Improving Adjustable Autonomy Strategies for Time-Critical Domains

Nathan Schurr
Aptima, Inc.
nschurr@aptima.com

Janusz Marecki
IBM T. J. Watson Research Center
marecki@us.ibm.com

Milind Tambe
University of Southern California
tambe@usc.edu

ABSTRACT

As agents begin to perform complex tasks alongside humans as collaborative teammates, it becomes crucial that the resulting human-multiagent teams adapt to time-critical domains. In such domains, adjustable autonomy has proven useful by allowing for a dynamic transfer of control of decision making between human and agents. However, existing adjustable autonomy algorithms commonly discretize time, which not only results in high algorithm runtimes but also translates into inaccurate transfer of control policies. In addition, existing techniques fail to address decision making inconsistencies often encountered in human multiagent decision making. To address these limitations, we present novel approach for Resolving Inconsistencies in Adjustable Autonomy in Continuous Time (RIAACT) that makes three contributions: First, we apply continuous time planning paradigm to adjustable autonomy, resulting in high-accuracy transfer of control policies. Second, our new adjustable autonomy framework both models and plans for the resolving of inconsistencies between human and agent decisions. Third, we introduce a new model, Interruptible Action Time-dependent Markov Decision Problem (IA-TMDP), which allows for actions to be interrupted at any point in continuous time. We show how to solve IA-TMDPs efficiently and leverage them to plan for the resolving of inconsistencies in RIAACT. Furthermore, these contributions have been realized and evaluated in a complex disaster response simulation system.

1. INTRODUCTION

Teams composed of humans and agents are gradually being leveraged to address the uncertain, time-critical domains of the real world. In such domains, adjustable autonomy [9] has proven useful by allowing for a dynamic transfer of control of decision making between human and agents. Adjustable autonomy has been applied in domains ranging from disaster response[10] to multi-robot control [11] to future-generation offices [2]. In situations where agents lack the global perspective or general knowledge to attack a problem, or the capability to make key decisions, adjustable autonomy enables agents to access the often superior decision-making capabilities of humans while ensuring that they are not bothered for routine decisions.

This paper focuses on adjustable autonomy in uncertain, time-critical domains. Often the domain complexity and distribution of information implies that humans may sometimes provide poor input to agents. In such domains, it may be impossible to provide the human with a timely accurate local perspective of individual agents in the team. An example of this is seen when adjustable autonomy was used in disaster response simulations [10]. While obtaining human input allowed the agent team to often improve its performance, in several cases the human input degraded overall team performance, and the agent team would have been better off if it had ignored the human and acted autonomously. Analysis revealed that the agent team had improved local information that was unavailable to the human participant in the disaster response simulations. In other words, there was an inconsistency between the agents' local and the human's global perspective. However, this inconsistency does not always imply that the agent team's decisions are superior, and the human input may not uniformly improve or degrade agent team's performance. This uncertainty poses a challenge for the agent team because it cannot simply accept or reject human input. Furthermore, if the agent team were to try and resolve the inconsistency, because the length of time for resolving may be long, the team may want to interrupt due to approaching deadlines.

Previous work in adjustable autonomy [9, 12] has failed to address these issues in time-critical domains. For example, previous work has relied on Markov Decision Problem (MDP) and Partially Observable MDP (POMDP) for planning interactions with humans [9, 12]. While successful in domains such as office environments [9], they fail when facing time-critical adjustable autonomy due to three major challenges.

First, adjustable autonomy planning has, until now, focused on transfer of control strategies that end once a decision has been made. However, in realistic, uncertain domains, human multiagent team performance can be improved by the detection and resolution of inconsistencies between human and agent decisions. Second, previous work has utilized discrete-time planning approaches, which are highly problematic given highly uncertain action durations and deadlines. For example, the task of resolving the inconsistency between a human and an agent takes an uncertain amount of time. Given deadlines, the key challenge is how long to attempt to resolve such inconsistency and whether to attempt a resolution in the first place. Discrete time planning with coarse-grained time intervals may lead to significantly lower quality in adjustable autonomy because the policy may miss a critical opportunity. Planning with very fine grained intervals unfortunately causes a state space explosion, and greatly increases solution runtimes. Third, previous work in adjustable autonomy has not allowed actions to be interruptible. Actions in many domains are often started and then interrupted if they do not finish on time or if another action becomes profitable.

This paper addresses these challenges with an approach called

Cite as: Improving Adjustable Autonomy Strategies for Time-Critical Domains, Nathan Schurr, Janusz Marecki, Milind Tambe, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 353–360

RIAACT: Resolving Inconsistencies in Adjustable Autonomy in Continuous Time. RIAACT makes three contributions. Firstly, RIAACT presents an extended adjustable autonomy model that incorporates planning into the strategy in order to overcome inconsistencies between the human and the agents. This allows the agents to avoid a potentially poor input from the human. The aim of this paper is to develop an overarching framework that will work with any inconsistency resolution method that is chosen, given its estimated action duration.

Secondly, RIAACT creates adjustable autonomy policies with precisely timed transfers of control. RIAACT leverages recent work in Time-Dependent Markov Decision Problems (TMDPs) [5, 6]. By exploiting the fastest current TMDP solution technique, we have constructed an efficient policy solver. The result is a continuous time policy that allows for actions to be prescribed at arbitrary points in time, without the state space explosion that results from using fixed discrete intervals.

Thirdly, we introduce a new planning framework that allows for actions to be interruptible, referred to as an Interruptible Action TMDP (IA-TMDP). Also, we will show concretely how to model time-critical adjustable autonomy within the IA-TMDP framework. Current TMDP techniques allow for the starting of an action at any point in continuous time. Unfortunately, if started, that action must be pursued until completion or else a new state must be created for each point in time at which the action maybe interrupted thereby causing a state space explosion. Interruptible actions enable IA-TMDPs to provide higher quality policies, which include not only accurate policies for the given states, but also an additional time-dependent policy for each interruptible action. The difficulty in implementing interruptible actions is to have the interruption be able to occur at any point in continuous time without an explosion in the state space. RIAACT avoids this state space explosion in IA-TMDPs. RIAACT incorporates these techniques into a practical solution for human-multiagent teams. We illustrate RIAACT's benefits with experiments in a complex disaster response simulation.

