a priority ordering over the set of stations that may respond to an emergency. The city is partitioned into response districts which are each assigned a running order.

In this paper, we describe a highly effective and usable tool called SOFER for Simulation-based Optimization of Fire and Emergency Response that combines simulation and optimization to find improved response policies. SOFER combines the two problems of where to host resources, e.g., fire trucks, and how to respond to emergencies, e.g., which truck to send, into an integrated optimization problem. Using this tool, it is possible to optimize both resource placement and emergency response to changes in new developments, population densities, and resource budgets on the one hand and the importance of different quality metrics on the other.

Our system can simulate a given policy on real data and gather statistics on its performance. The optimizer is built on top of the simulator and uses a two-level random-restart hill climbing search. At the higher level of resource placement, the system decides the home stations for each resource, e.g., engines, ladder trucks, ambulances. At the lower level of emergency response, the system allocates particular resources to emergency calls based on the location of the emergency and the availability of the resources.

We evaluated SOFER on real emergency call data collected over 5 years in Corvallis, OR, a small university town of about 53,000 people. Our system finds placement and response policies that outperform best practice by optimiz-

ing over a weighted sum of different metrics of quality of service. The system produces response policies which are sensitive to the importance of different metrics such as time for first response, time for the arrival of full service complement, etc. Secondly, we can easily evaluate the impact of new developments or population changes by adapting the call distribution to the changed situation and then optimizing the policies. In our particular case study, the results suggested that it is possible to close one of the fire stations and still achieve the desired level of performance with the usual volume of calls.

The rest of the paper is organized as follows. In the next section, we describe the problem of service policy learning for fire and emergency response for our particular city. We describe our simulation based on actual call data and our technical approach to the optimization problem. Next we describe the experimental results followed by discussion.

CFD System and Domain

Response Policy

The Corvallis Fire Department utilizes a "response district and running order" policy for response to fire and emergency medical requests. The entire service area is partitioned into separate response districts each of which maintains a preference over stations. This preference list is the fixed "running order" for that response district. Individual stations are led by lieutenants, who are responsible for memorizing their first-response district - those response districts where their station appears first in the running order. A lieutenant listening to a dispatch radio can alert his station to a coming call before the call is actually made. Response district boundaries are commonly drawn along major streets to simplify this memorization. When a request is received by the 9-1-1 dispatchers, it is identified by its geographic service district. The dispatchers assign responding units based on the running order and the currently available units. In cases when unit requests exceed available units, a request for assistance to neighboring departments is issued. In turn, CFD provides assistance to neighboring communities when necessary.

Individual firefighters and paramedics are loosely assigned to vehicles but tightly bound to their home stations. In many cases, a single complement of individuals may be co-located with multiple vehicles, and respond with whichever vehicle is currently requested, leaving the other vehicles in station and unstaffed. This flexible "either/or" response policy allows a smaller number of individuals to respond to a greater variety of requests in a shorter time period. In some stations, there are enough personnel to staff multiple vehicles simultaneously, and station lieutenants are at liberty to match appropriate personnel and vehicles as necessary.

Performance Metrics

Various metrics are used to measure the quality of responses to emergency requests. The National Fire Protection Association (NFPA) establishes performances standards (NFPA 2001) for local Fire and Rescue Departments, including standards for response times to fire and emergency incidents. The NFPA standards are non-binding and local departments

Figure 1: Map of Corvallis, OR. The five fire stations within the city limits are listed. An additional station is located three miles north of station 3 in the town of Lewisburg.

adapt the standards for their particular communities. The Corvallis Fire Department (CFD) goals for requests within the Corvallis city limits are that 45% of first arrivals be under 5 minutes, and that 90% of first arrivals be under 8 minutes. Performance relative to these goals, as well as goals for other geographic areas within the service area, is reported to the City of Corvallis on a quarterly basis (CFD 2008). In this paper, we will focus on the performance for calls within city limits, where the great majority of calls originate.

