A Decision Theoretic Framework for Emergency Responder Dispatch

Industrial Applications Track

Ayan Mukhopadhyay
Vanderbilt University
Nashville, TN, USA
ayan.mukhopadhyay@vanderbilt.edu

Zilin Wang
Vanderbilt University
Nashville, TN, USA
zilin.wang@vanderbilt.edu

Yevgeniy Vorobeychik
Vanderbilt University
Nashville, TN, USA
yevgeniy.vorobeychik@vanderbilt.edu

ABSTRACT

Efficient emergency response is a major concern in urban areas across the globe. The problem of predicting incidents and subsequently allocating responders spatially has been studied extensively. The problem of dynamically deploying responders, however, has received considerably less attention and has been noted as a difficult problem in prior literature due to inherent complexities in the environment in which such problems evolve. We formulate a decision-theoretic framework for the emergency responder problem, which effectively leverages state-of-the-art methods for continuous-time spatio-temporal incident forecasting. We formulate the responder dispatch problem as a Semi-Markov Decision Process (SMDP) that evolves in continuous time, and efficiently engineer its representation leveraging structural insights of the problem space. We then propose a novel approach to solve the problem based on policy iteration. First, we transform the SMDP into a discrete-time MDP (DTMDP). Then, we simulate our system to estimate value of states as well as learn the state transition probabilities of the transformed DTMDP. We also design heuristic policies with which our algorithm can be seeded. We validate the efficacy of our approach on real traffic and assault data from Nashville, USA, as well as synthetic data, and highlight that our approach outperforms the state of the art emergency responder dispatch system.

KEYWORDS

Incident Response; Decision Theory; Semi-Markov Decision Processes; Survival Analysis

ACM Reference Format:
Ayan Mukhopadhyay, Zilin Wang, and Yevgeniy Vorobeychik. 2018. A Decision Theoretic Framework for Emergency Responder Dispatch. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Managing and responding to urban incidents such as traffic accidents, fire and crime are fundamental challenges faced by cities across the world. An increase in housing density, population, and traffic has further complicated this problem. From the perspective of emergency responders, there are three major challenges: 1) predicting or learning where incidents might happen based on past data, 2) optimizing locations for emergency responders based on such predictions, and 3) deploying responders as and when incidents happen. While the first two problems have been studied extensively [9, 10, 14, 17]), the last problem has not received much attention. Without an approach for dynamic deployment of emergency responders, principled algorithmic techniques are often eschewed in practice as delays resulting from ad-hoc dispatch strategies can result in the loss of life [2], and erode the trust in the system.

We develop a principled decision theoretic framework for continuous -time resource dispatch. We assume that locations and counts of responders and stations are exogenously given, and focus on the dynamic dispatch strategy. We formulate the problem as a Semi-Markov Decision Process (SMDP) which evolves in continuous time. and derive an equivalent DTMDP for the formulation. In order to obtain an optimal policy for the SMDP, we propose an algorithm based on policy iteration. We access a simulator to simultaneously simulate our system to estimate values of states, as well as estimate transition probabilities for the DTMDP. We also design efficient heuristic policies leveraging problem structure and domain expertise that can be used to seed the policy iteration algorithm.

We validate our findings by comparing our approach to existing state-of-the-art approaches [9] using real traffic data from Nashville, a major metropolitan city in the US, as well as simulated data. Our results demonstrate that our principled approach to dynamic dispatch significantly outperforms the state-of-the-art alternative.

2 RELATED WORK

The problem of optimally locating and dispatching responders includes several subproblems that have been studied largely independently. The first is the incident prediction problem which looks at predicting when and where incident happen and is a well-studied problem [1, 10, 13, 15, 17]. Given such a model, the next problem is to allocate resources to meet a specific performance criterion. We refer readers to recent work done by Mukhopadhyay et al. [9], which provides a comprehensive summary of prior work in this domain.

We focus on the final subproblem in this paper: given an incident prediction algorithm and a fixed spatial distribution of responders, how do we optimally dispatch responders as incidents happen, particularly when multiple responder types are involved. Emergency response has been noted to face inherent uncertainty, with the additional challenge that while responders do not know when and

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

where incidents will happen, the expectation is that response is very timely [3]. This problem has typicaly been studied as part of the responder location problem [7] as well as a joint optimization problem of fairly distributing resources and optimizing response times [16]. Most of the approaches above to either the responder location, or dispatch, are *static*, given a particular distribution over incidents in space and time. The environment, however, is dynamic, and allowing dispatch to adjust to current information and environment is a crucial consideration in practice which is generally ignored in prior art. A recent approach has taken a decision-theoretic approach to dispatching responders and is quite principled and well-structured [6]. It however, suffers from a major flaw - it assumes that the total response times for emergency responders (sum of travel time and service time) are exponentially distributed. This strong distributional assumption results in two simplifications - first, it lets the usage of well-known distributions to model and estimate the state transition probabilities in closed-form and secondly, enables the usage of Continuous-Time Markov Decision Processes. However, travel times are not exponentially distributed in practice, and this limits the practical applicability of the work. Our work is a principled approach for dynamic responder optimization which addresses these limitations.

3 BACKGROUND

3.1 Optimal Static Dispatch

We describe here an approach from prior art that dispatches emergency responders on the basis of an allocation mechanism which has guarantees on wait-times [9]. We call this policy WTG (Wait-Time Guarantee). The resource allocation algorithm on which WTG is based is a two-fold approach. First, the total area to be serviced is discretized into a set of grids and a probabilistic model of temporal incident arrival is learned for each grid. Then, based on such a model, depots and responders are allocated in space to maximize the total coverage area of responders with bounds on wait times. The algorithm attempts to assign as many grids as possible to specific depots. Based on this allocation, when an incident happens in a grid, WTG first checks if the depot that the grid is assigned to has any free responders or not and dispatches one if available. If not, it looks for other responders returning from service to depots that have no pending incidents. If it fails to locate such responders, the incident enters a waiting queue in the depot that it is assigned to. The algorithm has been shown to outperform existing emergency response systems in Nashville, USA. We treat this approach as our baseline.

