Planning with Stochastic Resource Profiles: An Application to Human-Robot Co-habitation

Tathagata Chakraborti[1] and **Yu Zhang**[1] and **David Smith**[2] and **Subbarao Kambhampati**[1]

Department of Computer Science[1]
Arizona State University
Tempe, AZ 85281, USA
{tchakra2,yzhan442,rao}@asu.edu

Intelligent Systems Division[2]
NASA Ames Research Center
Moffett Field, CA 94035-1000, USA
david.smith@nasa.gov

Abstract

It is important for robotic agents to be respectful of the intentions of the human members cohabiting an environment and account for conflicts on the shared resources in the environment, in order to be acceptable members of human-robot ecosystems. In this paper we look at how maintaining predictive models of the human cohabitors in the environment can be used to inform the planning process of the robotic agents. We introduce an Integer Programming based planner as a general formulation of the "human-aware" planning problem and show how the proposed formulation can be used to model different behaviors of the robotic agent, showcasing compromise, opportunism or negotiation. Finally, we show how the proposed approach scales with the different parameters involved, and provide empirical evaluations to illustrate the pros and cons associated with the proposed style of planning.

In environments where multiple agents are working independently, but utilizing shared resources, it is important for these agents to maintain belief models of other agents so as to act intelligently and prevent conflicts. In cases where some of these agents are humans, as in assistive robots in household environments, these are required (rather than desired) capabilities of robots in order to be "socially acceptable" - this has been studied extensively under the umbrella of "human-aware" planning, both in the context of path planning (Sisbot et al. 2007; Kuderer et al. 2012) and in task planning (Cirillo, Karlsson, and Saffiotti 2009; Koeckemann, Pecora, and Karlsson 2014; Cavallo et al. 2014; Tomic, Pecora, and Saffiotti 2014). Probabilistic plan recognition can play an important role in this regard, because by not committing to a plan, that pre-assumes a particular plan for the other agent, it might be possible to minimize suboptimal (in terms of redundant or conflicting actions performed during the execution phase) behavior of the autonomous agent. Here we look at possible ways to minimize such suboptimal behavior by ways of compromise, opportunism or negotiation. There has been previous work (Beaudry, Kabanza, and Michaud 2010; Cirillo, Karlsson, and Saffiotti 2010) on some of the modeling aspects of the

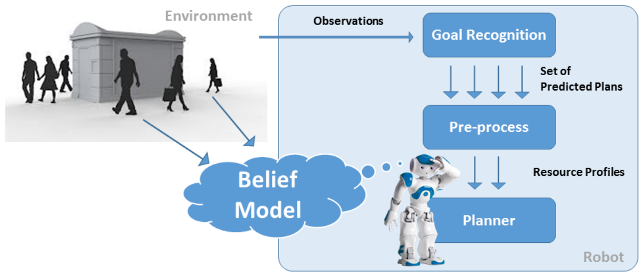


Figure 1: Architecture diagram - the robot has partial beliefs of the world, which it uses to predict and plan.

problem, in terms of planning with uncertainty in resources and constraints. In this paper we provide a unified framework of achieving these behaviors of the autonomous agents, particularly in such scenarios of human robot cohabitation.

The general framework of the problem addressed in this work is shown in Figure 1. The autonomous agent, or the robot, is acting (with independent goals) in an environment co-habited with other agents (humans), who are similarly self-interested. The robot has a model of the other agents acting independently in its environment. These models may be partial and hence the robot can only make uncertain predictions on how the world will evolve with time. However, the resources in the environment are limited and are likely to be constrained by the plans of the other agents. The robot thus needs to reason about the future states of the environment in order to make sure that its own plans do not produce conflicting states with respect to the plans of the other agents. With the involvement of humans, however, the problem is more skewed against the robot, because humans would expect a higher priority on their plans - robots that produce plans that clash with those of the humans, without any explanation, would be considered incompatible for such an ecosystem. Thus the robot will be expected to follow plans that preserve the human plans, rather than follow a globally optimal plan for itself. This aspect makes the current setting distinct from normal human robot teaming scenarios and produces a num-

