Emergency Response Optimization using Online Hybrid Planning

Durga Harish Dayapule*[1], Aswin Raghavan*[2], Prasad Tadepalli[1], Alan Fern[1]
[1] School of EECS, Oregon State University, Corvallis, OR, USA
[2] SRI International, 201 Washington Rd. Princeton, NJ08540
{dayapuld, alan.fern}@oregonstate.edu, aswin.raghavan@sri.com, tadepall@eecs.orst.edu

Abstract

This paper poses the planning problem faced by the dispatcher responding to urban emergencies as a Hybrid (Discrete and Continuous) State and Action Markov Decision Process (HSA-MDP). We evaluate the performance of three online planning algorithms based on hindsight optimization for HSA-MDPs on real-world emergency data in the city of Corvallis, USA. The approach takes into account and respects the policy constraints imposed by the emergency department. We show that our algorithms outperform a heuristic policy commonly used by dispatchers by significantly reducing the average response time as well as lowering the fraction of unanswered calls. Our results give new insights into the problem such as withholding of resources for future emergencies in some situations.

1 Application

The emergency response problem has been well studied in the planning and operations research communites in the last five decades. For example see the comprehensive surveys of [Goldberg, 2004; Bélanger *et al.*, 2015; Aringhieri *et al.*, 2016]. The critical goal is to improve the performance of emergency responders, which is typically measured by the time elapsed between the reporting of the emergency and the arrival of responders. This domain has attracted the attention of many cities, e.g., Melbourne [Maxwell *et al.*, 2014] and London [McCormack and Coates, 2015], in order to counteract the effects of urban sprawl (leading to larger service regions) alongside limited budgets and fewer responders. In this work we study a new domain-independent planning approach to the problem of emergency response. Our online planning approach coupled with a parametrized (relational) domain description provides a flexible tool that can easily be applied to different regions.

Prior research in this domain is traditionally categorized into *static* and *dynamic* problems. The static problems concern one-shot (time-invariant) decisions, exemplified by ambulance location and coverage problems [Brotcorne *et al.*, 2003; Bjarnason *et al.*, 2009] evaluated against a fixed dispatch policy. On the other hand, dynamic problems take into account the highly random and time-dependent nature of the demand, the state of the roads, weather conditions and real-time traffic information [Azizan *et al.*, 2013]) and output a set of actions (dispatch or relocation of units) for each emergency. The most commonly used policy is to dispatch the nearest responders in terms of travel-time [Jagtenberg *et al.*, 2015, see Table 2]. Previous work has shown that performance of the online planners dominate this greedy baseline [Haghani *et al.*, 2004; Dean, 2008]. However, these approaches have been domain-specific and the viability of a general on-line data-dependent planning approach remained open.

In principle, online action selection offers a scalable alternative both to global dispatch policy optimization and engineering a parametrized policy space. It allows quick and near-optimal responses for any given emergency taking into account the subsequent emergencies that may arise in the near future. The emergency domain is naturally formulated as a Hybrid State and Action Markov Decision Process (HSA-MDP, [Kveton *et al.*, 2006]). The state space consists of both continuous and discrete random variables that encode the status of all responders and details of the current emergency. Unfortunately the hybrid nature of the state space makes algorithms based on dynamic programming to be intractable in general. The second challenge of the online formulation is the factored action space [Raghavan *et al.*, 2012] that is the cross product of assignments to individual responders.

Our contributions are as follows. First, we encode the emergency response domain in the Relational Dynamic Influence Diagram Language (RDDL) [Sanner, 2011]. Second, we present a sampling approach to generate potentially large number of exogenous event sequences from a modest amount of real world data. Third, we evaluate the performance and scalability of a state of the art algorithm based on hindsight optimization (HOP) [Raghavan *et al.*, 2017] and its variants and compare them to a baseline. Our analysis shows that HOP-based algorithms outperform the widely used nearest available vehicle heuristic both in response time and service rate. It also gives useful insights into the problem such as when withholding of some resources for future emergencies improves performance.

*These authors contributed equally to this work

2 HSA-MDP Formulation

A discrete-time MDP is a tuple $(\mathbb{S}, \mathbb{A}, T, R)$ where $\mathbb{S}$ is a state set, $\mathbb{A}$ is an action set, $T : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow [0, 1]$ denotes the stochastic transition function for the next time step such that $T(s, a, s') = P(s'|s, a)$, and $R : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$ denotes the state-action reward function. In this paper we focus on finite horizon planning for a specified horizon h with the objective of maximizing the expected total reward over h steps.

In Hybrid State and Action MDPs (HSA-MDPs) [Kveton *et al.*, 2006; Sanner *et al.*, 2012] the state space $\mathbb{S}$ and action space $\mathbb{A}$ are products of finite sets of state variables $X = \{X_1, \dots, X_l\}$ and action variables $A = \{A_1, \dots, A_m\}$ respectively, where each variable is either discrete or continuous. The transition function of the MDP is factored over the state variables, ie. $P(s'|s, a)$ is represented as a product of conditional probability distributions $P_i(q_i) = P(X'_i|q_i)$ for $i = 1, \dots, l$, where $q_i \subseteq \{X, A, X'\}$, X' are the next state variables, and the set of dependencies is acyclic.