3.2 Semi-Markov Decision Processes

We formally model the problem of dynamic incident response as a semi-Markov decision process (SMDP) [4]. An SMDP is described by a tuple $\{S, A, p_{ij}(a), t(i, j, a), \rho(i, a), \alpha\}$ where S is a finite state space, A is the set of actions, $p_{ij}(a)$ is the probability with which the process transitions from state i to state j when action a is taken, $t(i, j, a)$ is a distribution over the time spent during the transition from state i to state j under action a, $\rho(i, a)$ is the reward received when action a is taken in state i, and α is the discount factor for future rewards. In our case, while the state space is finite, it is comprised of a collection of variables, and fully enumerating the state space is not tractable. We consequently leverage a factored state representation described below.

3.3 Dynamic Bayes Networks

Consider a stationary Markov chain representing a dynamical system, with state transition probabilities $P_{ss'}$, and consider a factored representation of states s using a collection of n variables $\{X_1, \ldots, X_n\}$. Let X and X' then be factored state representations for successive states s and s'. A Dynamic Bayes Network (DBN) is a graphical representation of $P_{ss'}$ in factored space, that is, a combination of an acyclic directed graph G_0 on X of intra-state edges, with (i, j) an edge from X_i to X_j, and G_1 a directed graph of inter-state edges on X, X', where direction is from variables (nodes) in X to variables in X', where i, j is an edge from X_i to X'_j. Let the graph G be the union of G_0 and G_1, and for each variable X'_i for a successor state, let $Pa(X'_i)$ be all variables in $X \cup X'$ which have directed edges to X'_i. Then, for each variable X'_i in the successor state we denote it's conditional probability distribution as $P(X'_i|Pa(X'_i))$. The probability distribution for the successor state X' conditional on current state X is then $P_{XX'} = \prod_i P(X'_i|Pa(X'_i))$. In a DBN representation, both the directed acyclic graph G, and the corresponding conditional probabilities $P(X'_i|Pa(X'_i))$ are given.

4 A CONTINUOUS-TIME SPATIO-TEMPORAL MODEL OF DYNAMIC EMERGENCY RESPONSE

Our goal is to develop an approach for making optimal decisions about emergency responder placement and incident response in a dynamic, continuous-time, stochastic environment. We begin with several assumptions on the problem structure and information provided *a priori*. First, we assume that we are given a spatial map broken up into a finite collection of grids G, and assume that we are given an exogenous spatio-temporal model of incident arrival in continuous time over this collection of grids (we describe one such model in Section 4.3). Second, we assume that for each spatial grid cell, the temporal distribution of incidents is homogeneous. This assumption is merely a reflection of the granularity of the spatial discretization: we can in principle always discretize space finely enough so that this assumption approximately holds. Our third assumption is that emergency responders are housed in a fixed and exogenously specified collection of *depots*, each with a pre-defined set of emergency responders. This reflects the relatively long time scale of decisions about the spatial location of the depots themselves.

We assume that when an incident happens, a free responder (if available) is dispatched to the site of the incident. Once dispatched, the time to service consists of two parts: 1) time taken to travel to the scene of the incident, and 2) time taken to attend to the incident. If no free responders are available, then the incident enters a waiting queue; once a responder becomes available, it is then assigned to incidents waiting for service in the order in which they appear in the queue. We can naturally accommodate incidents with different priorities in this framework by using a priority queue. Finally, we assume that the distribution of vehicles in depots is given; it can be calculated using a number of methods in the literature (e.g., [9]).

We assume that each responder is assigned to a specific depot, and must return to that depot when it is not responding to any incident.

We refer to the entire spatio-temporal process of incident arrival as well as the status of all responders and depots as our *world*. We consider the evolution of this world in continuous time. The dynamics of the world are primarily governed by two events that provide the scope of decision-making : occurrence of traffic incidents and completion of servicing. These are events when responders need to be either sent back to their depots or re-directed to other incidents. Observe that the effect of these actions is not instantaneous. For example, when an incident happens and the action of dispatching a responder is taken, the occurrence of the next state is dependent on the time taken by the concerned responder to travel to the site and attend to the incident. Therefore, given a snapshot of our world (which we refer to as a *state* in our decision making process), the next state depends not only on the given state and the action taken, but also on how *time* evolves between the states.

We model the continuous-time spatio-temporal dynamic decision problem faced by emergency responders using the machinery of Semi-Markov Decision Problems described above. We next describe each of the elements of the SMDP as it captures the emergency responder problem, while also explaining the special structure of this problem which is used for both representing and solving it.