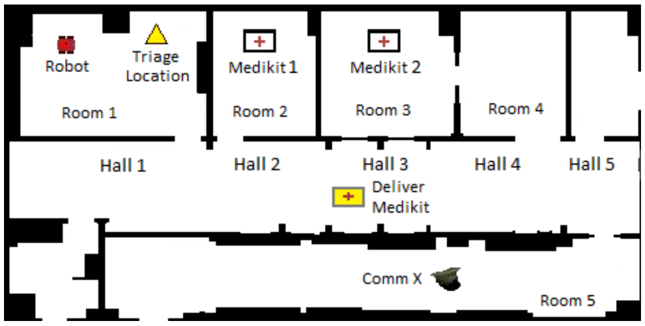


Figure 2: The running example - a human commander and a robot involved in an USAR setting, with constrained resources (medkits).

ber of its own interesting challenges. How does the robot model the humans' behavior? How does it plan to avoid friction with the human plans? If it is possible to communicate, how does it plan to negotiate and refine plans? These are the questions that we seek to address in this work. Our approach models human beliefs and defines resource profiles as abstract representations of the plans predicted on the basis of these beliefs of the human agents. The robot updates its own beliefs about the world upon receiving every new observation from its environment, and passes on the resultant profiles onto its own planner as shown in Figure 1. For the planning module, we introduce an IP-based planner that minimizes the overlap between these resource profiles and those produced by the robot's own plan in order to maintain least conflicts with the predicted human tasks in the future.

1 Planning with Resource Profiles

We will now go into details about each of the modules shown in Figure 1. We will be using a similar setting as the one described in (Talamadupula et al. 2014) (shown in Figure 2) as the running example throughout this discussion. The setting involves a commander `CommX` and a robot in a USAR (Urban Search and Rescue) scenario. The shared resources here are the two medkits - some of the plans the commander can execute will lock the use of and/or change the position of these medkits, so that from the set of probable plans of the commander we can extract a probability distribution over the usage (or even the position) of the medkit over time based on the fraction of plans that conform to these facts. These *resource availability profiles* provide a way for the agents to minimize conflicts with the other agents. Before going into details about the planner that achieves this, we will first look at how the agents are modeled and how these profiles are computed in the next section.

1.1 The Belief Modeling Component

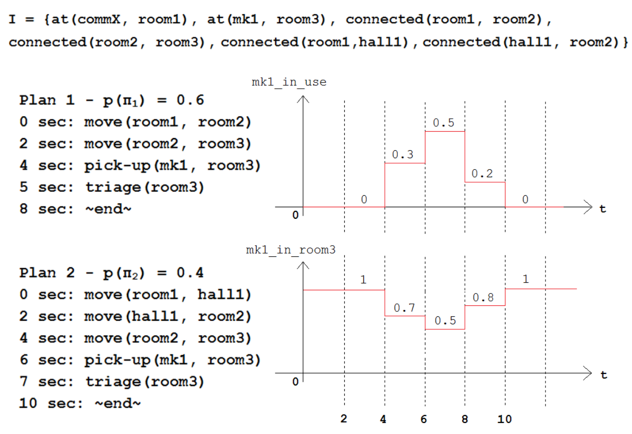
The notion of modeling beliefs introduced by the authors in (Talamadupula et al. 2014) is adopted in this work and described briefly here. Beliefs about state are defined in terms of predicates $bel(\alpha, \phi)$, where α is an agent with belief $\phi = true$. Goals are defined by predicates $goal(\alpha, \phi)$, where agent α has a goal ϕ. The set of all beliefs that the robot ascribes to α together represents the perspective for the robot of α. This is obtained by a belief model Bel_α of agent α, defined as $\{ \phi \mid bel(\alpha, \phi) \in Bel_{self} \}$, where Bel_{self} are the first-order beliefs of the robot (e.g., $bel(self, at(self, room1))$). The set of goals ascribed to α is similarly described by $\{goal(\alpha, \phi) | goal(\alpha, \phi) \in Bel_{self}\}$.