2. BACKGROUND

This section provides a brief background for RIAACT in the areas of adjustable autonomy and TMDPs. RIAACT extends existing work in adjustable autonomy and develop a new model that extends TMDPs to address the constraints of the new adjustable autonomy models.

2.1 Adjustable Autonomy

Early work in mixed-initiative and adjustable autonomy interactions often focused on one-shot autonomy decisions. Such permanent transfers over decisions were highly problematic given uncertain human response times in time-critical domains, later work focused on carefully planning adjustable autonomy interactions, that allowed for back-and-forth transfer of control [9, 12, 4, 11]. For example, using MDPs or POMDPs for adjustable autonomy [9, 12] results in policies for agents to transfer control to a human and then if a human decision is not made by a certain time, to take control back or take a postpone action to increase the time available for human decision making.

There are weaknesses with these prior adjustable autonomy approaches. First, the planning approaches did not allow for error checking or discussion once a decision had been made. For example, in [10] experiments were conducted where a human-multiagent team consisting of fire fighter agents interacted with a human in order to extinguish fires that were quickly spreading through a dense urban environment. However, at times adjustable autonomy policies that incorporated human input ended up performing worse than if the agent team had acted autonomously. Further analysis revealed that the agents had local priorities that were inconsistent with human inputs.

Second, in order to ensure high-quality decisions, prior work has used very fine-grained discretization in planning, but the result was a large state space, with consequently longer run times (as seen for two time slices in Figure 1). For example, [9] reports thousands of MDP states for planning interactions for a single office meeting. In time-critical domains such as the disaster response domain of interest in this paper, such a state-space explosion is unacceptable because a policy must be determined quickly and executed during the ongoing event.

2.2 Time Dependent MDPs (TMDPs)

In many realistic domains, agents execute actions whose durations are uncertain and can only be characterized by continuous probability density functions. A common approach to model such domains with continuous time characteristics has been to use the framework of semi-Markov Decision Process [8]. However, semi-MDP policies are not indexed by the current time and as such, might not have the desired expressivity when dealing with time-critical domains. For such domains, one can use the Time dependent MDP (TMDP) model [1]. TMDP's approach to modeling continuous time is to create a finite, hybrid state-space where each discrete state has a corresponding continuous time dimension. This allows TMDP policies to be both accurate and conditioned on time limits, as required in our domains.

The TMDP model [1] is defined as a tuple $\langle S, A, P, D, R\rangle$ where S is a finite set of discrete states and A is a finite set of actions. P is the discrete transition function, i.e., $P(s, a, s')$ is the probability of transitioning to state $s' \in S$ if action $a \in A$ is executed in state $s \in S$. Furthermore, for each combination of s, a, s' there is a corresponding probability density function $d_{s,a,s'} \in D$ of action duration, i.e., $d_{s,a,s'}(t)$ is the probability that the execution of action a from state s to state s' takes time t. Also, R is the time-dependent reward function, i.e., $R(s, a, s', t)$ is the reward for transitioning to state s' from state s via action a completed at time t. As such, the earliest time Δ after which no reward can be earned for any action is referred to as the *deadline*. Finally, a policy π for a TMDP is a mapping $S \times [0, \Delta] \rightarrow A$ and the optimal policy π^* assigns the optimal action $\pi^*(s, t)$ to each pair $s \in S; t \in [0, \Delta]$. The expected utilities of following a policy π from state s at time t are then denoted as $U^\pi(s, t)$ and can be viewed as continuous *value functions* over the time interval $[0, \Delta]$ for each discrete state $s \in S$.

Recently, there has been a significant progress on solving TMDPs [1, 5, 6]. The primary challenge that any TMDP solver must address is how to perform the value iteration given that the time dimension is continuous. Consequently, each TMDP solution technique must balance the tradeoffs of the algorithm run time and the quality of the solution. In our implementation of RIAACT we have chosen to extend the Continuous Phase (CPH) solver [6], which is currently the fastest TMDP solver.

3. RIAACT APPROACH

RIAACT is focused on realistic domains that are time-critical, contain uncertain action durations, and inconsistency arises between human and agent decisions. RIAACT aims to provide an overarching framework for modeling adjustable autonomy in these challenging, realistic domains as well as to demonstrate an efficient technique for solving problems modeled in this framework. In describing this framework we will first propose a new RIAACT model for richer adjustable autonomy policies (see Figure 2). Next, in or-

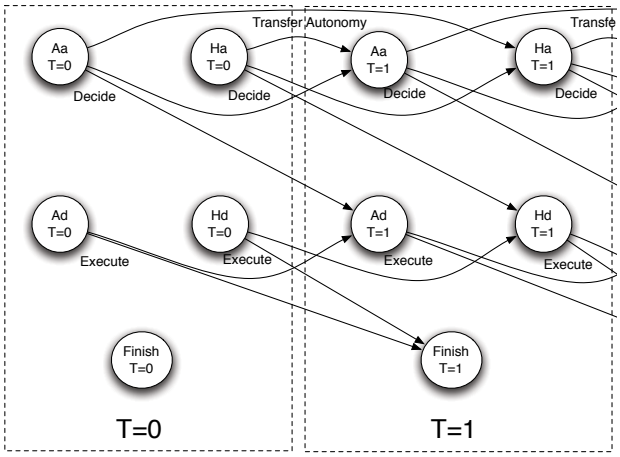


Figure 1: Previous adjustable autonomy discretized model.

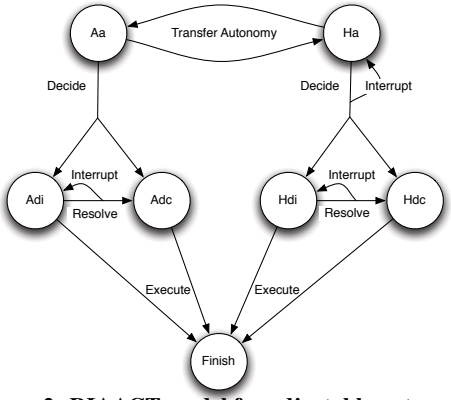


Figure 2: RIAACT model for adjustable autonomy.

der to address this new RIAACT model, we will present an Interruptible Action Time dependent MDPs (IA-TMDP) model. Lastly, we will describe how we have efficiently applied IA-TMDPs to the RIAACT model.