In addition to the arrival of first response units, NFPA also establishes standards for the arrival of the full assignment for fire suppression incidents and the arrival of advanced life support equipment to emergency medical incidents. Rather than focus on average response times, NFPA recommends standards based on the 90th percentile of responses. In addition to these metrics, we also report the percentage of requests that result in system overwhelms, when CFD cannot provide the requested service due to previous assignments.

The Data

Our data consists of a call log of 24,739 9-1-1 call requests received by CFD from Oct 20, 2003 to Sept 2, 2008. Each data entry contains information describing the time, location and specific nature of a single request. For privacy reasons, the x/y locations provided for each request have been rounded to the nearest 200 feet.

Simulation

Our simulator is a simplification of actual CFD policy and performance. It is driven by the provided call log and a dispatch policy. Some parameters within the simulation

are stochastic in nature, such as response time and assignment duration. To simplify the simulation process we have replaced these stochastic processes with fixed parameters, based on averages collected from the original data set.

Response and Travel Time Entire response time is calculated as a combination of elements from each request. We approximate 1 minute for the dispatcher to request a service unit and an additional 1 minute for the service unit to turnout to the call. We approximate the travel speed of response units as a function of distance traveled. For distances under 5 miles, we approximate a travel speed of 31 mph; distance $<$ 15 miles: 43 mph; distance $<$ 25 miles: 50 mph; distance $>$ 25 miles: 52 mph. We utilize a "taxi-cab" distance, summing the distances traveled in the x and y coordinates.

Assignment Duration The duration of each assignment is determined by the nature and cause of the request. We grouped individual calls into distinct response templates. Because various emergencies require the same complement of resources for roughly the same duration, these can be grouped together as equivalent in the simulation. Each template is defined by a unique complement of required personnel and vehicles and/or assignment duration for each unit type. Each assigned unit is considered to be "out of service" for the entire request period, which includes the 2 minutes of dispatch and turnout time, travel time to incident, completion of assignment at incident and return travel to home station. In practice, units may be called into service before returning to their home station, or even re-assigned to a different request while in service, although the latter is rare. In our simulations, each unit is considered to be out of service until it has completed the current assignment and returned to its home station.

Personnel, Vehicles and Units Of the various types of vehicles employed by CFD, we simulate the responses of the four most commonly requested vehicle types: tanker engines, ladder trucks, ambulances and the battalion chief. In order to limit combinatorial search, our simulation abstracts individual personnel and vehicles into combined "units". Each unit consists of personnel to operate a single vehicle and may contain one vehicle or two, not to be utilized simultaneously. When multiple units occupy a single station, the personnel of one unit may not operate the vehicles of another unit, even if it is idle and requested. Additionally, our simulation allows unlimited capacity for vehicles and personnel in each station, although the optimization process rarely exceed the actual capacity of each station. The actual number of CFD staff in service varies by time of day and day of week. Our simulation maintains a single battalion chief (BC) and sufficient personnel to staff 7 service units at all times. Our simulation maintains a vehicle for the BC and 11 additional vehicles: 3 ambulances, 6 engines and 2 ladder trucks. A map of the city of Corvallis is shown in Figure 1. Five of the six stations can be seen. Station 6 is to the northeast of Corvallis in the town of Lewisburg, part of the CFD service district. Our representation of CFD default station assignment is shown in Table 1a.

a.

U1	St 1	B. Chief	U5	St 3	Amb
U2		Eng & Lad	U6	St 4	Eng & Amb
U3	St 2	Eng	U7	St 5	Eng & Amb
U4	St 3	Eng & Lad	U8	St 6	Eng

b.	FR $<$5	FR $<$8	FR avg
CFD (reported)	62.00	94.00	4.70
CFD (simulated)	56.95	91.86	4.84

Table 1: a: Default station assignments for the 8 units in our simulation. b: The simulated performance of this configuration using an approximated CFD running order compared to figures reported by CFD.