Next, we turn our attention to the domain model D_α of the agent α that is used in the planning process. Formally, a planning problem $\Pi = \langle D_\alpha, \pi_\alpha \rangle$ consists of the domain model D_α and the problem instance π_α. The domain model of α is defined as $D_\alpha = \langle T_\alpha, V_\alpha, S_\alpha, A_\alpha \rangle$, where T_α is a set of object types; V_α is a set of variables that describe objects that belong to T_α; S_α is a set of named first-order logical predicates over the variables V_α that describe the state; and A_α is a set of operators available to the agent. The action models $a \in A_\alpha$ are represented as $a = \langle \mathbb{N}, \mathbb{C}, \mathbb{P}, \mathbb{E} \rangle$ where $\mathbb{N}$ denotes the name of that action; $\mathbb{C}$ is the cost of that action; $\mathbb{P}$ is the list of pre-conditions that must hold for the action a to be applicable; and $\mathbb{E} = \{eff^+(a), eff^-(a)\}$ is a list of predicates in S_α that indicates the effects of applying the action. The transition function $\delta(\cdot)$ determines the next state after the application of action a in state s as $\delta(a, s) = (s \setminus eff^-(a)) \cup eff^+(a), s \subseteq S_R$. For this work, we assume that the action models available to an agent are completely known to all the other agents in the scenario; that is, we rule out the possibility of beliefs on the models of other agents.

The belief model, in conjunction with beliefs about the goals / intentions of another agent, will allow the robot to instantiate a planning problem $\pi_\alpha = \langle \mathbb{O}_\alpha, \mathbb{I}_\alpha, \mathbb{G}_\alpha \rangle$, where $\mathbb{O}_\alpha$ is a set of objects of type $t \in T_\alpha$; $\mathbb{I}_\alpha$ is the *initial state* of the world, and $\mathbb{G}_\alpha$ is a set of *goals*, which are both sets of the predicates from S_α initialized with objects from $\mathbb{O}_\alpha$. First, the initial state $\mathbb{I}_\alpha$ is populated by all of the robot's initial beliefs about the agent α, i.e. $\mathbb{I}_\alpha = \{\phi \mid bel(\alpha, \phi) \in Bel_{robot}\}$. Similarly, the goal is set to $\mathbb{G}_\alpha = \{\phi \mid goal(\alpha, \phi) \in Bel_{robot}\}$. Finally, the set of objects $\mathbb{O}_\alpha$ consists of all the objects that are mentioned in either the initial state, or the goal description: $\mathbb{O}_\alpha = \{o \mid o \in (\phi \mid \phi \in (\mathbb{I}_\alpha \cup \mathbb{G}_\alpha))\}$. This planning problem instance (though not directly used in the robot's planning process) enables the goal recognition component to solve the compiled problem instances.

1.2 The Goal Recognition Component

It is unlikely for the robot to be aware of the goals of other humans in its environment completely, but it can be proactive in updating its beliefs incrementally based on observations of what the other agents are doing. To accommodate this, the robot's current belief of α's goal, $\mathbb{G}_\alpha$, is extended to a *hypothesis goal set* Ψ_α. The computation of this goal set can be done using planning graph (Blum and Furst 1995) methods. In the worst case, Ψ_α corresponds to all possible goals in the final level of the converged planning graph. Having further (domain-dependent) knowledge (e.g. in our scenario, information that `CommX` is only interested in triage-related goals) can prune some of these goals by removing the goal conditions that are not typed on the triage variable. At this point we refer to the work of Ramirez and Geffner who



```
I = {at(commX, room1), at(mk1, room3), connected(room1, room2),
connected(room2, room3), connected(room1,hall1), connected(hall1, room2)}

Plan 1 - p(π1) = 0.6
0 sec: move(room1, room2)
2 sec: move(room2, room3)
4 sec: pick-up(mk1, room3)
5 sec: triage(room3)
8 sec: ~end~

Plan 2 - p(π2) = 0.4
0 sec: move(room1, hall1)
2 sec: move(hall1, room2)
4 sec: move(room2, room3)
6 sec: pick-up(mk1, room3)
7 sec: triage(room3)
10 sec: ~end~
```

Figure 3: Different types of profiles corresponding to the two recognized plans.

in (Ramrez and Geffner 2010) provided a technique to compile the problem of goal recognition into a classical planning problem. Given a sequence of observations θ, the probability distribution Θ over $G \in \Psi_\alpha$ is recomputed by using a Bayesian update $P(G|\theta) \propto P(\theta|G)$, where the prior is approximated by the function $P(\theta|G) = 1/(1+e^{-\beta\Delta(G,\theta)})$ where $\Delta(G,\theta) = C_p(G-\theta) - C_p(G+\theta)$. Thus, solving two planning problems, with goals $G-\theta$ and $G+\theta$, gives the posterior distribution Θ over possible goals of α. We then compute the optimal plans for the goals in Ψ_α, which are used to compute the resource profiles described in the next section. Note here that one immediate advantage of using this specific goal recognition approach is that while computing the plan to a particular goal G we can reuse the compiled problem instance with the goal $G+\theta$ to ensure that the predicted plan conforms to the existing observations.