3.1 Extended Adjustable Autonomy Model

In order to address the challenges brought about by dealing with time-critical domains, we have created a new model for adjustable autonomy policies. Figure 1 shows the abstracted states of the MDP for solving the adjustable autonomy problem as it was implemented in previous work [9, 10]. Each dashed-line box represents a discretized time slice at T=0 and T=1. As time is broken into more and more intervals, more states and transitions will be added. This figure represents a single team decision and each circle represents a state that the decision can be in.

The RIAACT model (Figure 2) improves on the previous model of adjustable autonomy (Figure 1) in three important aspects: (i) the reasoning about and modeling of the resolving of inconsistencies, (ii) action durations (the arrows in the diagram) are now modeled by continuous distributions, and (iii) allowing for the interruption of actions in continuous time. Instead of the number of states increasing to model the same states over time, single states now have policies that are functions over time. In addition, each arrow in Figure 2 represents not a constant duration, but an entire action duration distribution that can be any arbitrary distribution. Note that this model represents a single team decision, and one of these would be instantiated for each team decision.

States - Each circle in Figure 2 represents a state that a team decision can be in. We assume that each state is fully observable. In order to create a clear and concise example, broad state categories are used. Each of these state categories can be broken into subcategories to more accurately model the world. For instance, the state of an inconsistent human decision, Hdi, can be split into several possible inconsistent states, each with their own reward, or just be modeled as a single state with an average reward. The RIAACT model in Figure 2 represents a single team decision in one of the following states: (i) *Agent has autonomy (Aa)* - The agent team has autonomy over the decision. At this point, the agent team can either transfer control to the human or try to make a decision. (ii) *Human has autonomy (Ha)* - Human has the autonomy over the decision. At this point, the human can either transfer control to an agent or make a decision. (iii) *Agent decision inconsistent (Adi)* - The agent has made a decision and the human disagrees with that decision. (iv) *Agent decision consistent (Adc)* - The agent has made a decision and the human agrees with that decision. (v) *Human decision inconsistent (Hdi)* - The human has made a decision and the agent believes that the decision will result in substantial decrease in average reward for the team. The agent may determine this by factoring in its detailed local information when predicting the results of implementing the human decision. (vi) *Human decision consistent (Hdc)* - The human has made a decision and the agent believes that the decision will either increase the reward for the team or does not have enough information to raise an inconsistency. (vii) *Task finished (Finish)* - The task has been completed and a reward has been earned. The reward varies based on which decision was executed.

Actions - The arrows in Figure 2 represent the actions that enable state transitions. However, now in the RIAACT model, much like the real world, actions do not take a fixed amount of time. Instead, each arrow also has a corresponding function which maps time to probability of completion at that point in time. There are four available actions: $transfer$, $decide$, $resolve$, $execute$ (as seen in Figure 2). $Transfer$ results in a shift of autonomy between a human and an agent. $Decide$ allows for a decision to be made and results in a consistent state with probability $P(c, A)$ for an agent decision or $P(c, H)$ for a human decision. Conversely an inconsistent state is reached with probability $1-P(c, A)$ by an agent and $1-P(c, H)$ by a human. $Resolve$ is an action that attempts to transition from an inconsistent state Adi or Hdi to a consistent state Adc or Hdc, which yields higher rewards. To $Execute$ a particular decision, results in the implementation of that decision towards the $finish$ state. The RIAACT model in Figure 2 provides the ability to interrupt the human $decision$ action and both $resolve$ actions. The $interrupt$ action is drawn with an arrow and also has an action duration distribution. For example, if the resolving of an inconsistency is taking too long, the agent may wish to $interrupt$ the $resolve$ action and return to the inconsistent human decision Hdi so that the $Finish$ state can be reached before the deadline.

Rewards - The reward R for a state is only received if that state is reached before the deadline. RIAACT reasons about the average reward for successfully completing (arriving at $Finish$) the $execute$ action from a decision state. In previous adjustable autonomy work, then the decision Ad or Hd would have been made by either party and assumed to have an average quality or reward $R(Ad)$ or $R(Hd)$. In order to model the discrepancies that the agents and humans can have, we extend the model to categorize the decision as either consistent Adc or Hdc or inconsistent Adi or Hdi and the reward function varies accordingly.

Policy - The policy prescribes the best team action for each state from now until the time of deadline td. The time of deadline for a particular role is the point at which that role becomes unachievable or irrelevant. Thus, reasoning about the role further would not provide any benefit. For every team decision, there is an adjustable autonomy policy between the human and the agent team.

The goal of this policy is to determine the critical point at which a decision should be delegated ($transfer$), when a decision should be made ($decide$), and when to undergo inconsistent decision resolution ($resolve$). In addition, this RIAACT model requires a new class of policies that give the optimal amount of time to continue an action before choosing to $interrupt$ (we will later refer to these as "transitory policies").

3.2 Inconsistency Resolution in RIAACT

Once an inconsistency has been detected, the agent team can attempt to $resolve$ it, but must consider the potential gain of reward, the time available until the deadline, and the duration of the $resolve$ action. RIAACT plans for the $resolve$ action to be attempted, but also that action can be interrupted at any point in continuous time. Consequently, the real benefit RIAACT is that it produces both a policy that can suggest a $resolve$ action, but an additional (transitory state) policy that determines the optimal time to interrupt that action. There are many different methods of interaction between a human and an agent that could be implemented in order to $resolve$. However, RIAACT focuses on the modeling of the action duration so that an overarching adjustable autonomy policy can be created, independent of the method of resolution.

Inconsistency Detection - When an inconsistency is detected by the agents in an uncertain environment, an important aspect is the probability of that inconsistency being raised leading to a different decision, which is defined as $P(IU)$. As $P(IU)$ approaches 0, it becomes less useful to attempt resolving a detected inconsistency. Also, as $P(IU)$ approaches 1, the benefits of a resolve action increases.