Among the data that we received from CFD was a file of 1068 requests that were annotated with both x/y location data and full running orders to each location. From this data we were able to construct an approximation of actual CFD policy within our simulator. For the 24,739 calls in the main data set, we approximate the CFD running order to each call with the running order to the closest of the 1068 calls in the annotated file. CFD reports their performance quarterly to the City of Corvallis (CFD 2008) based on three metrics: average response time for first responder (FR avg), percentage of requests in which the first responder arrived in less than 5 minutes (FR$<$5), and the same metric for 8 minutes (FR$<$8). We compare our simulated performance with the actual data reported by CFD in Table 1b. Differences between our approximated policy and the most recent CFD data illustrate the difficulty in accurately modeling complex systems.

Response District Grid Without explicit knowledge of CFD response districts, we have drawn response districts by overlaying the service area with a two-scale grid. The large grid is delineated by the boundary box defined by the entire service area and contains 6 square cells, measuring 100,000 feet to a side (roughly 360 square miles each). The smaller grid is delineated by the boundary box drawn around the 6 active stations, and contains 55 cells. Sides of small cells measure 5000 feet (0.9 square miles). Each of these 61 cells is assigned a running order policy for requests originating from within the boundaries of the cell.

Optimization Methods

Policies in the fire and emergency response domain can be altered in two distinct ways—by changing the basic unit composition/station assignments and by changing the running orders of the response districts. It is important that our optimized configurations and response policies are similar in structure to those actually implemented by CFD. Responding units must remain in their stations until requested, and must be assigned based on the running order priorities of the requesting response district. Finding improved response policies within such restrictions can be achieved in one of two ways—either learning directly in the restricted space or approximating a restricted policy from an optimized unrestricted policy.

Reward

Collectively optimizing the various performance criteria will require compromise. To help us balance performance over all metrics, we have designed a weighted reward for individual responses based on four parameters. We formulate the optimization problem as maximizing the average expected reward per call. Each of the parameters is closely associated with a goal that will improve the quality of service within the response area. These are the four parameters:

FR measures the elapsed time between the placement of the 9-1-1 call and the arrival of the *first responder* on scene.

FC measures the elapsed time between the 9-1-1 call and the arrival of the last of the requested units, the time when the *full complement* of responding units is on scene.

TB is a fixed *time bonus* received when the first responder arrives on scene in less time than the stated time goal for that response region. For requests originating within the Corvallis city limits, this time goal is 5 minutes.

OP is a fixed *overwhelm penalty* received when there are not enough resources available to meet current requests. In these cases, additional units are requested from neighboring municipalities.

The default reward is represented as

$$-(w_1\text{FR}) - (w_2\text{FC}) + (\text{FR} < 5 \text{ ? TB}) + (\text{overwhelm ? OP})$$

with w_1=w_2=1, TB=10 and OP= –100 where units of all variables are measured in minutes.

Learning Unrestricted Policies

We considered a number of techniques to optimize the policies in this space. Some optimization techniques that rely on continuous variables, such as linear programming are not suitable to our discrete policy space. Traditional dynamic-programming-style methods are problematic because of our large state space, but using our reward function and a parameterized value function over state-action pairs, we were able to learn a successful policy using an approximate dynamic programming method called Least Squares Policy Iteration (LSPI) (Lagoudakis and Parr 2003). We use LSPI to learn a value function over available actions, given the current state, which takes the place of the "running order" method of choosing actions.

Our value function is a linear combination of 10 weighted features: one feature for each of the 8 units indicating the duration (in minutes) until the unit returns to its home station, the reward received from the current action and a boolean feature indicating if the system is currently overwhelmed. This function approximates the Q-value of taking an action, a, from a state, s, given the weights w: $Q(s,a,w) = \phi(s,a)^{\mathsf{T}}w$, where $\phi(s,a)$ is the matrix of basis features for the state-action pair (s,a). LSPI fits the weights, w, to minimize the squared difference between the estimated Q-values and the discounted rewards observed in simulations of the call log. The weights learned by LSPI estimate Q-values and choose actions using an ϵ-greedy algorithm with ϵ-exploration decreasing over subsequent iterations. LSPI will converge to a set of weights that we can use to approximate the value of available actions.

	FR% <5	FR% <8	FR avg	% over
LSPI	60.64	93.66	4.59	6.14
LSPI (apprx)	56.56	92.88	4.83	6.44

Table 2: results from LSPI and an approximated running order based on the simulated LSPI policy.