4.1 States

A state at time t is represented by s^t which consists of a tuple $\{I^t, R^t\}$, where I^t is a collection of grid indices that are waiting to be serviced, ordered according to the relative times of incident occurrence. Thus, for any indices j, k and corresponding $i^t_j, i^t_k \in I^t$, $j > k$ implies that incident at grid i^t_j occurred after the incident at grid i^t_k. R^t corresponds to information about the set of responders at time t with $|R^t| = b \, \forall t$, where b the total number of responders. Each entry $r^t_j \in R^t$ is a set $\{h^t_j, p^t_j, d^t_j, c^t_j\}$, where h^t_j corresponds to the depot that responder j is assigned to, p^t_j is the position of responder j, d^t_j is the destination that it is traveling to (where $d^t_j = 0$ indicates that responder j has no destination assigned), and c^t_j is its current condition, all observed at the state of our world at time t. Observe that $h^t_j, p^t_j, d^t_j \in G \, \forall t, j$. Additionally, $c^t_j \in \{0, 1\} \, \forall j, t$, with $c^t_j = 0$ meaning that the responder is currently engaged in service and $c^t_j = 1$ meaning that it is free and available to be dispatched. We acknowledge that while it is not necessary to include h^t_* as a part of the state since responders assignments to depots do not change over time and are exogenously provided, we include this information nonetheless as it makes the states self-containing. Finally, the set of all states is denoted by S.

4.2 Actions

Actions in our world correspond to directing responders either to incidents or back to their depots. We denote the set of all actions by A, with an action by $a^i_{kj} \in A$ corresponding to a decision to send a responder which is from depot k and is currently in grid j, to an incident at grid i. Additionally, we use $A(s^i)$ to denote the set of actions that are available in state $s^i \in S$. In addition, we impose a constraint that whenever responders are available and an incident occurs, we always immediately dispatch *some responder*. We now make several important observations. First, due to the continuous-time nature of our model where a single incident arrives at any point in time, in our model of the world, at most a single action in A is ever taken. This is a result of two model features: 1) since routing a responder to an incident necessarily effects a state transition, and 2) since we always respond to incidents if responders are available, whenever we have multiple available responders, it must be the case that there is at most one new incident to respond to.

4.3 Transitions

We first look at how our *world* evolves before describing transitions. In our model, states evolve between events that provide the scope of decision-making . We refer to these times as decision epochs and such states as decision making states. For convenience, we segregate the two types of events (occurrence of incidents and completion of servicing) and refer to states in which responders become free as completion-states S_c and states in which incidents occur as incident-states S_a. We also observe that states can evolve between decision epochs, and every such change in state does not present a chance to make decisions (i.e., the corresponding $A(s)$ is empty). As an example, responders going back to their depot move through different grids, which updates the state variable R, but presents no scope for decision making in our process, unless an incident happens. We also make the assumption that no two events (incident or completion of servicing) can occur simultaneously in our world. In case such a scenario arises, since the world evolves in continuous time, we can add an infinitesimally small time interval to segregate the two events and create two separate states.

In order to segregate decision making states from other states, we divide the model of our world into two processes as in prior work [12]: an embedded MDP that is observed only at decision epochs, and a natural process that evolves as if it is observed continuously through time, but presents no relevant information to the decision maker, unless the world is at a decision epoch. Since the decision making process is essentially the natural process observed at special (decision making) instances, the two processes are always the same at decision epochs. This segregation helps us in two ways: first, it truncates the state space by only looking at states that are relevant from the perspective of decision-making, and more importantly, it lets us remain agnostic about how the natural process evolves, thereby letting us exploit well-established models for transitions between our decision epochs.

Having described the evolution of our *world*, we now look at both the transition time between events, as well as the probability of observing a state, given the last state and action taken. We define the former first, denoting the time between two states s^i and s^j by the random variable t_{ij}.

Since decision epochs are governed by the occurrence of incidents and service completions, we first describe our models between these events. We denote the time between incidents by the random variable t^a and time to service an incident by t^s. We model inter-arrival times between incidents by using survival-analysis, that has been widely used to model time to events, and has recently been used to forecast urban incidents [9, 10], making it a natural candidate for our purposes. Survival analysis is a broad class of methods that are used to model the distribution of time and risk for events. We use the accelerated time effect (AFT) model in which co-variates increase or decrease the expected time to next incident [8], letting us directly model time, rather than risk. Formally, we model the time t between successive incidents as $f(t|w)$, where t follows an exponential distribution, and w is a set of arbitrary features that affect t. As in standard AFT models, we model the arrival rate for the exponential distribution as a log-linear model in terms of the features w. Thus, the probability distribution of time to next incident t^a in a given spatial grid is

$$f(t^a|w) = \lambda^a e^{-\lambda t^a} \tag{1}$$

where $\log \lambda^a = \sum_{w_j \in W} \theta_j w_j$ and θ are the regression coefficients learned from data. This form of the model is particularly useful to us in simulating our world, the purpose of which we explain in Section 5. As in prior work [9], we model service time t^s by an exponential distribution

$$f(t^s) = \lambda^s e^{-\lambda t^s}.$$

In combination, our models of t^a and t^s are crucial as they induce memoryless arrival and service times which allow us to model the entire dynamic dispatch process as an SMDP with special structure.

To understand the temporal transitions, we first explain how time evolves between two states by considering the following scenarios. First, consider a series of two events as shown in Fig. 1, where at time t_1, an incident happens at grid i (state s^{t1}) and a vehicle is dispatched from grid j, and at time t_2, the vehicle finishes servicing the incident (state s^{t_2}). The time between these two states is $d_{t_1 t_2} v + t^s$, where $d_{t_1 t_2}$ is the distance between the concerned grids at states s^{t_1} and s^{t_2} and v is the (known) velocity of the responders. Now, let us consider the scenario that an incident or completion of service happens at time t_3, resulting in a new state s^{t_3}, such that $t_1 < t_3 < t_2$. The time between states s^{t_3} and s^{t_2} now depends on whether the responder traveling to grid j had reached its destination or not. If it had not, we can calculate the distance left for it to travel and hence the time $t_{t_3 t_2}$ based on information available in s^{t_3}. In case it had reached and started servicing, we can leverage memorylessness of the service time distribution and *reset* the remaining service time, thereby estimating $t_{t_3 t_2}$. Next, suppose that an incident happens at time t_1 (state s^{t_1}), and the next incident occurs at time t_2 (state s^{t_2}). The transition between incidents can be modeled directly by the survival model in Eq. 1. Now, imagine that a new event (service completion, state s^{t_3}) happens at time t_3 such that $t_1 < t_3 < t_2$, as shown in Fig. 2. Observe that since incident arrivals are also memoryless, the time to state s^{t_2}, as seen from state s^{t_3} can again be reset.