Inconsistent State - Inconsistency occurs as a result of the gap in information between the agent team and the human. As mentioned earlier, the likelihood that this inconsistency will occur is modeled in RIAACT by $1 - P(c, H)$ for the human and $1 - P(c, A)$ for an agent. Inconsistency is detected during execution by a teammate evaluating an input decision and comparing it to its own decision.

Resolve Action - The resolve action is a team action that transitions from an inconsistent state (Adi or Hdi) to a consistent one (Adc or Hdc). This is beneficial, if the expected utility of executing a consistent decision is sufficiently higher than that of executing an inconsistent one. In order to be able to react to the approaching deadline, the resolve action is interruptible.

Resolve Action Duration - RIAACT assumes that all inconsistencies can be resolved, yet it may take an uncertain amount of time to do so. The IA-TMDP model enables the $resolve$ action duration to follow a distribution. If an inconsistency is difficult to resolve, this is captured in the action duration distribution being very long.

3.3 Interruptible Action TMDPs

In order to realize the RIAACT adjustable autonomy model presented above, we will introduce our own framework for planning with interruptible actions. Traditional TMDPs can give precise timings for policies, however do not allow actions to be interrupted, i.e., once an action has been started, its execution must be continued (even if it is no longer desired) until the action terminates naturally. In contrast, the Interruptible Action TMDP model (IA-TMDP) that we now propose allows actions to be interrupted and changed at any point in time. Consequently, the policies of IA-TMDPSs show not only what action should be started given the current time, but also (for each interruptible action), how long each action should be executed before being interrupted. In order to explain how IA-TMDPs work, recall the definition of TMDPs in Section 2.2. IA-TMDP differs from TMDP in that it allows for actions to be both interruptible and non-interruptible. For non-interruptible actions, IA-TMDP behaves exactly as a TMDP. However, for interruptible actions, IA-TMDP behaves differently: When an interruptible action $a \in A$ is executed in state $s \in S$ at time t, IA-TMDP first immediately (at time t) transitions to a corresponding **transitory state** $\langle s, a, s' \rangle$, with probability $P(s, a, s')$. This transitory state can be thought of as being between two states while executing an interruptible action. The agent can then either **continue** or **interrupt** the execution of a, in a transitory state $\langle s, a, s' \rangle$:

- If the agent decides at time $t' > t$ to continue the execution of a, two outcomes are possible: (i) with probability $\int_0^{t'-t} d_{s,a,s'}(x)dx$ the execution of a will terminate naturally, i.e., the process will transition at time t' to state s' and the agent will receive reward $R(s, a, s', t')$ or (ii) with probability $1 - \int_0^{t'-t} d_{s,a,s'}(x)dx$ the execution of a will not terminate, i.e., the process will stay at the transitory state $\langle s, a, s' \rangle$ but the current time t' will increase to $t' + \epsilon$ for an infinitesimally small $\epsilon > 0$. At this point, the agent can again decide whether to continue or interrupt the execution of a.
- If the agent decides at time $t' > t$ to interrupt the execution of a, the process returns to the starting state s at time $t' + \delta$ where δ is the delay in time in interrupting action a sampled from a given distribution $\delta_{s,a,s'}$. (The penalty for interrupting the execution of a is to irreversibly lose $t' + \delta - t$ units of time.)

The result of planning with interruptible actions is that IA-TMDP policies are more expressive than TMDP policies. Specifically, for each pair $s \in S; t \in [0, \Delta]$ an IA-TMDP policy not only specifies an action a to be executed from state s at time t, but also, provides a time after which the execution of a is to be interrupted, which we will refer to as a "transitory policy." We show in Figure 4 that IA-TMDP yield improved policies which result in significantly better results when applied.

3.4 Efficient Implementation of RIAACT

In order to use most existing techniques to allow for plan for interrupting an action at any point in continuous time would require an enormous amount of decision states. Each of these states would determine whether to continue the current action or to interrupt it and return to the originating state. However, we were able to leverage recent advancements in TMDP solution techniques in order efficiently solve IA-TMDPs. We extended the CPH solver [6], which is currently the fastest TMDP solver. The CPH solver plans for an infinite amount of decision points with only a discrete amount of additional intermediate states. This is due to the fact that transition durations are not deterministic, but rather follow a exponential distribution in their duration. Consequently, this allows for one state transition to model many different instances in time, with the opportunity to interrupt.

CPH approximates each action duration distribution with a phase type distribution, which is a directed graph whose transitions are sampled from an exponential distribution (for example $e^{-\lambda_1 t}$, $e^{-\lambda_2 t}$, $e^{-\lambda_3 t}$ ...). In order to increase the accuracy of the approximation of a non-exponential distribution, the process adds extra intermediate states, each with exponential transitions. This is the case with a common approximation technique known as a Coxian [13]. The Coxian is a standard phase-type approximation that only allows transitions from the current state to either the next intermediate state, the final state, or a self transition. This allows for an accurate approximation without having exponential numbers of transitions as the number of phases increases. After each action distribution is approximated as one or more exponential distributions,

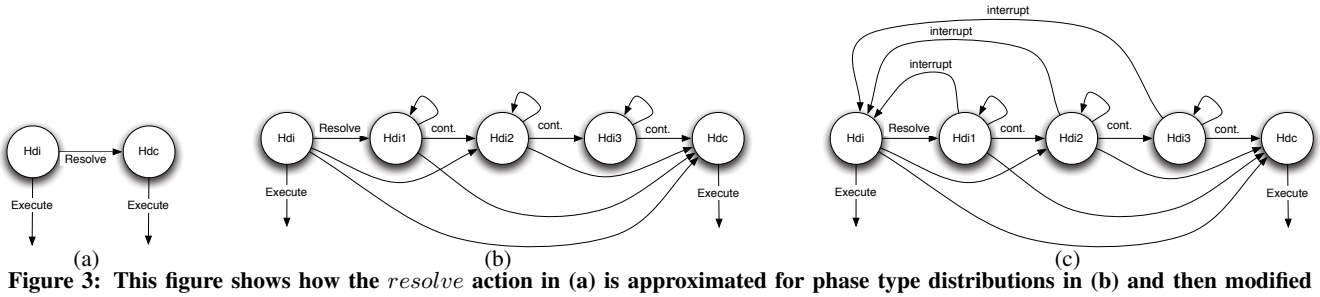


Figure 3: This figure shows how the $resolve$ action in (a) is approximated for phase type distributions in (b) and then modified further for interruption in (c).