Our unrestricted action space allowed LSPI to assign *any* available unit to a request independent of station assignment or running order. The value function learned by LSPI is able to predict the value of actions in real time. Such a policy cannot be directly applied to our domain because response policies must be static and defined over a fixed set of running orders. We approximate a running order based on our LSPI policy by recording the frequency of station requests for each cell in the response district grid. The running order for each grid cell is fixed to the order of most frequently requested stations. In Table 2 we can see that the LSPI policy outperforms the approximated policy on all four metrics. In addition to the three metrics reported by CFD, we also measure the percentage of calls that result in a system overwhelm (%over). Given the absence of policy restrictions, it is not surprising that LSPI performs well, but it is not clear how to construct a good restricted policy from the unrestricted solution.

Learning Restricted Policies

Rather than rely on an intermediate step that constructs one policy from another, we have found a simple solution that directly improves an existing policy within the restricted space. Our two-phase hill climbing algorithm is shown in Algorithm 1. The running orders for the grid cells are fixed in place while the unit station assignments are optimized, which are then fixed while the running orders are optimized. These two phases continue until the policy converges.

```
1 while policy not converged do
2     optimize unit station assignments
3     optimize running orders
```

Algorithm 1: Two-phase hill climbing optimization.

Optimization of Running Orders The hill-climbing technique for running orders optimizes each cell in the response grid individually while the running orders for the other cells are held fixed. Each of the 61 grid cells is optimized, in turn, by simulating the call log with each of the 720 (6!) possible running orders. This algorithm (described in Algorithm 2) is factorial in the number of stations. While this is practical for the 6-station CFD, additional approximation methods will be required for larger departments with a dozen or more stations.

Optimization of Unit Assignments Without limits placed on station capacity, testing every possible unit assignment

```
1 for each cell, c do
2     for each running order, r do
3         t = simulate calls(using r for calls in cell c);
4         if t > best-score then
5             best-score = t;
6             best-order = r;
7     c.running-order = best-order;
```
Algorithm 2: Running Order Optimization Method

configuration would not only be combinatorially expensive, but would also test many unreasonable configurations. To optimize the unit assignments, we employ a hill climbing method, described in **Algorithm 3**, that tests each possible single-vehicle-reassignment. These reassignments can be of two forms: either moving a vehicle from a multiple-vehicle-unit to a single-vehicle-unit or moving an entire single-vehicle-unit to a different station. Units are not allowed more than two vehicles are are not allowed to contain multiple vehicles of the same type. On each pass through the loop, the reassignment that improves performance the most is made permanent.

```
1  while improving do
2      improving = false;
3      for each vehicle, v and each station, s do
4          if move (v to s) then
5              t = simulate calls(using fixed running order);
6              if t > best-score then
7                  best-score = t;
8                  improving = true;
9                  best-v = v; best-s = s;
10     move(best-v to best-s);
```
Algorithm 3: Unit Assignment Optimization

Random Initialization This two-phase optimization process performs well if it is seeded with a reasonable unit configuration and accompanying running order. For our domain in general, this may be a reasonable assumption, as fire chiefs and city planners will have access to current best practices (or reasonable approximations) for their particular department. However, by starting with fixed configurations and running orders, we may be predisposing our method against interesting and novel solutions. To avoid this, we prefer to start our algorithm with random unit configurations and random running orders. Unfortunately, the described algorithm overwhelmingly leads towards poor local minima when initialized randomly. Optimizing running orders for random unit configurations reinforces the mediocre station assignments. In turn, the unit assignment optimization favors station re-assignments that perform best for the bad running orders.

To break this self-reinforcing loop, we introduce a small amount of search in the unit configuration optimization step. **Algorithm 3** is expanded to include a partial optimization of running orders in the simulation on **line 5**. For each vehicle reassignment, an optimization over all grid cells similar to **Algorithm 2** is performed, but only for the first two elements of each running order. This reduces the number of simulations per cell from 720 to 30. While not exact, it approximates the optimization that will occur during the running order optimization step, should that particular vehicle reassignment be made permanent. Thus the evaluation of each vehicle reassignment is a more accurate reflection of its true value. Using this improved heuristic to evaluate station reassignments enables the algorithm to break out of self-reinforcing local minima.