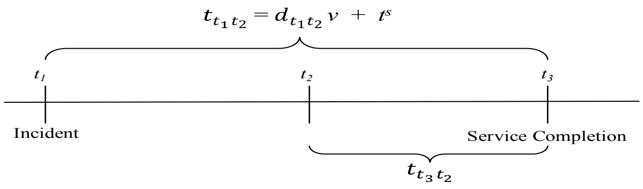


Figure 1: Transition Times when $s^j \in S_c$

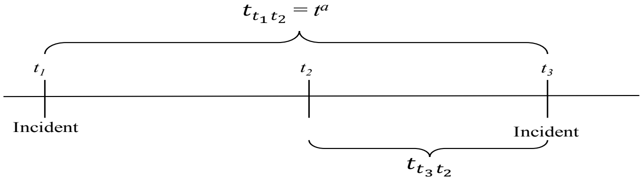


Figure 2: Transition Times when $s^j \in S_a$

To summarize, for any two states s^i and s^j

$$t_{ij} = \begin{cases} \sum_{k=1}^{b} \mathbb{I}\{(c_k^j - c_k^i) > 0\} d_{ij} v + t_s & \text{if } s^j \in S_c \\ \\ t_{ij}^a \quad \text{if } s^j \in S_a. \end{cases} \tag{2}$$

Having considered the temporal evolution between states, we now consider the probability of observing a state at a decision epoch, given a particular state at the last decision epoch and the associated action. We denote the probability that the process moves from state s^i (at a decision epoch) to state s^j (in the next decision epoch) under action a by $p_{ij}(a)$, where states belong to the embedded MDP. Note that the probability of transition between these states cannot solely be modeled using the incident and service time distributions $f(t^a)$ and $f(t^s)$, as the natural process evolves between decision epochs. $p_{ij}(a)$ thus captures the transition probabilities for states between decision epochs while taking into account the implicit evolution of the natural process.

For a compact representation of the state transition distribution p, we use a Dynamic Bayes Network (DBN) to leverage conditional independence relationships among the state variables. The structure of the DBN is visualized in Fig. 3. Specifically,

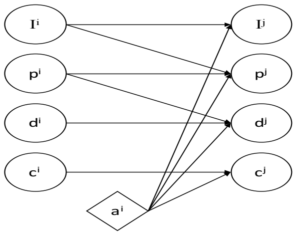


Figure 3: Dynamic Bayes Network to model Inter-State Transition

$$p_{ij}(a) = P(I^j|I^i, a^i)P(p^j|p^i, I^i, a^i)P(d^j|d^i, p^i, a^i)P(c^j|c^i, a^i) \tag{3}$$

where a^i is the action taken at state s^i. While the structure of the DBN is self-explanatory, we point out some of the key insights that leverage the structure of the responder dispatch problem. For predicting incidents, we use the standalone survival model which captures all relevant information for predicting future incidents. Consequently, future incidents are conditionally independent of the other state variables, given current incidents. However, pending incidents are removed from states as they are serviced; hence, the transition function for I is dependent on the action taken.

4.4 Rewards

Rewards in SMDP usually have two components: a lump sum instantaneous reward for taking actions, and a continuous time reward as the process evolves. Our system only involves the former, which we denote by $\rho(s, a)$, for taking action a in state s. Observe that the best (myopic) scenario in emergency response is when a responder is already present at the scene of the incident, while the worst scenario occurs when the only available responder has to travel a distance equal to the largest possible distance in the given area to attend to an incident. We design our reward structure in accordance with these observations. When a responder is dispatched in state s^i to attend to an incident that results in a state completion state s^j, we denote the reward as $d_{max} - d_{ij}$, where d_{max} is the largest possible distance in the given area under consideration and d_{ij} is the distance between the concerned grids. Also, we assume that the action of sending a responder back to its depot has 0 reward, since this is the default action that must be taken if an incident is not waiting to be serviced.

4.5 Dispatch Process and Decision Problem

In summary, the evolution of the responder dispatch world happens as follows:

(1) Once the system is in state s^i, an action $a \in A(s^i)$ is taken.
(2) The system receives an instantaneous reward $\rho(s^i, a)$.
(3) The system transitions to state s^j according to the probability distribution $p_{ij}(a)$
(4) The system takes time t to make the transition, where $t \sim t_{ij}$

Having described the structure of the SMDP representing dynamic continuous-time spatio-temporal problem of emergency response, we proceed to outline the general goal in solving it. Any solution to this problem is to obtain a policy π, which for any given state s^i prescribes an action $\pi(s^i)$ to be taken in that state. Ideally, the policy should produce an *optimal*, i.e., utility-maximizing, action a, with the notion and form of utility defined beforehand. Formally, for any arbitrary state s^i, we define the expected discounted total reward over an infinite horizon as

$$V^{\pi}(i) = \sum_{n=0}^{\infty} \mathbb{E}\{e^{-\alpha T_n} \rho(s^n, \pi(s^n))\}$$

where s^n is the state at n^{th} decision epoch, and T_n its duration. Our goal is to find the optimal policy π^* which, starting from for an arbitrary state i, maximizes the sum of expected discounted rewards, with a minor caveat. Emergency responders, in practice, are often governed by instructions to send the responders that are close to the scene of an incident. Although this is myopic, it provides the best chances of dealing with the current incident at hand, and failing to do so might result in immediate damage and/or loss of life. We take this into effect by adding a constraint to our optimization problem as follows

$$\begin{aligned} &\sup_{\pi} V^{\pi}(i) \\ &s.t \\ &\rho(s, \pi(s)) \geq \gamma \rho(s, a) \,\forall a \in A(s) \,\forall s \in S. \end{aligned} \tag{4}$$

The constraint enforces that the immediate reward taken at a step is at least γ times the best reward, where γ is a user-defined parameter based on the nature of the specific emergency response system.