the result is a new model where every transition is an exponential distribution, albeit with possibly more states. Next, a uniformization process is applied [8] such that all exponential durations are now the same. Figure 3a shows a subset of Figure 2 in which the $resolve$ action has an arbitrary non-exponential duration distribution for transitioning from state Hdi to state Hdc. Figure 3b shows the creation of extra states and transitions that result after approximations of phase type distributions and uniformization, which converts all the exponentials to have the same λ, but introduces self-transitions [8]. The transition $cont.$ in Figure 3b represents the continuing of the original action, $resolve$. Later, the state transitions are extended to allow for the interrupting of actions (as seen in Figure 3c). Figure 3c shows only how one action can be altered, however the entire RIAACT model (Figure 2) must have all non-exponential distributions approximated as a sequence of phase-type exponential functions for CPH. The interruption results in the originating state, Hdi in this example, after some interruption duration distribution, denoted by the interrupt arrows in Figure 3c.

By using the model shown in Figure 3c, we are able to use a CPH solver that has been augmented to allow for an extra action in each intermediate states ($Hdi1$, $Hdi2$, $Hdi3$) and compute a new expected utility over these states. Previously, these extra states existed only for planning about action durations and did not yield a policy, since they don't represent states in the original problem. However, being in an intermediate state represents the continuing of an action or more specifically, in a transitory state $\langle s, a, s' \rangle$ according to the IA-TMDP model shown in Section 3.3).

In order to calculate a policy for attempting an interruptible action, we must also reason about the likelihood of being in each of the intermediate states ($Hdi1$, $Hdi2$, or $Hdi3$ in Figure 3c). This is because the intermediate states do not represent actual states, but exist for approximation purposes. We can leverage their policies however in order to aggregate them into the transitory state policies. We ran a monte-carlo simulation in order to quickly obtain the belief distribution over the intermediate states. A histogram can be created by running the simulated action ($resolve$) from start (Hdi) to finish (Hdc) a number of times and recording for each run at what time each intermediate state is reached. Then a belief distribution is created by looking at a certain amount of elapsed since the action as started (this is referred to as the transitory state in an IA-TMDP) and evaluating how often on average the agent is in a particular intermediate state. This does not factor in, the probability that the action has completed, because once the next state has been reached, there is a policy for that state. The action (continue or interrupt) expected values of the intermediate phases are weighted with the belief distribution of being in those particular intermediate phases after a certain amount of elapsed time. This creates a policy that maps the transitory state over time to a weighted expected utility of interrupt versus continue.

4. EXPERIMENTS

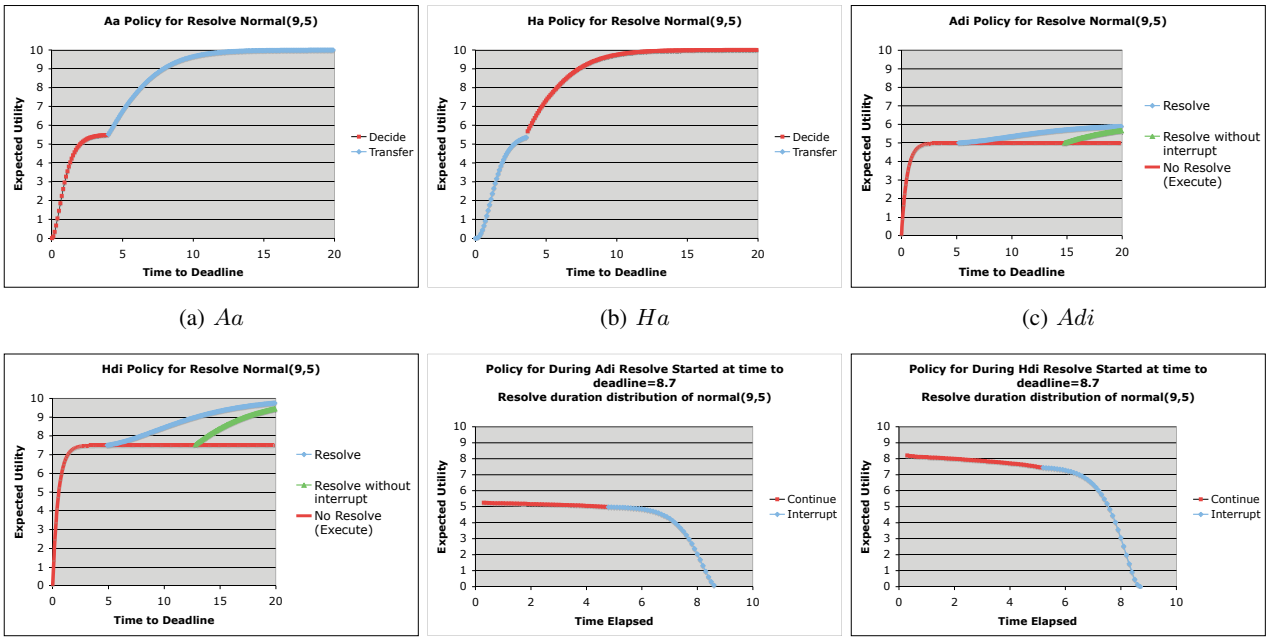
We have conducted two sets of experiments to investigate RIAACT. The first set are policy experiments used to explore the advantages of RIAACT over previous adjustable autonomy models. The second set applies these policies to a disaster simulation system, DEFACTO. Both sets of experiments use the same motivating disaster response scenario.