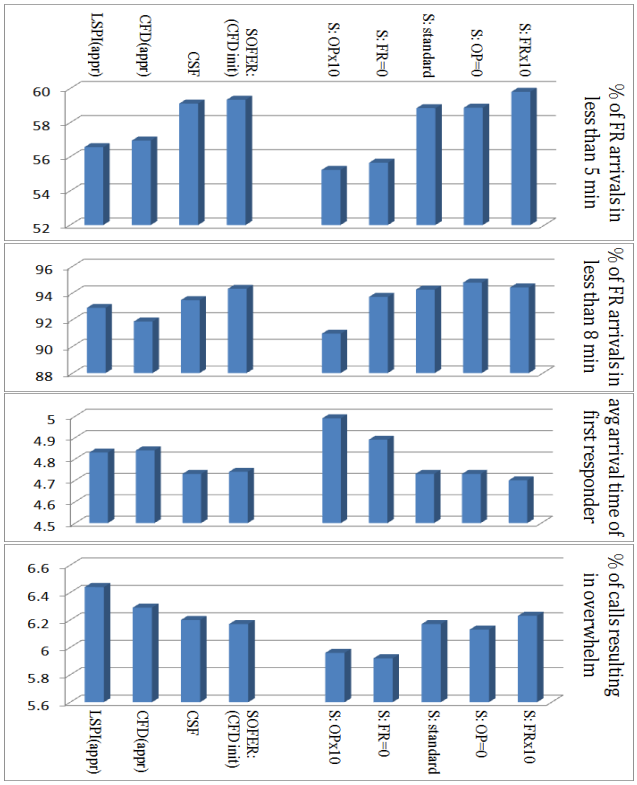

Figure 2: Comparison of response policies over four metrics.

Evaluation of Results

All results from SOFER are from tests on unseen data. We have trained our policies using the call from the first half of the data set and report performance by simulating the learned policies on the second half of the data.

Policy Comparison

We compare our results against four baseline standards—the approximated performance of the LSPI algorithm, the simulated performance of the approximated CFD running order, a closest-station-first policy and our own SOFER algorithm initialized with the default CFD unit station assignments and approximated running order, without random restarts. Each of these reference policies were simulated using the second half of the call data. The SOFER policy was trained on the first half of the data. In addition to the standard reward, we

report results for 4 additional adjusted rewards, in which weights for two of the parameters in the reward function have been changed to magnify performance for specific metrics. For the FR and OP parameters, we report performance for when the regular values of these parameters are multiplied by 10 as well as when they are eliminated completely. For each of the reward structures, SOFER was restarted and allowed to converge 5 times. The policy that performed best on the training data is retained and used to simulate the test data, which results are reported. These results are displayed in Figure 2. For the top two graphs, higher values indicate improved performance. The converse is true for the lower two graphs.

It is interesting to note that that "closest station first" (CSF) policy does not fit within the restricted policy space of our running order policies. Such a policy would require drawing response districts along unnatural boundaries. It is impressive that at least one of our SOFER policies (and usually more) are better than the CSF policy for each of the four metrics.

The results illustrate the necessity to balance performance across all metrics. When emphasis is placed on the arrival time of the first responder (FR$\times$10), relative performance improves in the first three metrics, associated with arrival times of first responders, while performance decreases in the fourth graph, measuring the percentage of system overwhelms. A similar effect is seen when the overwhelm penalty is set to zero (OP=0). The opposite effect is seen when the value of the overwhelm penalty is increased (OP$\times$10) or the value of the average first responder is eliminated (FR=0). One interesting artifact of our tests is the consistent tendency of SOFER to converge to policies that leave Station 5 unmanned, which occurred in 4 of the 5 random-restart tests, two of which we examine in Table 3. More surprising is that Station 5 is left unmanned in the SOFER policy initialized with default CFD responses, which we would expect to have less of a tendency to stray from the initial CFD configuration.