5 SOLUTION APPROACH

We now present an approach for computing an optimal policy of the formulated SMDP model of dynamic emergency response. Our approach is based on policy iteration [12], which is a two-step process for computing an optimal policy for an MDP by iteratively improving upon a starting seed policy until convergence. In the first step, the values of states are estimated under a fixed policy from the last iteration. The second *policy improvement* step then incrementally improves upon the previous policy by finding a better action in each step.

Unlike conventional MDPs with a small state space, we face two challenges that make direct policy iteration impractical. First, we have a combinatorially large state space. Second, the state transition probabilities $p_{ij}(a)$ are unknown; rather, we can simulate transitions and use such simulations to estimate transition probabilities. Our approach below addresses these problems.

5.1 Discretization

We first present an approach that assumes that the state transition probabilities are known. We subsequently relax this assumption.

A general approach of solving an SMDP is to derive an equivalent discrete-time MDP (DTMDP) [4], and then solve the DTMDP by standard techniques like policy iteration. Before presenting the conversion, we introduce some additional notation. We denote by F the cumulative distribution function for the random variable t, that is used to model the transition time between states. To begin with, for each pair of states s^i and s^j, and action a, we define

$$\beta_{\alpha}(i, j, a) = \int_0^{\infty} e^{-\alpha t} F_{ij}^{a}(dt). \tag{5}$$

Using Eq. 5, we then define the *expected discount factor* as

$$\beta_{\alpha}(i, a) = \sum_j p_{ij}(a) \beta_{\alpha}(i, j, a)$$

Intuitively, $\beta_{\alpha}(i, a)$ measures the significance of one unit of reward obtained at the current decision epoch, valued at the next decision epoch, when the continuous time discount factor is α. Finally, we define $\beta_{\alpha} = \sup_{i,a} \beta_{\alpha}(i, a)$.

As mentioned earlier, transition times between two states s^i and s^j depend on whether $s^j \in S_a$ or $s^j \in S_c$. We look at these two cases separately.

Case I: When $j \in S_c$, from equation 2

$$t = c(i, j) + t^s \Rightarrow t^s = t - c(i, j) \tag{6}$$

where $c(i, j) = \sum_{k=1}^{b} \mathbb{I}\{(c_k^j - c_k^i) > 0\} d_{ij} v$, which given two states, is a constant. Let $g(t)$ represent the density function of the random variable t, and $f(t^s)$ represent the density of the service time distribution, as mentioned earlier. Now,

$$g(t) = f(t^s) = f(t - c(i, j)) = \lambda_s e^{-\lambda_s (t - c(i,j))} \tag{7}$$

and

$$F(t) = \int_{c(i,j)}^{t} \lambda_s e^{-\lambda_s (t - c(i,j))} dt.$$

Then,

$$\begin{aligned}\beta_\alpha(i,a,j) &= \int_{c(i,j)}^{\infty} e^{-\alpha t}\lambda_s e^{-\lambda_s(t-c(i,j))}dt \\ &= \frac{\lambda_s}{\lambda_s+\alpha}e^{-\alpha c(i,j)}.\end{aligned} \tag{8}$$

Case II: When $j \in S_a$, let the time from the last incident to s_i be τ. Since t^a is distributed exponentially,

$$g(t > x) = f(t^a > \tau + x | t^a > \tau)$$

Hence, similar to Case I,

$$\beta_\alpha(i,a,j) = \frac{\lambda^a_{ij}}{\alpha + \lambda^a_{ij}}.$$

Having described the necessary transformations, we define the corresponding Discrete-time Markov Decision Process (DTMDP) as $\{S, A, \bar{p}_{ij}, \rho, V_\beta, \beta_\alpha\}$, where $\bar{p}_{ij}(a) = \beta_\alpha^{-1}\beta_\alpha(i,a,j)p_{ij}(a)$ is the scaled probability state transition function and β_α is the updated discount factor. The value of a state s^i in the transformed MDP can then be represented as

$$V_\beta(s^i) = \sup_a\{\rho(i,a) + \sum_j \beta_\alpha(i,a,j)p_{ij}(a)V_\alpha(j)$$

This DTMDP is equivalent to the original SMDP according to the total reward criterion (for the proof of this equivalence, see Hu and Yue [4]).

5.2 SimTrans : Simulate and Transform

The transformed DTMDP still suffers from the two technical difficulties discussed above. We now proceed to address these through a novel algorithm. We call this algorithm *SimTrans*, as it combines **sim**ulating a generative model and **trans**forming it into an equivalent DTMDP, in order to solve a SMDP formulation.

The basic idea is to approach policy iteration by simulating the world to estimate values of states with one important addition. Choosing an optimal action in a state when given access to a simulator has been previously explored by Kearns et al. [5], Péret and Garcia [11]. SimTrans accesses the simulator to estimate a state's value but at the same time, iteratively builds confident estimates of the state transition probabilities, which can then be used for the transformed DTMDP. Once such estimates are available for any state-action pair, the algorithm chooses to accept such an estimate and avoids simulating, thus reducing computational costs. We present SimTrans in Algorithm 1.