2.1. RDDL Representation of Reward and Dynamics

Following recent work on HSA-MDPs [Sanner *et al.*, 2012; Vianna *et al.*, 2015], we use the description language RDDL [Sanner, 2011] to specify HSA-MDPs. RDDL allows for relational templates to specify random variables which are instantiated via a set of objects. As a preprocessing step, we ground the templates with a concrete set of objects and expand relational quantifiers over their finite domains to obtain a propositional HSA-MDP as defined above.

The RDDL source code specifies the factored transition and reward functions through a sequence of assignment and sampling statements that define intermediate variables and next state variables, using algebraic expressions in these definitions. Recent work identified a fragment of RDDL that leads to a linear time reduction of finite horizon hybrid planning to Mixed Integer Linear Programs (MILP) [Raghavan *et al.*, 2017]. We use the same definition of stochastic piecewise linear (PWL) syntax for this work.

3 Online Hybrid Planning using HOP

Hindsight Optimization (HOP) [Chang *et al.*, 2000; Chong *et al.*, 2000] is a computationally attractive heuristic for online action selection. HOP approximates the optimal value function by optimally solving different realizations of random trajectories of the MDP called *futures* and aggregating their values. Despite its provable suboptimality in some cases, previous work has shown that it performs well in many benchmark probabilistic planning problems [Yoon *et al.*, 2008]. Recent work extended HOP-based algorithms by allowing continuous and discrete variables with state-action dependent stochasticity [Raghavan *et al.*, 2017].

Given a fixed policy, the MDP model induces a distribution over length-h trajectories. Viewing the choice of policy, and the random choices of the MDP as separate processes, we can make the random choices in advance, e.g., by fixing the seed for the random number generator. This selection which produces a random trajectory any given policy is known as a random future.

HOP requires sampling random futures and once the random choices are fixed, a future represents a deterministic planning problem. The optimal value of any state in the MDP is $V_h^*(s^0) = \max_\pi E_f[\sum_{t=0}^h R(s_f^t, \pi_f^t)]$, which is the maximum expected value over random futures f of length h. In contrast, the hindsight value $V_h^{hop}(s^0) = E_f[\max_{a_f^0,\dots,a_f^h} \sum_{t=0}^h R(s_f^t, a_f^t)]$ is the expected value of the optimal values of each future. Since a future is deterministic, the maximization is over *plans* instead of a policies.

HSA-HOP [Raghavan *et al.*, 2017] works by first sampling F futures, and then reducing the RDDL definition in linear time to a Mixed Integer Linear Program (MILP) over action variables $A_j^{f,t}$, where j indexes the action factor, f indexes the future and t indexes the time step. The MILP constraints express the transition functions of the next-state variables in the RDDL model. The objective function of the MILP is the h-step accumulated reward averaged over F futures.

3.1. HOP Variants

We employ three different variants of HOP described in [Raghavan *et al.*, 2017]. We evaluate all these variants through rolling horizon planning, i.e., by planning for h steps, executing the first action, and then replanning.

In the first variant called **Consensus** no other constraints are added to the above MILP, which means that the plan for each future is independent of other futures. An action is selected by majority vote (ties broken randomly) among the $A_j^{f,0}$ across the futures.

In the second variant called **HSA-HOP**, the MILP solutions for different futures are made dependent by adding an additional set of constraints to the MILP restricting the action variables at $t = 0$ to be identical across futures:

$$A_j^{f,0} = A_j^{0,0}, j = 1, \dots, m; f = 1, \dots, F \qquad (1)$$

When the constraints in Equation 1 are added, the MILP implements a one-step lookahead where the solutions are coupled by requiring the first action to be the same.

The third variant is called a **straight line plan** and is also known as an open loop policy or conformant plan. It commits to a sequence of future actions regardless of the probabilistic outcomes of earlier actions. This is achieved by replacing the constraint in Equation 1 with

$$A_j^{f,t}{=}A_j^{0,t}, j{=}1, \dots, m; f{=}1, \dots, F; t{=}0, \dots, h-1 \qquad (2)$$

The straight line value converges to the optimal value of the best open loop policy as the number of futures increases. Since in this case we are limiting the set of policies considered, the value of the optimal straight line plan is a lower bound on the optimal value of any state. Note that, although this formulation commits to an entire plan, in our evaluation we still use rolling horizon planning.

4 Emergency Domain : RDDL Encoding

In our formulation[1], each state presents a new emergency and the action consists of the set of responders to dispatch.

[1] https://tinyurl.com/yd22wrjo

State Space: The state space consists of (a) the emergency state $(x, y, t, \texttt{code})$, where the x, y coordinates in miles represent the location, the continuous variable $t > 0$ is a floating-point representation of time, and the nature code (`code`) is a discrete random variable that takes up to 15 distinct values, and (b) the responder state $(x(r), y(r), t_{\text{in}}(r), t_{\text{out}}(r), t_{\text{home}}(r))$ for each responder r, where (x, y) are the coordinates of the responder's last known location, t_{in} is the absolute time of arrival at the scene of the last emergency, and t_{out} and $t_{\texttt{home}}$ are the expected time of completion of its previous assignment and time of return to its home base respectively.

Action Space : In our simplified setup, each responder can fill exactly one of four *roles* namely Engine, Ambulance, Ladder and Command. Each MDP action is an assignment to boolean action variables of the form `dispatch(responder,role)`.