4.1 Scenario and Experimental Setup

Here, we will instantiate a specific disaster response scenario where a human incident commander is collaborating with a team of fire engine agents. The scenario includes 6 fire engines that must address a large scale disaster in which 2 high-rise buildings in an urban area have caught on fire. These fires are engulfing the buildings quickly and each have the chance of spreading to adjacent buildings. A decision must be made quickly on how the team is to divide their limited resources (fire engines) among the fires. Each allocation decision can be made by either a human or an agent. Agents may determine that the decision made by the human is inconsistent with their desired allocations. These fire engines have the goal of resolving inconsistency, however, the resolution may take a long time and the team is acting in a disaster response situation where time is short. There is a deadline that occurs when the fire spread to adjacent buildings. Consequently, the action of resolving is interruptible.

We use the RIAACT model as explained in Section 3 and now will instantiate the values for this particular scenario. We assume that this situation is uncertain and the probability of consistency is 0.5 for both the human $P(c, H)$ and the agent $P(c, A)$. As in Section 2.2, we will express the reward as $R\langle s, a, s', t \rangle$. In our disaster response domain, we measure the reward in terms of buildings saved compared to the maximum that would catch fire if not addressed (10). We assume that the reward for an agent decision is less than that of a human. Thus, the reward of an agent decision that is consistent $R\langle Adc, Execute, Finish \rangle = 6$, whereas an agent's inconsistent reward $R\langle Adi, Execute, Finish \rangle = 5$. The reward of a consistent human decision $R\langle Hdc, Execute, Finish \rangle = 10$ is assumed to be the maximum, whereas an inconsistent human decision reward gives $R\langle Hdi, Execute, Finish \rangle = 7.5$. Note that these rewards are averages over the multiple potential decisions that would be made of each category: Adc, Adi, Hdc, Hdi. The reason that $R\langle Hdc, Execute, Finish \rangle$ does not necessarily equal $R\langle Adc, Execute, Finish \rangle$ is that the distributions of consistent decisions are different depending on the decision maker. Because the complexity of this domain and distribution of information across humans and fire engine agents, we first assume the probability that raising the inconsistency is useful $P(IU) = 0.5$. However, we vary this value later in Section 4.3.

For application to the DEFACTO simulation, we treat the RIAACT IA-TMDP policy as a team plan, composed of joint actions [3]. Upon generation of the policy, an agent communicates that policy to the rest of the team, which allows us to leverage existing



(a) Aa (b) Ha (c) Adi

(d) Hdi (e) $\langle Adi, resolve, Adc \rangle$ (f) $\langle Hdi, resolve, Hdc \rangle$

Figure 4: RIAACT model IA-TMDP policy output given that the resolve action duration fits a Normal(9,5). Subfigures a-d are policies for corresponding states. Subfigures e and f show the IA-TMDP interrupt policies for two transitory states.

team coordination algorithms such as those based on [3, 10]. An added benefit of this approach is that multiple agents will not simultaneously commit to resolve, thereby preventing conflicting or redundant resolve team plans. This hybrid approach avoids using computationally expensive distributed MDPs for coordination [7].

For these experiments, we categorize actions as either involving a human or being a machine-only action. It is customary to model the action duration involving a human as having a normal distribution and machine-only actions are modeled as having an exponential distribution [13]. Consequently, we assume the machine-only action duration to be a short duration exponential function with a mean of 0.5 seconds. This applies to the $transfer$ of autonomy action, the $decide$ action for an agent, the $execute$ decision action, and the interrupt action in our domain. The $decide$ action for human (agent wait for human decision) and the $resolve$ action for both agent and human will be modeled as a normal distributions. We will assume the human $decide$ action duration at a distribution of Normal(3,1)[1]. We model each normal distribution action duration with a 3 phase Coxian approximation (as seen in Figure 3c). Although we assign normal and exponential durations for this scenario, the RIAACT IA-TMDP can model the action duration to have any arbitrary distribution. These experiments will focus on the $resolve$ action, which allows us to demonstrate the distinct benefits of RIAACT: resolving inconsistencies, developing a continuous time policy, and allowing interruptible actions.

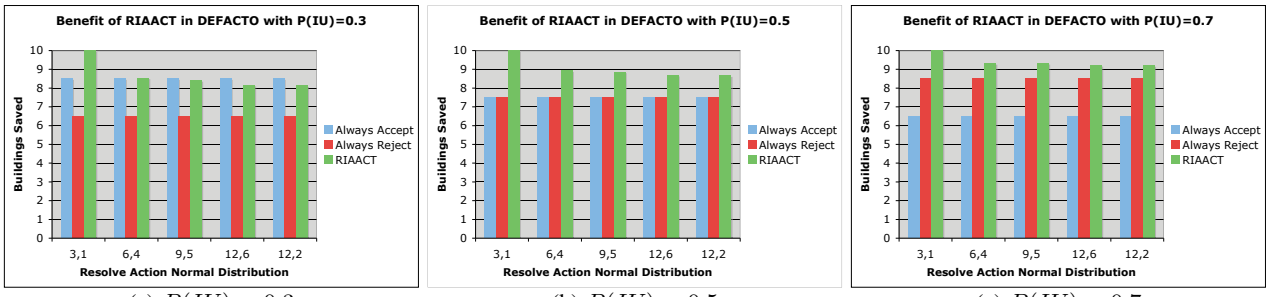
4.2 Testbed Policy Experiments

For these first experiments, we created a testbed domain to construct a policy that included 6 agents, where the $resolve$ action duration follows a Normal(9,5). These experiments show the RIAACT model benefits of (i) continuous time, (ii) the resolve action, and (iii) interrupting of an action. The result of the experiment was that each of the benefits are shown and this confirms the usefulness of the RIAACT model in the testbed environment.

Figure 4 shows an example of a policy where the $resolve$ action duration distribution is a Normal(9,5). The policies for states Adc and Hdc have been omitted from the figure since they show only one action over time to be taken from these consistent decisions, $execute$. For each state, the policy shows the optimal action to take and the expected utility of that action as a function over time. Figure 4c and 4d include additional policies, but the optimal policy is following the action with the highest utility over time. On each x-axis is the amount of time left until the deadline and on the y-axis is the expected utility. Thus, if any state is reached, given the time to deadline, the optimal action is chosen. For example, if the human has the autonomy (Ha) and the time to deadline is greater than 3.6 seconds, then the optimal action is to attempt a human decision. Otherwise, the optimal action is to transfer that decision over to the agent in order to have the agent make a quicker, but lower average quality decision. Figure 4a shows that the dominant action for the agent has autonomy state, Aa, is to transfer the decision to the human up until 3.9 seconds before the deadline. On the other hand, Figure 4b shows that the dominant action for the human has autonomy state Ha is to $decide$ up until 3.6 seconds before the deadline.