The algorithm presents standard policy iteration with one modification, a procedure *ESTVAL*, that is added to estimate values of states. We start with a subset of states and, given an arbitrary state s^i and a policy π, SimTrans decides whether to simulate the world to estimate $V^\pi(s^i)$ or access pre-computed transition probabilities and directly calculate $V_\beta(s^i)$. To do this, *ESTVAL* first accesses a procedure $conf(s^i, \pi(s^i)$ (refer operation 28 in Algorithm 1) to check if it has access to *confident* estimates of $p_{ij}(\pi(s^i)\ \forall j \in R(i)$, where $R(i)$ represents the set of possible next states from s^i (we formally define the notion of *confidence* later). If such estimates are available (refer to operation 29), the algorithm moves on to find the expected discounted rewards from future states, the expectation being taken with respect to $\hat{p}_{ij}(\pi(s^i))$ (the estimated value of $p_{ij}(\pi(s^i)))$. In case such estimates are not available, the algorithm estimates $V^\pi(s^i)$ by a direct Monte-Carlo estimation approach. It simulates the world m times under policy π, starting from s^i, where m is the user-defined Monte-Carlo budget (refer operation 32). At any iteration l of the simulation, we obtain an estimate $\hat{V}^\pi_l(s^i)$. The final estimate is simply calculated as the sample mean of the estimates. Thus, $V^\pi(s^i) = \frac{\sum_{l=i}^m \hat{V}^\pi_l(s^i)}{m}$. Also, every time the world is simulated, the algorithm tracks the state transitions generated (referred by ϕ in SimTrans) and updates estimates of state transition probabilities (refer operation 34). Throughout the process, SimTrans only looks at actions that are at least γ times the best myopic action available.

We now look at how the procedure checks confidence in estimates generated by the algorithm. We point out that while any notion of statistical *confidence* can be used in the algorithm, we choose the standard two-sided confidence bound, with two user-defined parameters. The algorithm takes as input a tolerance bandwidth ω and a confidence parameter r. For any arbitrary $p(.)$, given a series of its estimates, the algorithm calculates with confidence r, the interval $\tilde{p}$ within which the true parameter lies as $\tilde{p} = \frac{2Z_r s}{\sqrt{n-1}}$, where s is the sample deviation, n is the number of available samples and Z_r is the critical value of the standard normal for confidence r. If $\tilde{p} \leq \omega$, the algorithm accepts the estimates; otherwise, it simulates the world to get an estimation.

Having looked at the approach that combines simulation and discretization, we now address the problem of an extremely large state space. To address this issue, we take two measures. First, we start with an initial subset of states and gradually add states as we simulate the world. Second, we seed our policy iteration algorithm with heuristic policies, that are designed based on prior work [9] and domain expertise. Thus, for any state, we always have access to a default policy.

We consider two seed policies. The first is WTG, which is based on a static assignment of depots to spatial service grids, described in Section 3. The second is a novel heuristic policy, Multiple Depot Heuristic (MDH) that addresses the two shortcomings of WTG. First, although the placement algorithm of WTG is built on guarantees on upper bounds on wait times, the dispatch algorithm, in practice, does not guarantee bounds on wait times for all incidents. This happens since some grids (say grid i) are not assigned to any depots and the nearest depot (say depot j) is contacted in case incidents occur in such grids. This causes wait time bounds to marginally fail for depot j (as responding to grid i was not a part of the resource placement algorithm and is enforced on the depot during dispatch). Second, domain expertise dictates that while cross-depot dispatch should be reduced (for vehicle wear and tear as well as other maintenance issues), it should not be completely ignored. We present this dispatch policy formally in Algorithm 2. When an incident happens in grid i in state s^t, MDH first looks at the depot that grid i is assigned to (refer operation 3) according to a responder placement algorithm, such as Mukhopadhyay et al. [9], Silva and Serra [14] (Mukhopadhyay et al. [9] also looks at vehicles returning after serving incidents). In case no free responders are available, it sorts all free responders available on the basis of proximity to the scene of the incident (refer operation 8). It then iteratively checks each responder, and sends one from a depot that has no pending calls. While this approach honors the fixed assignment of depots

Algorithm 1 SimTrans

```
INPUT: Initial Policy $\pi_0$, Initial states $S_0$, Maximum Iterations $MAX\_ITER$, Confidence Parameters $\omega$, $r$
OUTPUT: Final Policy $\pi^*$
for $l = 1..MAX\_ITER$ do
Policy Evaluation:
    for $i \in S_{l-1}$ do
        $V_l^{\pi_{l-1}}(i)$ = EstVal($\pi, i, b, m, l$)
    end for
Policy Improvement:
    for $i \in S_{l-1}$ do
        if conf($i, \pi(i)$) $\leq \omega$ then
            $\pi_l(i) = \arg\max_{a \in A(i)} \rho(i, a) + \sum_{j \in R(i))} \hat{p}_{ij}(\pi_{l-1}(i))\beta_\alpha(i, a, j)$ EstVal($\pi_{l-1}, j, b, m, l$)
        else
            for $a \in A(i)$ do
                $\pi' = \pi_{l-1}$
                $\pi'(i) = a$
                $V^a(i) =$ EstVal($\pi', i, b, m, l$)
            end for
            $\pi_l(i) = \arg\max_a V^a(i)$
        end if
    end for
    $S_l$ = UpdateStates($S_{l-1}$)
end for


procedure EstVal($\pi, i, b, m, l$)
    if Available($V_l^\pi(i)$) then
        return $V_l^\pi(i)$
    end if
    if b=0 then
        return $\rho(i, \pi(i))$
    end if
    if conf($i, \pi(i)$) $\leq \omega$ then
        $V^\pi(i) = r_\alpha(i, \pi(i)) + \sum_{j \in R(i))} p_{ij}(\pi(i))\beta_\alpha(i, a, j)$ EstVal($\pi, j, b-1, m, l$)
    else
        for k = 1..m do
            $\hat{V}_k^\pi(i), \phi_k$ = Simulate($\pi, i, b$)
        end for
        UpdateP($\phi_1, ..., \phi_m$)
        $V^\pi(i) = \frac{\sum_{k=1}^{k=m} \hat{V}_k^\pi(i)}{m}$
    end if
    return $V^\pi(i)$
end procedure
```