When a dispatch action is taken, the responder travels from $(x(r), y(r))$ to (x, y) at a constant speed. Upon arrival, the responder spends a constant amount of time at the scene that is determined by `code` (denoted by `stay(code)`). Depending on `code`, the responder must additionally proceed to one of two special locations (hospital or transfer) before the assignment is considered complete (denoted by $\texttt{dest}_x$ and $\texttt{dest}_y$). After a mandatory waiting period (denoted by `wait(code)`) at the end point of the assignment, it returns to its appointed (fixed) home base (denoted by $\texttt{home}_x$ `and` $\texttt{home}_y$). We allow a novel type of redeployment than prior work, specifically we allow a responder to be directly dispatched to a new emergency during this waiting period. We assume `wait(code)` and `stay(code)` are constants for any given `code`.

Travel Time : we assume that the travel time of a responder is a deterministic function of the distance between any two points. Following [Leathrum and Niedner, 2012], we use the city block distance ($d = |x_1 - x_2| + |y_1 - y_2|$ between any two points $(x_1, y_1), (x_2, y_2)$).

We model the travel speed as a piecewise constant function with 10 pieces, which leads to a PWL model of the travel time as shown in the following equations.

$$\begin{aligned}
d(r) &= |x(r) - x| + |y(r) - y| \\
t(r) &= \frac{d(r)}{38}(d(r) \leq 5) + \frac{d(r)}{42}(d(r) > 5 \wedge d(r) \leq 7) \\
&\quad + \ldots \\
t'_{in}(r) &= \texttt{if } (\exists_{s:\texttt{role}} \texttt{dispatch}(r, s)) \\
&\quad \texttt{then } t + t(r) \texttt{ else } t_{in}(r) \\
d_2(r) &= |\texttt{dest}_x\texttt{(x,code)} - x| + \\
&\quad |\texttt{dest}_y\texttt{(y,code)} - y| \\
t_2(r) &= \text{//Similar to } t(r) \text{ but using } d_2(r) \\
t'_{out}(r) &= \texttt{if } (\exists_{s:\texttt{role}} \texttt{dispatch}(r, s)) \texttt{ then } t'_{in}(r) \\
&\quad + \texttt{stay(code)} + t_2(r) \texttt{ else } t_{out}(r)
\end{aligned}$$

$$\begin{aligned}
d_3(r) &= |\texttt{home}_x\texttt{(r)-dest}_x\texttt{(x,code)}| + \\
&\quad |\texttt{home}_y\texttt{(r)-dest}_y\texttt{(y,code)}| \\
t_3(r) &= \text{//Similar to } t(r) \text{ but using } d_3(r) \\
t'_{home}(r) &= \texttt{if } (\exists_{s:\texttt{role}} \texttt{dispatch}(r, s)) \texttt{ then } t'_{out}(r) \\
&\quad + \texttt{wait(code)} + t_3(r) \texttt{ else } t_{home}(r)
\end{aligned}$$

Depending on the nature of the emergency, the function $\texttt{dest}_x\texttt{(x,code)}$ $(\texttt{dest}_y\texttt{(y,code)})$ returns the x-coordinate (y-coordinate, respectively) of the hospital, transfer location or simply returns x (y, respectively). The transition for $x(r)$ and $y(r)$ are analogous.

$$\begin{aligned}
x'(r) = \;& \texttt{if } (\exists_{s:\texttt{role}} \texttt{dispatch}(r, s)) \\
& \texttt{then } \texttt{dest}_x\texttt{(x,code)else if } (t > t_{home}(r)) \\
& \texttt{then } \texttt{home}_x(r)\texttt{else } x(r)
\end{aligned}$$

Response Time : Following [Bjarnason *et al.*, 2009], the performance of responders is evaluated in terms of *first response* and *full response*. The first response δ_1 is the time elapsed between the reporting of the emergency and the arrival of the first responder. The full response δ_2 is defined as the maximum of the time of arrival of all required responders. The requirements are predefined as a mapping from `code` to the number of required responders of each role and denoted as `required`. For example, Code-1 Medical emergency needs one ambulance. The specific constants we use are based on the Code3Sim simulator [Leathrum and Niedner, 2012].

We define the intermediate boolean variable `overwhelm` to check if the requirement of full response is not met. The `overwhelm` is set to True, if all required responders are not dispatched to the emergency scene and vice-versa. In the event of an overwhelm, the full response time is set to the maximum possible response time Δ = 20 Min . These can be represented by the following equations.

$$\begin{aligned}
\delta_1 = \min_r[&\texttt{if } (\exists_{s:\texttt{role}} \texttt{dispatch}(r, s)) \texttt{ then } t(r) \\
&\texttt{else } \Delta]
\end{aligned}$$

$$\begin{aligned}
\texttt{overwhelm} = \exists_{s:\texttt{role}}(&\texttt{required}(\texttt{code}, s) \\
&> \sum_r \texttt{dispatch}(r, s))
\end{aligned}$$

$$\begin{aligned}
\delta_2 = \;& \texttt{if (overwhelm) then } \Delta \texttt{ else} \\
& \max_r[(\exists_{s:role} \texttt{dispatch}(r, s)) \times t(r)]
\end{aligned}$$

Constraints : The following constraints enforce the availability of resources and forward flow of time.