Figure 4c and 4d show the times at which the $resolve$ action is optimal. In order to show the benefit that the $resolve$ action provides and the benefit that $resolve$ receives from being interruptible, a new set of experiments was run. The results of this experiment can be seen in Figure 4c and 4d. The No Resolve line represents the policy from previous work, where the inconsistent decision is executed immediately. The Resolve without interrupt line represents where the policy deviates from $execute$ if the $resolve$ action was not interruptible. Lastly, the Resolve line represents the benefits of having both the resolve action available and having it be interruptible. As seen in both charts, the standard $resolve$ action (with interrupt) provides a higher expected reward at times. For example, as seen in Figure 4c the policy for Adi is to attempt to resolve an inconsistency if it is detected with at least 14.8 seconds if the

[1]We will represent normal distributions as Normal(Mean,Standard Deviation).



(a) $P(IU) = 0.3$ (b) $P(IU) = 0.5$ (c) $P(IU) = 0.7$

Figure 5: Experiments given a simulated human and varying the probability that resolving the inconsistency would be useful $P(IU)$.

$resolve$ is not interruptible, however the resolve action is beneficial if the inconsistency is detected with at least 5.2 seconds if the $resolve$ is interruptible.

Figure 4e and 4f shows two transitory state ($\langle s, a, s' \rangle$) policies that are created in addition to the traditional state policies, as a result of using an IA-TMDP. If the action were not interruptible, then the action inherently chosen over all execution time would be the continue action. These policies are for a resolve action if it were to be started at 8.7 seconds to deadline from either the Adi or Hdi. To express this in terms of the IA-TMDP model, these are interruption action policies for the transitory states $\langle Adi, resolve, Adc \rangle$ and $\langle Hdi, resolve, Hdc \rangle$ respectively (for more on transitory states, see Section 3.3). The x-axis here shows time elapsed since the action has started and the y-axis shows the expected utility of the optimal action (continue or interrupt in this case). According to the policy that can be derived from this graph, $\langle Adi, resolve \rangle$ should be continued for 4.7 seconds, whereas $\langle Hdi, resolve \rangle$ should be continued for 5.2 seconds before it should be interrupted. This longer time before interrupting the $resolve$ action is due to the greater potential benefit from resolving an inconsistent human decision (7.5 to 10 reward) versus an inconsistent agent decision (5.0 to 6.0 reward).

4.3 DEFACTO Experiments

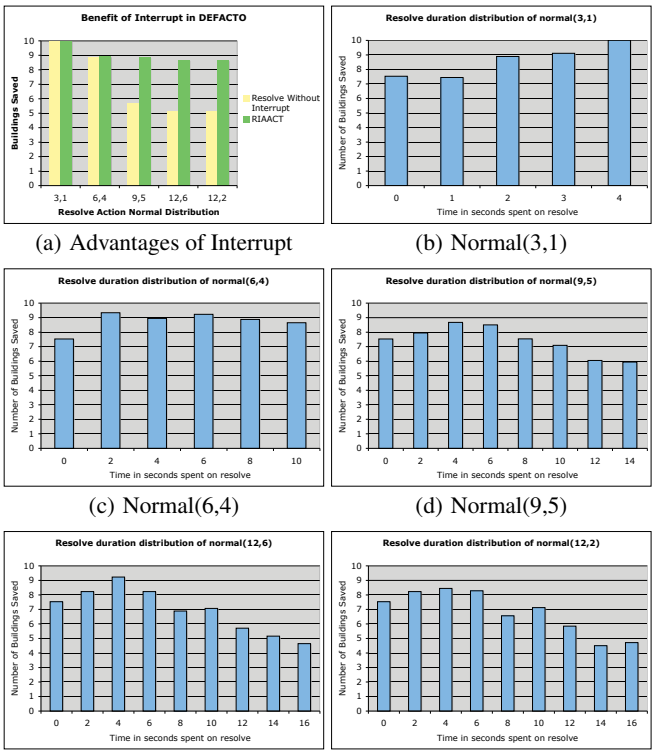
We have also implemented RIAACT in a disaster response simulation system (DEFACTO), which is a complex system that includes several simulators and allows for humans and agents to interact together in real-time [10]. These experiments have been conducted in the DEFACTO simulation in order to test the benefits of the RIAACT policy output. In the scenario that we are using for these experiments, the human had the autonomy and has made a decision. However, this decision is found to be inconsistent (Hdi) and now a RIAACT IA-TMDP policy is computed (following the scenario in Section 4.1) to determine two things: 1. whether, at this time, a $resolve$ action is beneficial and 2. if so, for how long should the $resolve$ action be continued before it is beneficial to interrupt it and $execute$ the inconsistent human decision Hdi.

These experiments were set up to show the application of the IA-TMDP policies to a human-multiagent disaster response team. The experiments included 6 agents and a simulated human. Section 4.2 explained the RIAACT policy space for an experimental setting where the $resolve$ duration was kept as Normal(9,5). In these experiments, we create a new RIAACT policy for each of the following $resolve$ duration distributions: Normal(3,1), Normal(6,4), Normal(9,5), Normal(12,6), and Normal(12,2). This serves to explore the effects of modeling varying resolve durations and how they affect the policy and eventually the team performance. In each of the experiments, the deadline is the point in time at which fires spread to adjacent buildings and becomes uncontrollable, which in the simulation is 8.7 seconds until deadline.

Using the RIAACT policies, we conducted experiments where DEFACTO was run with simulated human input. Simulated human data was used to allow for repeated experiments and to achieve statistical significance in the results. To show the benefit that the $resolve$ action gets from being interruptible, experiments were conducted comparing the performance of the $resolve$ action following the RIAACT policy, Always Accept policy or the Always Reject policy (see Figure 5). Each of the subfigures varies the probability that the detected inconsistency was useful, $P(IU)$. The duration is sampled from the varying normal distributions, shown on the x-axis. These are averaged over 50 experimental runs. The y-axis shows performance in terms of amount of buildings saved. The Always Accept policy is the equivalent of previous work in adjustable autonomy where a decision was assumed to be final and implemented. Alternatively, the Always Reject policy will ignore a decision if an inconsistency is detected. The RIAACT policy improves over both of these static policies.