Algorithm 2 Multi Depot Heuristic

```
INPUT: Grid $i$, State $s^t$, Allocation $Alloc$
OUTPUT: $\pi(s^t)$ : Policy for the current state
Let $j$ = $Alloc(i)$
count = Free($j$)
if count > 0 then
    return $a^i_{jj}$
end if
RespSort = Sort($R^t$, $i$)
for $r \in$ RespSort do:
    if Pending($h_r^t$) = 0 then
        return $a^i_{h_r^t p_r^t}$
    end if
end for
```

to grids, it accesses cross-depot dispatch in some cases, when the assigned depot is unable to service.

6 EXPERIMENTAL EVALUATION

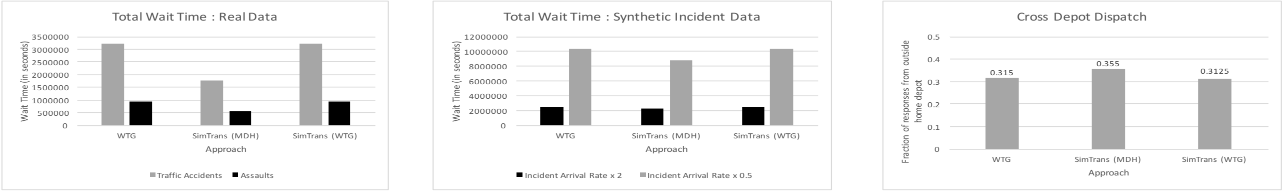
6.1 Data

Our evaluation uses traffic accident data and assault data obtained from the *fire* and *police* departments in a medium-size city in the US, with a population of approximately 700,000. For this fire department, traffic accidents and crimes requiring ambulance services comprise a large majority of incidents it responds to (fires, in contrast, are relatively rare). We looked at traffic accident data for 26 months, from 2014 - 2016, comprising of a total of 19,373 traffic accidents, and assault data for the year 2014, consisting of a total of 7,100 incidents. Each accident is accompanied by its time of occurrence, the time at which the first responding vehicle reached the scene and the time at which the last responding vehicle was back at service, which refers to completion of servicing an incident. We use the same incident prediction model as in prior work [9]. We also produce two synthetic datasets using our incident prediction model, by scaling the incident arrival rate in the exponential distribution by 0.5 and 2. This provides us with a test bed to evaluate the model on potential urban areas that are different than the one in our dataset.

6.2 Results

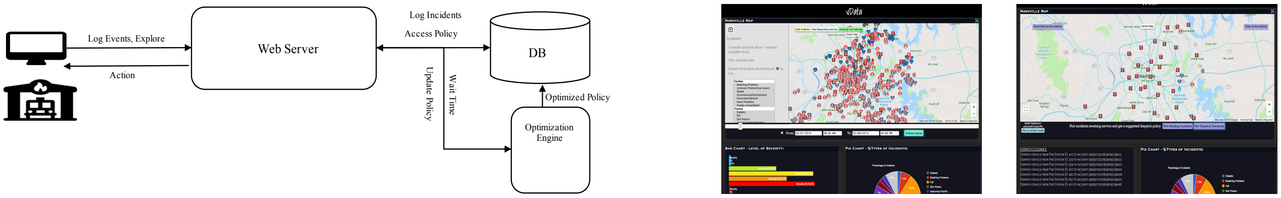
We evaluate the proposed solution by a direct comparison of wait times for incidents. We set $\omega = 0.1$ and $r = 0.95$ for our experiments. As a constraint on the action space, we set $\gamma = 0.95$, since quick emergency response by ambulances is critical to saving lives. This means that, for any state, SimTrans only looks at actions that provide at least 0.95 times the immediate reward of the best available action. For the complete set of incidents, we dispatch responders based on standalone WTG and SimTrans. We test the performance of SimTrans by first using WTG (SimWTG) as a seeding policy, and then MDH (SimMDH). We present the results in Fig. 4a. The results are shown after 3 iterations of SimTrans. We see that SimMDH reduces wait times by almost 50% for both traffic accidents and assaults. SimWTG, on the other hand, outperforms standalone WTG only marginally on both types of incidents.

The results on synthetic dataset are shown in Fig. 4b. In this case, SimWTG edges out WTG marginally, while SimMDH again shows a remarkable reduction in wait times. The overall wait times for synthetic data are slightly higher than real data as the generative model assigns non-zero probability to grids that are unlikely to see traffic accidents. These grids are rarely sampled, but the presence of incidents in such grids drives wait times higher. In order to analyze the reduction in wait times, we check for each algorithm the percentage of incidents that are served from assigned depots



(a) Response Times : Real Data **(b) Response Times : Synthetic Data** **(c) Cross Depot Dispatch**

Figure 4: Results (Lower is Better for Response Times)



(a) Dashboard Architecture **(b) Data Visualization** **(c) Dispatch Decisions**

Figure 5: Incident Prediction and Dynamic Dispatch Dashboard

(Fig. 4c). We see that SimMDH increases cross depot dispatch by over 4%, thereby increasing the availability of responders to travel shorter distances when they need to.