$$\begin{aligned}
&\forall_{r:\texttt{responder}} \forall_{s:\texttt{role}}[\texttt{dispatch}(r, s) \Rightarrow [(t \geq t_{out}(r)) \\
&\qquad \wedge (t \leq t_{out}(r) + \texttt{wait(code)})] \\
&\qquad \vee (t \geq t_{home}(r))] \\
&[\forall_{r:\texttt{responder}} \forall_{s:\texttt{role}} \texttt{sum(dispatch(r,s)}] \leq \\
&\qquad \forall_{c:\texttt{code}} \texttt{sum(required(c))} \\
&\forall_{r:\texttt{responder}} \sum_{s:\texttt{role}} \texttt{dispatch}(r, s) \leq 1 \\
&\forall_{r:\texttt{responder}} 0 \geq t_{in}(r) \geq t_{out}(r) \geq t_{free}(r)
\end{aligned}$$

The first constraint is an action precondition to ensure that the responder is either at the scene after completion of the previous assignment (tarry) or is idling at the base station. The second constraint ensures that the number of dispatching responders is less than or equal to the required number for the nature code c. The third constraint ensures that each responder is assigned at most one role. Finally, the fourth constraint ensures the validity of the timers for each responder.

5 Dataset & Sampling Futures

The online planning experiments are carried out using emergency data for Corvallis, USA (pop. $\approx$ 54k). The dataset has historical emergency calls from 2007 to 2011($\approx$ 1825 days, $\approx$ 35,222 calls), each call consists of (x, y) :location, t : time, and $severity$: type of the emergency.

Recall that the HOP variants work by sampling random futures. Importantly, the factored transition of the emergency state is conditionally independent of the transition of the responder's state. For a random future, given an initial emergency state $(x, y, t, code)$ chosen from the data, the corresponding next emergency state (t') is obtained by a two-step process: (1) build a list of next closest emergencies for each day, i.e, the first emergency that occurs after time (t%24) on that day, and (2) sample one emergency uniformly randomly from this list and return it. This sampling approach assumes emergencies to be independent across days, and allows the creation of synthetic data similar to the real world data without excessive modeling effort.

6 Experiments

Baseline: As mentioned earlier, there are no domain-independent MDP algorithms that can scale to hybrid state and action spaces of high dimensions. Our reference baseline will be the dispatch-closest policy that is used extensively in the literature. The policy works as follows : for each code of current emergency call (Engine, Ambulance, Ladder, Command), the Policy dispatches the closest required responder if the responder is available in service; otherwise it dispatches a random responder. The required number is specified by `required(code)`. The closest responders are identified using the travel time.

Setup : The evaluation is done in online replanning mode, i.e., planning is repeated at every world state and one action is output. The average accumulated reward is evaluated by running 30 trails on a fixed test data (240 calls between Jan 1-15, 2010). The data used for generating futures is sampled from 2400 calls between June 1 to Dec 31, 2010 unless otherwise stated. Each evaluation has three experimental parameters: (1) time limit t per decision in seconds, (2) the lookahead depth (h) = the length of sampled futures, and (3) the number of sampled futures (F) per decision.

In practice, the time limit must be small. In the current scenario, the dispatch must be within two minutes of the received call. We found that for small values of h and F, Gurobi (version 7.5) solves the resulting MILPs optimally for many states in under two minutes.

Multiple Objectives: we compare the real world performance for different choices of the reward function. We employ different linear combinations of δ_1 and δ_2 as $\texttt{reward} = (1-\theta)\delta_1 + \theta\delta_2$, for $0 \leq \theta \leq 1$. These two objectives oppose each other because they are coupled via resource constraints.

Figure 1 shows the performance (first response, full response) of baseline and the three HOP-based algorithms with 95% confidence interval. We used 5 responders for this experiment, namely a Command, Engine, Ladder and two Ambulances, lookahead $h = 4$ and $F = 5$ futures with 120 seconds being the maximum time allowed for optimization.

The left panel shows that the average first response time of three HOP variants are significantly smaller (by about $\approx$ 1.5 min) for $\theta = 0$ case compared to baseline, whereas the full response time has similar performance to the closest dispatch policy. The right panel shows overwhelm rates, i.e. the fraction of calls left unanswered. The figure denotes *complete overwhelms* ($\delta_1 = \delta_2 = \Delta$) whenever no responder is dispatched in contrast to a *partial overhwelm* when $\delta_2 = \Delta$ and the emergency is not fully satisfied. We observe that the complete overwhelm percentage is smaller, from about 5% of calls for the baseline compared to less than 0.2% with HSA-HOP, Straight Line and Consensus except for $\theta = 1$. The fraction of partial overwhelms is similar in percentage compared to the baseline.

Scaling with Fleet Size: Figure 2 shows the performance of the three HOP variants and the baseline when the number of responders is increased to 8, consisting of a Command, three Ambulances, two Engines and two Ladders. We observe a larger improvement with respect to the baseline policy for all three HOP-based algorithms. The mean first response time shows ($\sim$ 1.3 min) an improvement for $\theta = 0$ case and the full response time shows an $\sim$ 2.1 min improvement. Similarly, the overwhelm rates of MILP algorithms show significant improvements compared to the baseline policy as shown in the right panel of the figure.