	X: (FR$\times$10)		Y: (FR=0)	
U1	St 1	B.Chief	St 1	B. Chief
U2		Eng &Amb		Eng &Amb
U3		Eng		Eng
U4	St 2	Eng & Amb	St 2	Ldr & Amb
U5	St 3	Ldr & Amb		Eng
U6		Eng	St 3	Eng & Amb
U7	St 4	Eng & Ldr		Ldr & Amb
U8	St 6	Eng	St 6	Eng

	FR% <5	FR% <8	FR avg	% over
X:(FR$\times$10)	59.81	94.41	4.70	6.23
Y:(FR=0)	55.65	93.70	4.89	5.92
CFD(simulated)	56.95	93.46	4.84	6.29
CSF	59.12	94.78	4.73	6.20

Table 3: Two example configurations generated by SOFER. Configurations X and Y were generated with rewards in which the FR component of the reward had ten times its normal value, and where it was eliminated completely, respectively. Both leave Station 5 unstaffed. Configuration Y also leaves station 4 unstaffed. These are compared to our simulated CFD policy and a "closest station first" policy, based on four metrics.

Population Changes

The results involving Station 5 are particularly interesting in light of the projected growth in the surrounding areas. To the southeast of Station 5 is a large area outside of the city limits, roughly 110 acres of which has recently been annexed by the City of Corvallis and projected to fill with subdivisions over the next 10 years. In an attempt to see how the system would behave under a stressed load of calls, we projected calls into random locations in this area. The calls were sampled from a mixture of high and low density housing areas of Corvallis, and randomly assigned dates and times according to the existing distribution of calls. In an attempt to learn a policy that would allocate resources to the fringe stations, we simulated 2000 calls in the unincorporated area to the southeast of station 5. Additionally, we simulated 400 calls in the small rectangular area at the southern boundaries of the city limits that is surrounded by property within city boundaries. Even with an additional 2000 calls placed right across the street from Station 5, SOFER still converged to policies that left Station 5 unmanned in 3 of 5 trials. In each of these trials SOFER found configurations that increased staffing in Stations 2 and 3 to cover the additional calls.

We ran an additional simulation with even more calls in the northwest of town, retaining the 2000 calls from the previous test, and adding 800 calls roughly 1 mile WNW of Station 5. This area encompasses roughly 100 acres of farmland and is well out of the 5-minute reach of Stations 2 and 3. With a significant number of simulated calls outside the range of other stations, SOFER finally consistently used Station 5. Under these severely stressed simulations, our averaged SOFER results handily beat the simulated CFD policy in all 4 metrics. The results for both of these tests, marked A and B, are reported in Figure 3.

Discussion

We have presented results in the Fire, Rescue and Emergency Response domain, based on a 5 year log of actual emergency requests. We presented SOFER, a two-level random restart hill climbing algorithm that optimizes unit station assignments and in-time response policies. Our simulated results using the SOFER algorithm improve on simulated best-known policy, and are sensitive to preference over measured metrics and population density changes. One novel conclusion of our results strongly suggest that performance across multiple metrics can be maintained even if one of the six stations is closed. This conclusion is counterintuitive, providing valuable insight into policy structures, especially for city governments that may be investigating policies that maintain performance utilizing reduced resources.

Simulation Improvements Our current simulation is a coarse abstraction of actual performance of the Corvallis