Figure 5b also shows that as the $resolve$ action duration increases, the benefit gained from using RIAACT decreases. This is due to the approaching deadline and the decreased likelihood that the $resolve$ will be completed in time. Although, the difference in performance for the Normal(12,2) case may be the smallest, the results show statistical significance $P < 0.05$ ($P = 0.0163$). Figure 5b shows how the benefits that RIAACT brings are affected by the probability that the inconsistency that was detected is useful $P(IU)$. At first (subfigure b), this was assumed to be 0.5 since the domain was so uncertain. However, if the probability that the detected inconsistency, if resolved, leads to a better solution increased to 0.7, as in Figure 5c, the performance of RIAACT increases. This is due to the fact that the attempted $resolve$ action is more likely to result in higher performance. However, if $P(IU) = 0.3$, as in Figure 5a, then the benefits of RIAACT decrease with respect to Always Accept (previous strategy) until in the Normal(9,5) case, RIAACT can be seen to do worse. These experiments highlight how the resolve action, and RIAACT in general, provides performance improvement as long as the estimate of $P(IU)$ is accurate and relatively high. Otherwise, if the agent team is poor at detecting inconsistencies that will result in better team performance, then it might be better to not resolve these inconsistencies.

As can be seen from Figure 6a, the benefit in team performance is not only from having the $resolve$ action itself, but by allowing it to be interruptible. Again, the x-axis shows the various distributions tested, and the y-axis shows team performance in terms of number of buildings saved. The gains are not as present in Normal(3,1) because the action duration is so short that it need not be interrupted. However, as seen in Normal(9,5), an average of 3.2



(a) Advantages of Interrupt (b) Normal(3,1)

(c) Normal(6,4) (d) Normal(9,5)

(e) Normal(12,6) (f) Normal(12,2)

Figure 6: Experiments showing the advantage of interrupt (a) and continuous time (b-f) in RIAACT.

more buildings were saved over an uninterruptible action.

In order to explore the benefit of having these policies expressed in continuous time, we conducted experiments that varied the amount of time that the $resolve$ would wait before being interrupted. Figure 6 subfigures b-f show the results from implementing the RIAACT policy in DEFACTO and running experiments with a simulated human that has a response time sampled from varying distributions. They have been averaged over 50 runs. These distributions are listed in the format of Normal(mean, standard deviation) in minutes. The y-axis shows the average amount of buildings that were saved. On the x-axis is a varying amount of time that was spent on the resolve action before interrupting it and executing the inconsistent decision. For example, in the Normal(12,6) graph, resolving for 6 seconds, the expected amount of buildings saved drops to 8.2 versus the 8.7 buildings saved on average by rather than the 4.8 seconds determined by the RIAACT policy seen in Figure 5. This shows the benefit of being able to give a policy in continuous rather than a policy with time intervals that would only be able to interrupt at 6 seconds.

5. CONCLUSION

In this paper, we have presented an approach to address the challenges that arise in adjustable autonomy problems for human-multiagent teams acting in uncertain, time-critical domains, called RIAACT. This research is motivated by challenges that arose in a realistic application, where human-multiagent teams must interact and coordinate [10]. Our approach makes three contributions to the field in order to address these challenges. First, our adjustable autonomy framework models resolution of inconsistencies between human and agent view, rather than assuming that decisions must be final. Second, agents plan their interactions in continuous time, avoiding a discretized time model, while remaining efficient. Third, we plan for the ability of an adjustable autonomy team action to be interrupted at any point in continuous time.

In addition, we have introduced a new model, Interruptible Action - Time dependant Markov Decision Problem (IA-TMDP), which allows for planning with actions that are interruptible at any point in continuous time. This results in new "transitory state" policies that determine the optimal time to interrupt an action. We have conducted experiments that both explore the RIAACT policy space and apply these policies to an urban disaster response simulation. These experiments have shown how RIAACT can provide improved policies that increase human-multiagent team performance.

In the future, a worthy topic of interest would be apply RIAACT to other time critical domains and increase the fidelity of RIAACT's state space. Also, IA-TMDPs have the potential to be applied to many other time critical planning problems beyond adjustable autonomy.

6. ACKNOWLEDGMENTS

This research was supported by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under grant number N00014-05-0630.

7. REFERENCES

[1] J. Boyan and M. Littman. Exact solutions to time-dependent MDPs. In *NIPS*, pages 1026–1032, 2000.

[2] CALO http://www.ai.sri.com/project/CALO, http://calo.sri.com. *CALO: Cognitive Agent that Learns and Organizes*, 2003.

[3] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2–3):213–261, 1990.

[4] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *SIGART Conference on Human-Robot Interaction, HRI*, 2007.

[5] L. Li and M. Littman. Lazy approximation for solving continuous finite-horizon MDPs. In *AAAI*, pages 1175–1180, 2005.

[6] J. Marecki, S. Koenig, and M. Tambe. A fast analytical algorithm for solving markov decision processes with real-valued resources. In *IJCAI*, January 2007.

[7] R. Nair and M. Tambe. Hybrid bdi-pomdp framework for multiagent teaming. *Journal of Artificial Intelligence Research (JAIR)*, 23:367–420, 2005.

[8] M. Puterman. *Markov decision processes*. John Wiley and Sons, New York, 1994.

[9] P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.

[10] N. Schurr, P. Patil, F. Pighin, and M. Tambe. Using multiagent teams to improve the training of incident commanders. In *AAMAS '06*, NY, USA, 2006. ACM.

[11] B. P. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE - Special Issue on Multi-Robot Systems*, July 2006.

[12] P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical pomdps for personal assistant agents. In *AAMAS*, 2005.

[13] H. Younes and R. Simmons. Solving generalized semi-MDPs using continuous phase-type distributions. In *AAAI*, 2004.