Scaling with Futures: In this experiment, we increased the number of futures linearly by fixing lookahead(h) = 4. Table 1 shows the performance of closest dispatch policy and three variants of HOP using 5 and 8 responders. For the case with 5 responders, we did not observe any improvements in baseline policy, whereas, for 8 responders, the HOP algorithms have shown a significant improvement in test reward $((1-\theta)\delta_1 + \theta\delta_2)$($\sim$ 1.3 min) for all the future values. On the other hand, the Consensus Policy has shown an improvement in test reward as the futures are increased and shows its best performance at 20 futures. The HSA-HOP policy performs better for lower number of futures, whereas, the Straight Line policy with 8 responders has shown the best performance at 20 futures.

Scaling with Lookahead: Table 2 shows the performance of the three HOP based algorithms with increasing lookahead steps. For Consensus and HSA-HOP policy with 5 responders, we observe a slight improvement in performance with increasing lookahead from 1 to 4. They did not show improvement after 4 lookahead steps. Moreover, the performance of Straight Line policy with 5 responders also shows no improvement. For responders with 8 vehicles, we observe that lookahead does not improve and performs badly in all the three HOP variants. HOP algorithms perform best for myopic planning (i.e lookahead=1), suggesting that deep lookahead is

V	Policy	F	First Response (δ_1) (Min.)	Full_Response ((δ_2) (Min.)	Test Reward (Min.)	Complete Overwhelm (%)	Partial Overwhelm (%)
5	Closest Dispatch	-	6.117(0.01)	9.38(0.12)	7.748(0.015)	5.00	27.9
	Consensus	5	6.071(0.08)	9.253(0.139)	7.662(0.110)	0.747	25.144
		10	6.017(0.062)	9.092(0.137)	7.5545(0.100)	0.805	23.506
		15	5.999(0.067)	9.124(0.148)	7.5615(0.108)	0.718	23.463
		20	**5.983(0.054)**	9.049(0.123)	**7.516(0.089)**	**0.733**	**23.06**
	HSA-HOP	**5**	**6.21(0.05)**	**9.057(0.161)**	**7.6335(0.106)**	**0.374**	**22.299**
		10	6.288(0.051)	9.082(0.104)	7.685(0.078)	0.287	22.155
		15	6.334(0.075)	9.16(0.174)	7.747(0.116)	0.187	22.241
		20	6.378(0.118)	9.329(0.151)	7.8535(0.159)	4.325	26.394
	Straight Line	**5**	**6.363 (0.119)**	**9.239(0.236)**	**7.801 (0.178)**	**0.963**	**24.397**
		10	6.457 (0.075)	9.218(0.171)	7.837(0.123)	1.135	23.534
		15	6.496(0.084)	9.192(0.119)	7.844(0.102)	1.437	23.089
		20	6.443(0.143)	9.199(0.217)	7.821(0.180)	8.563	28.865
8	Closest Dispatch	-	6.063(0.02)	9.016(0.13)	7.539(0.10)	4.58	25.8
	Consensus	5	5.774(0.048)	6.796(0.08)	6.285(0.06)	0	4.698
		10	5.785(0.053)	6.746(0.075)	6.2655(0.05)	0	3.994
		15	**5.743(0.049)**	**6.689(0.05)**	**6.216(0.04)**	**0**	**3.764**
		20	5.745(0.03)	6.692(0.046)	6.2185(0.05)	0	3.708
	HSA-HOP	5	5.801(0.05)	6.821(0.096)	6.311(0.07)	0.029	4.497
		10	5.894(0.05)	6.997(0.108)	6.4455(0.08)	0	5.146
		15	5.945(0.09)	7.061(0.138)	6.503(0.114)	0.029	5.342
		20	**5.63(0.196)**	**6.866(0.208)**	**6.248(0.202)**	**0.205**	**5.121**
	Straight Line	5	5.864(0.088)	7.022(0.15)	6.443(0.119)	0.101	6.063
		10	5.892(0.07)	7.025(0.129)	6.4585(0.1)	0.014	5.651
		15	5.909(0.058)	6.938(0.122)	6.4235 (0.09)	0.058	4.873
		20	**5.682(0.141)**	6.768((0.161)	**6.225(0.151)**	**0.036**	**4.223**

Table 1: Performance variation with Futures (F) with Vehicles (V) = 5, 8 using fixed lookahead (h) = 4, fixed θ = 0.5 and timeout = 120 sec.

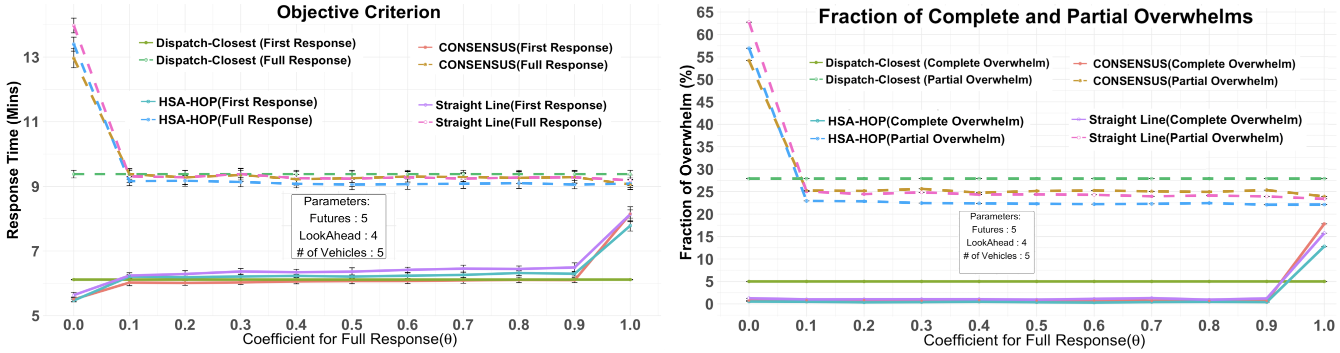

Figure 1: Performance with 5 responders and $F = 5$ futures under different reward functions of the form $(1-\theta)\delta_1 + \theta\delta_2$. The calls are randomly sampled from 240 calls in June 1 - Dec 31, 2010 for planning. Testing data is a fixed sequence of 240 emergencies between Jan 1-15, 2010.

not needed when there are plenty of resources.