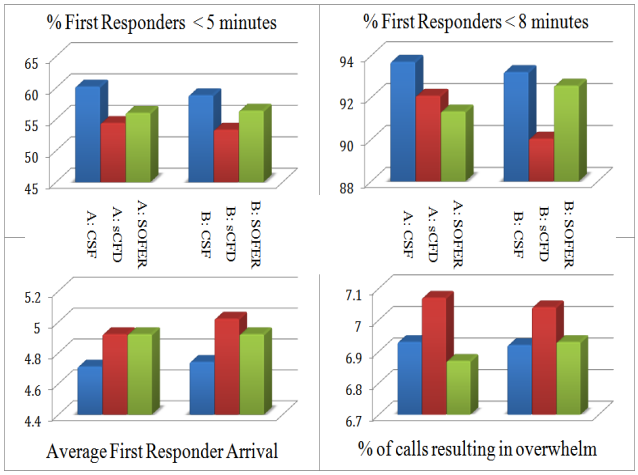

Figure 3: Comparison of response policies for two separate growth projections (A and B). For the top two charts, higher values increase performance; for the lower two, lower values are better. We compare our results of SOFER against a closest-station-first (CSF) policy and against the CFD simulated policy (sCFD). Tests A and B of projected growth are described in the section on Population Changes.

Fire Department responses. We can yet improve the accuracy of our simulator on many levels. We can improve travel time estimates using more reliable travel data and/or an accurate travel model of the city of Corvallis. We could more accurately model current CFD responses if we could obtain and incorporate boundaries for actual response districts and the running orders for those districts.

Optimization Improvements We were able to see marked improvement in our algorithm by inserting an approximation step into our unit assignment hill climbing step. It is possible that a similar step in the running order step may also yield valuable results.

The policy restrictions inherent to our system prevent the direct application of some standard optimization techniques. We have shown that even approximations of very successful unrestricted policies perform poorly compared to our algorithm that optimizes within the restricted space. Nevertheless, it seems reasonable to think that one might find an improved policy by first beginning with an optimal solution. We will continue to work in this area, adapting standard techniques and improving approximations of unrestricted policies. More work needs to be done to understand the scope of such approaches.

Multiscale Optimization Part of the difficulty of optimizing policies in this domain is because making station assignments limits the possible actions that can be taken at the base level (e.g. if there isn't an ambulance in Station 2, we can't choose to send an ambulance from there). Part of this problem is alleviated by our parameterized action space, in the form of running orders, but the problem is not entirely solved. We refer to domains of this type as "multi-scale optimization" domains. They are characterized as optimization problems in which optimization must be done at different scales. In our case the scales are temporal. On a large scale, we must decide where to permanently place our units. On the smaller scale, we must decide how to operate them on daily basis. Unlike in the traditional hierarchial reinforcement learning literature (Dietterich 2000; Sutton, Precup, and Singh 1999; Parr and Russell 1998), the higher level tasks do not consist of the lower level tasks but impose complex constraints on them. We would like to design novel and efficient algorithms for this interesting class of problems.

Acknowledgements This material is based upon work supported by DARPA grant xyz. Special thanks to the Corvallis Fire Department for valuable consultation and the data logs and to Neville Mehta for helpful discussion.

References

Brotcorne, L.; Laporte, G.; and Semet, F. 2003. Ambulance location and relocation models. *European Journal of Operational Research* 147(3):451–463.

CFD. 2008. *(Corvallis Fire Department), EMAC Minutes, City of Corvallis Quarterly Report*.

desJardins, M.; Bulka, B.; Carr, R.; Hunt, A.; Rathod, P.; and Rheingans, P. 2006. Heuristic search and information visualization methods for school redistricting. In *Proceedings of IAAI*. AAAI Press.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13:227–303.

Fitzsimmons, J. A. 1973. A methodology for emergency ambulance deployment. *Management Science* 19(6):627–636.

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

Lagoudakis, M. G., and Parr, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4:1107–1149.

NFPA. 2001. *(National Fire Protection Association), NFPA 1710: Standard for the Organization and Deployment of Fire Suppression Operations, Emergency Medical Operations, and Special Operations to the Public by Career Fire Departments, 2001 Edition*. Quincy, Massachusetts: National Fire Protection Association, Inc.

Parr, R., and Russell, S. 1998. Reinforcement learning with hierarchies of machines. *Advances in Neural Information Processing Systems* 10.

ReVelle, C., and Snyder, S. 1995. Integrated fire and ambulance siting: A deterministic model. *Socio-Economic Planning Sciences* 29(4):261–271.

Su, S., and Shih, C.-L. 2003. Modeling an emergency medical services system using computer simulation. *International Journal of Medical Informatics* 72(1-3):57–72.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1-2):181–211.