7 IMPLEMENTATION

We have created an open-source web-based platform that acts as a one-stop complete solution for incident response. The tool facilitates low-latency bi-directional communication between clients (emergency response stations) and a central server. A high-level software architecture for the tool is shown in Fig. 5a. The tool is currently in the process of being deployed at the Nashville Fire Department. We briefly describe the features of the tool here.

The tool provides the ability to each client to visualize historical data in the form of both markers and heat maps, as well as analytics based on such incidents (Fig. 5b), thus helping in providing an immediate visual summary of incidents. Users can also instantly shift to a *future* mode, and predict incidents based on Eq. 1 for specific dates or date ranges, which serves as an important mechanism for resource planning and budget allocations. Finally, when an incident happens, the server pushes a notification to each client about the location of the incident as a marker on the map. It also highlights which emergency responder should be sent to the site of the incident (Fig. 5c). This feature provides crucial assistance to the emergency responders thus aiding real-time decision making. The tool also provides an exploratory mode for making long-term decisions. It provides the ability to users to add a new depot at a specific location, allocate responders to it and understand how much the expected wait time changes upon the addition of such a depot. This helps organizations strategically during expansions or relocating depots and/or responders.

8 CONCLUSION

We proposed a principled decision-theoretic framework to address the problem of emergency responder dispatch in a continuous-time, dynamic and stochastic environment. We framed the problem as a Semi-Markov Decision Process leveraging insights from the problem structure. We used a well established incident prediction model and a Dynamic Bayes Net to create a factored and compact representation of the state transitions. Then, we proposed a novel algorithm to solve it, that simulates the environment and simultaneously learns transition probabilities. Also, we designed efficient heuristics to seed our proposed algorithm. We evaluated our algorithm on both real data from Nashville, a major metropolitan area in USA as well as synthetic data and showed that our model outperforms previous state-of-the-art. Finally, we created an open-source tool for all emergency response organizations that is in the process of getting deployed in a major city in USA.

9 ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation (CNS-1640624, IIS-1649972, and IIS-1526860), Office of Naval Research (N00014-15-1-2621) and Army Research Office (W911NF-16-1-0069).

REFERENCES

[1] Williams Ackaah and Mohammed Salifu. 2011. Crash prediction model for two-lane rural highways in the Ashanti region of Ghana. *IATSS research* 35, 1 (2011), 34–40.
[2] Robert Davis. 2005. Six Minutes to Live or Die. *USA Today* (2005).
[3] Stefan Felder and Henrik Brinkmann. 2002. Spatial allocation of emergency medical services: minimising the death rate or providing equal access? *Regional Science and Urban Economics* 32, 1 (2002), 27–45.
[4] Qiying Hu and Wuyi Yue. 2007. *Markov decision processes with their applications.* Vol. 14. Springer Science & Business Media.
[5] Michael Kearns, Yishay Mansour, and Andrew Y Ng. 2002. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine learning* 49, 2 (2002), 193–208.
[6] Sean K Keneally, Matthew J Robbins, and Brian J Lunday. 2016. A markov decision process model for the optimal dispatch of military medical evacuation assets. *Health care management science* 19, 2 (2016), 111–129.
[7] Xueping Li, Zhaoxia Zhao, Xiaoyan Zhu, and Tami Wyatt. 2011. Covering models and optimization techniques for emergency response facility location

and planning: a review. *Mathematical Methods of Operations Research* 74, 3 (2011), 281–310.

[8] Rupert G Miller Jr. 2011. *Survival analysis*. Vol. 66. John Wiley & Sons.

[9] Ayan Mukhopadhyay, Yevgeniy Vorobeychik, Abhishek Dubey, and Gautam Biswas. 2017. Prioritized Allocation of Emergency Responders based on a Continuous-Time Incident Prediction Model. In *International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 168–177.

[10] Ayan Mukhopadhyay, Chao Zhang, Yevgeniy Vorobeychik, Milind Tambe, Kenneth Pence, and Paul Speer. 2016. Optimal Allocation of Police Patrol Resources Using a Continuous-Time Crime Model. In *Conference on Decision and Game Theory for Security*.

[11] Laurent Péret and Frédérick Garcia. 2004. On-line search for solving Markov decision processes via heuristic sampling. In *Proceedings of the 16th European Conference on Artificial Intelligence*. IOS Press, 530–534.

[12] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[13] Martin B Short, Maria R D'Orsogna, Virginia B Pasour, George E Tita, Paul J Brantingham, Andrea L Bertozzi, and Lincoln B Chayes. 2008. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences* 18, supp01 (2008), 1249–1267.

[14] Francisco Silva and Daniel Serra. 2008. Locating emergency services with different priorities: the priority queuing covering location problem. *Journal of the Operational Research Society* 59, 9 (2008), 1229–1238.

[15] Praprut Songchitruksa and Kevin Balke. 2006. Assessing weather, environment, and loop data for real-time freeway incident prediction. *Transportation Research Record: Journal of the Transportation Research Board* 1959 (2006), 105–113.

[16] Hector Toro-DíAz, Maria E Mayorga, Sunarin Chanta, and Laura A Mclay. 2013. Joint location and dispatching decisions for emergency medical services. *Computers & Industrial Engineering* 64, 4 (2013), 917–928.

[17] Chao Zhang, Arunesh Sinha, and Milind Tambe. 2015. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *International Conference on Autonomous Agents and Multiagent Systems*. 1351–1359.