Stress analysis by increasing Emergency calls: In this experiment, we increased the number of calls per day by a factor of ~ 2. This was a reasonable factor to consider given that we observed the maximum number of emergencies to be $\approx$ 50 in our dataset. The increase in calls is achieved by learning a model from past 5 years of data (35,222 emergency calls). The procedure followed for generating emergency calls $(x, y, t, Code)$ is a three step process: (a) location variables (x, y) of emergency calls (N = 10,000 samples) were sampled using a fitted Gaussian distribution with parameters μ_x = 1416.138, σ_x = 2.69, μ_y = 64.27 and σ_y = 3.88. (b) For sampling the time of the event, we used a derived random variable called GapTime (absolute time gap between t and $t+1$ event) which was sampled from a Weibull distribution (shape (λ) = 0.914, scale (k) = 0.3). (c) We use a multivariate adaptive regression splines for learning the nature code using location and GapTime as independent features for the model. Table 3 shows the performance comparison of the HOP algorithms and the baseline with 8 responders by varying lookahead under stressed emergency calls. We observe that all the three variants of HOP are performing significantly better than the baseline, closest dispatch policy, in terms of test reward ($\sim$ 2.1 min) and fraction of unanswered calls ($\sim$ 20%). We observe that the consensus performance improves with lookahead steps, whereas the other two variants show no sign of improvement.

7 Discussion

We presented an MDP formulation for the emergency response problem consisting of a hybrid state space and a discrete action space. We demonstrated the application of recently developed domain-independent and online planning algorithms for this problem. Our results show that online planners perform significantly better compared to heuristic policy in a significant region of domain and planner parameters. The overall best performance is achieved with 8 responders in terms of test reward ($\sim$ 1.3 min) and the number of overwhelms ($\sim$ 22 % reduction compared to the closest dispatch).

Overall, all the three variants of HOP perform well with 8 responders compared to 5 responders. With 8 responders, the planners can easily send the required vehicles to the emer-

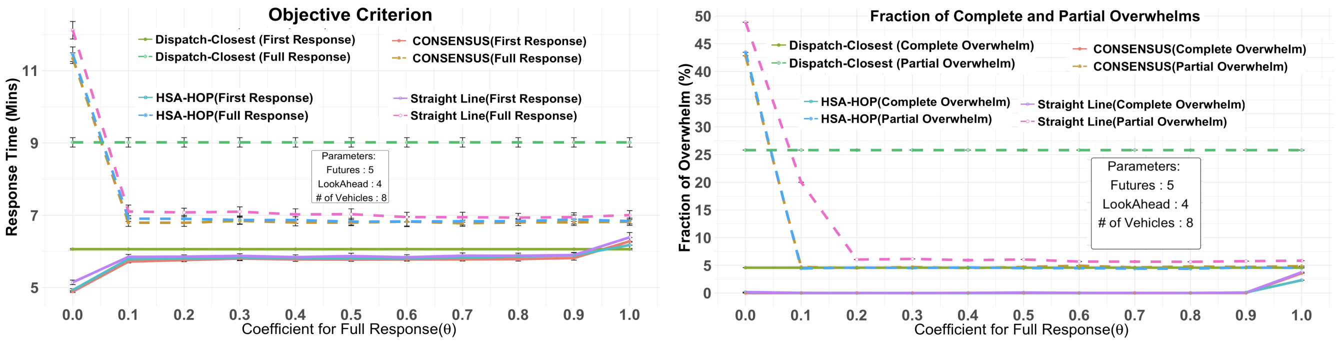

Figure 2: Performance with 8 responders and $F = 5$ futures under different reward functions of the form $(1-\theta)\delta_1 + \theta\delta_2$. Training data for HSA-HOP is randomly sampled from 240 calls in June 1 Dec 31, 2010. Testing data is a fixed sequence of 240 emergencies between Jan 1-15, 2010.

V	Policy	L	First Response (δ_1) (Min.)	Full Response (δ_2) (Min.)	Test Reward (Min.)	Complete Overwhelm (%)	Partial Overwhelm
5	Closest Dispatch	-	6.117(0.01)	9.38(0.12)	7.748(0.015)	5.00	27.9
	Consensus	1	6.046(0.01)	9.393(0.0)	7.72(0.005)	0.417	23.333
		2	6.18(0.06)	9.312(0.159)	7.746(0.109)	1.02	24.468
		4	**6.071(0.08)**	**9.253(0.139)**	**7.662(0.11)**	**0.747**	**25.144**
		8	6.039(0.104)	9.339(0.217)	7.689(0.16)	0.719	25.213
	HSA-HOP	1	6.12(0.033)	9.402(0.008)	7.761(0.02)	0.603	23.75
		2	6.224(0.055)	9.2(0.15)	7.712(0.103)	0.718	22.457
		4	**6.21(0.05)**	**9.057(0.161)**	**7.633(0.106)**	**0.374**	**22.299**
		8	6.241(0.073)	9.11(0.14)	7.676(0.107)	0.273	22.519
	Straight Line	**1**	**6.123(0.033)**	**9.403(0.007)**	**7.763(0.02)**	**0.618**	**23.75**
		2	6.291(0.076)	9.363(0.166)	7.827(0.121)	1.135	23.937
		4	6.363(0.119)	9.239(0.236)	7.801(0.177)	0.963	24.397
		8	6.474(0.104)	9.623(0.165)	8.049(0.134)	1.237	26.46
8	Closest Dispatch	-	6.063(0.02)	9.016(0.13)	7.539(0.10)	4.58	25.8
	Consensus	**1**	**5.706(0.012)**	**6.644(0.018)**	**6.175(0.015)**	**0**	**3.75**
		2	5.701(0.031)	6.669(0.053)	6.185(0.042)	0	4.009
		4	5.774(0.048)	6.796(0.085)	6.285(0.067)	0	4.698
		8	5.808(0.058)	6.827(0.1)	6.318(0.079)	0.014	4.62
	HSA-HOP	**1**	**5.719(0.013)**	**6.654(0.013)**	**6.187(0.013)**	**0**	**3.75**
		2	5.757(0.042)	6.791(0.092)	6.274(0.067)	0	4.54
		4	5.805(0.05)	6.825(0.094)	6.315(0.072)	0.029	4.497
		8	5.873(0.071)	6.899(0.13)	6.386(0.1)	0	4.452
	Straight Line	**1**	**5.723(0.021)**	**6.658(0.02)**	**6.19(0.02)**	**0**	**3.75**
		2	5.834(0.071)	6.922(0.161)	6.378(0.116)	0	5.187
		4	5.872(0.08)	7.03(0.148)	6.451(0.114)	0.101	6.063
		8	5.968(0.047)	7.128(0.117)	6.548(0.082)	0.043	6.04

Table 2: Performance vs. increasing steps of lookahead with $V = 5$ and $V = 8$ vehicles, using $F = 5$ futures and timeout of 120 seconds.

Policy	(h)	First Response (δ_1)	Full Response (δ_2)	Test Reward (Min.)	Complete Overwhelm (%)	Partial Overwhelm (%)	MILP Gap(%)
Closest Dispatch	-	7.953(0.01)	13.35(0.01)	10.65(0.01)	5.00	46.25	-
Consensus	1	7.418(0.025)	11.13(0.01)	9.274(0.018)	0.417	27.902	0.0615
	2	7.266(0.083)	11.147(0.095)	9.207(0.089)	0.043	29.598	0.3837
	4	**6.828(0.104)**	**11.48(0.145)**	**9.154(0.124)**	**0.144**	**34.871**	**0.5245**
HSA-HOP	**1**	**7.398(0.018)**	**11.131(0.007)**	**9.265(0.013)**	**0.417**	**27.917**	**0.0544**
	2	7.419(0.107)	11.456(0.091)	9.437(0.099)	0.216	30.072	0.3819
	4	7.368(0.092)	11.502(0.14)	9.435(0.116)	0.316	30.761	0.5592
Straight Line	**1**	**7.398(0.02)**	**11.13(0.005)**	**9.264(0.013)**	**0.417**	**27.917**	**0.0515**
	2	7.02(0.089)	11.765(0.16)	9.392(0.124)	0.302	35.79	0.3086
	4	6.032(0.113)	13.05(0.186)	9.541(0.149)	0.46	52.601	0.6949

Table 3: Performance vs Increasing steps with V = 8 for Stressed emergency calls data using F= 5 futures and timeout of 120 seconds.

gency spot, thus showing a huge improvement in terms of unanswered calls. The MILP planners with more responders can use the flexibility of sending fewer vehicles and keep some of the vehicles in available state for serving future emergencies, whereas with fewer responders, the planners have less decisive power to choose between future emergencies and the current emergency. Therefore, with 5 responders, the performance is similar to that of the closest dispatch policy.

We found that, Consensus and HSA-HOP policy perform similarly with fewer futures (F=5), whereas the Straight Line policy has poor performance for all futures values. This is not surprising due to its demanding action constraints. We found that HSA-HOP policy does produce a small fraction (~ 5 %) of infeasible solutions at 20 futures. This is due to the large optimization problem (with 190,000 variables and 400,000 constraints). We also found that Consensus policy with 8 responders shows a strong consensus ratio of $\sim 79\%$ at 5 futures and the ratio drops down to $\sim 70\%$ for 20 futures. The variation of lookahead did not improve performance significantly. This can be attributed to high uncertainty of emergency calls $(x, y, t, code)$. For large lookahead steps, we found that the planners take suboptimal actions for the current emergency by sending high response time vehicles. But, this sacrifice is not always helpful due to the stochastic nature of future emergencies. The experiment with increasing the frequency of emergency calls reinforces the success of using online planners. The planners show a significant increase in performance in terms of test reward (~ 1.4 min) as well as partial overwhelm rate ($\sim$ 20 %) except for the Straight Line policy at lookahead = 4. The difference in full response time between the HOP variants and the closest dispatch is ~ 2.2 min, which shows that, unlike the baseline policy, the HOP-based planners are able to save some vehicles for potential future dispatches by not dispatching them to satisfy the current request.

Acknowledgements

This work is supported by NSF under grant nos. IIS-1619433 and CNS-1406049 and by DARPA under grant no. N66001-17-2-4030. We thank Carl Niedner of Coele Company of Design and the Corvallis Fire Department for their advice and sharing of data.

References

[Aringhieri *et al.*, 2016] R Aringhieri, ME Bruni, S Khodaparasti, and JT van Essen. Emergency Medical Services and Beyond: Addressing New Challenges Through a Wide Literature Review. *Computers & Operations Research*, 2016.

[Azizan *et al.*, 2013] Mohd Hafiz Azizan, Cheng Siong Lim, Go TL Hatta WALWM, and SS Teoh. Simulation of Emergency Medical Services Delivery Performance Based on Real Map. *International Journal of Engineering and Technology*, 5(3):2620–2627, 2013.

[Bélanger *et al.*, 2015] Valérie Bélanger, Angel Ruiz, and Patrick Soriano. *Recent Advances in Emergency Medical Services Management*. Tech. Rep. CIRRELT-2015-28, CIRRELT, 2015.

[Bjarnason *et al.*, 2009] Ronald Bjarnason, Prasad Tadepalli, Alan Fern, and Carl Niedner. Simulation-based optimization of resource placement and emergency response. In *IAAI*, 2009.

[Brotcorne *et al.*, 2003] Luce Brotcorne, Gilbert Laporte, and Frederic Semet. Ambulance Location and Relocation Models. *European journal of operational research*, 147(3):451–463, 2003.

[Chang *et al.*, 2000] H. S. Chang, R. L. Givan, and E. K.P. Chong. Online Scheduling via Sampling. In *Proceedings of the Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[Chong *et al.*, 2000] E. K.P. Chong, R. L. Givan, and H. S. Chang. A Framework for Simulation-Based Network Control via Hindsight Optimization. In *IEEE CDC '00: IEEE Conference on Decision and Control*, 2000.

[Dean, 2008] Stephen F Dean. Why the Closest Ambulance Cannot be Dispatched in an Urban Emergency Medical Services System. *Prehospital and Disaster Medicine*, 23(02):161–165, 2008.

[Goldberg, 2004] Jeffrey B Goldberg. Operations Research Models for the Deployment of Emergency Services Vehicles. *EMS management Journal*, 1(1):20–39, 2004.

[Haghani *et al.*, 2004] Ali Haghani, Qiang Tian, and Huijun Hu. Simulation Model for Real-Time Emergency Vehicle Dispatching and Routing. *Transportation Research Record: Journal of the Transportation Research Board*, (1882):176–183, 2004.

[Jagtenberg *et al.*, 2015] CJ Jagtenberg, Sandjai Bhulai, and RD van der Mei. An Efficient Heuristic for Real-Time Ambulance Redeployment. *Operations Research for Health Care*, 4:27–35, 2015.

[Kveton *et al.*, 2006] Branislav Kveton, Milos Hauskrecht, and Carlos Guestrin. Solving Factored MDPs with Hybrid State and Action Variables. *J. Artif. Intell. Res.(JAIR)*, 27:153–201, 2006.

[Leathrum and Niedner, 2012] Roland Leathrum and Carl Niedner. Corvallis Fire Department Response Time Simulation. 2012.

[Maxwell *et al.*, 2014] Matthew S Maxwell, Eric Cao Ni, Chaoxu Tong, Shane G Henderson, Huseyin Topaloglu, and Susan R Hunter. A Bound on the Performance of an Optimal Ambulance Redeployment Policy. *Operations Research*, 62(5):1014–1027, 2014.

[McCormack and Coates, 2015] Richard McCormack and Graham Coates. A Simulation Model to Enable the Optimization of Ambulance Fleet Allocation and Base Station Location for Increased Patient Survival. *European Journal of Operational Research*, 247(1):294–309, 2015.

[Raghavan *et al.*, 2012] Aswin Raghavan, Saket Joshi, Alan Fern, Prasad Tadepalli, and Roni Khardon. Planning in Factored Action Spaces with Symbolic Dynamic Programming. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Raghavan *et al.*, 2017] Aswin Raghavan, Scott Sanner, Roni Khardon, Prasad Tadepalli, and Alan Fern. Hindsight Optimization for Hybrid State and Action MDPs. 2017.

[Sanner *et al.*, 2012] Scott Sanner, Karina Valdivia Delgado, and Leliane Nunes De Barros. Symbolic Dynamic Programming for Discrete and Continuous State MDPs. *arXiv preprint arXiv:1202.3762*, 2012.

[Sanner, 2011] S. Sanner. Relational Dynamic Influence Diagram Language (rddl): Language Description. *Unpublished ms. Australian National University*, 2011.

[Vianna *et al.*, 2015] Luis GR Vianna, Leliane N De Barros, and Scott Sanner. Real-Time Symbolic Dynamic Programming for Hybrid MDPs. 2015.

[Yoon *et al.*, 2008] S. Yoon, A. Fern, R. Givan, and S. Kambhampati. Probabilistic Planning via Determinization in Hindsight. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 2, pages 1010–1016, 2008.