1.3 Resources and Resource Profiles

As we discussed previously, since the plans of the agents are in parallel execution, the uncertainty introduced by the commander's actions cannot be mapped directly between the commander's final state and the robot's initial state. However, given the commander's possible plans recognized by the robot, we can extract information about at what steps, or at what points of time, the shared resources in the environment are likely to be locked by the commander (given that we know what these shared resources are). This information can be represented by *resource usage profiles* that capture the expected (over all the recognized plans) variation of probability of usage or availability over time. The robot can, in turn, use this information to make sure that the profile imposed by its own plan has minimal conflicts with those of the commander's.

Formally, a profile is defined as a mapping from time step T to a real number between 0 and 1, and is represented by a set of tuples as follows $\mathcal{G} : \mathbb{N} \rightarrow [0,1] \equiv \{(t,g) : t \in \mathbb{N}, g \in [0,1],$ such that $G(t) = g$ at time step $t\}$.

The idea of the resource profiles can be handled at two levels of abstraction. Going back to our running example, shared resources that can come under conflict are the two (locatable typed objects) medkits, and the profiles over the medkits can be over both usage and location, as shown in Figure 3. These different types of profiles can be used (possibly in conjunction if needed) for different purposes. For example, just the usage profile shown on top is more helpful in identifying when to use the specific resource, while the resource when bound with the location specific groundings, as shown at the bottom can lead to more complicated higher order reasoning (e.g. the robot can decide to wait for the commander's plans to be over, as he inadvertently brings the medkit closer to it it with high probability as a result of his own plans). We will look at this again in Section 2.

Let the domain model of the robot be $D_R = \langle T_R, V_R, S_R, A_R \rangle$ with the action models $a = \langle \mathbb{N}, \mathbb{C}, \mathbb{P}, \mathbb{E} \rangle$ defined in the same way as described in Section 1.1. Also, let $\Lambda \subseteq V_R$ be the set of shared resources and for each $\lambda \in \Lambda$ we have a set of predicates $f^\lambda \subseteq S_R$ that are influenced by λ, and let $\Gamma : \Lambda \rightarrow \xi$ be a function that maps the resource variables to the set of predicates $\xi = \cup_\lambda f^\lambda$ they influence. Without any external knowledge of the environment, we can set $\Lambda = V_\alpha \cap V_R$ and $\xi = S_\alpha \cap S_R$, though in most cases these sets are much smaller. In the following discussion, we will look at how the knowledge from the hypothesis goal set can be modeled in terms of resource availability graphs for each of the constrained resources $\lambda \in \Lambda$.

Consider the set of plans Ψ_α^P containing optimal plans corresponding to each goal in the hypothesis goal set, i.e. $\Psi_\alpha^P = \{\pi_G = \langle a_1, a_2, \dots a_t \rangle \mid G = \delta(a_t, \dots \delta(a_2, \delta(a_1, \mathbb{I}_\alpha))) \; \forall \; G \in \Psi_\alpha$ and $a_i \in A_\alpha \forall i\}$ and let $l(\pi)$ be the likelihood of the plan π modeled on the goal likelihood distribution $\forall \; G \in \Psi_\alpha, \; p(G) \sim \Theta$ as $l(\pi_G) = c|\pi_G| \times p(G)$, where c is a normalization constant.

At each time step t, a plan $\pi \in \Psi_\alpha^P$ may lock one or more of the resources λ. Each plan thus provides a profile of usage of a resource with respect to the time step t as $\mathcal{G}_\pi^\lambda : \mathbb{N} \rightarrow \{0,1\} = \{(t,g) \mid t \in [1, |\pi|]$ and $g = 1$ if λ is locked by π at step t, 0 otherwise$\}$ such that $\mathcal{G}_\pi^\lambda(t) = g \; \forall \; (t,g) \in \mathcal{G}_\pi^\lambda$. The resultant usage profile of a resource λ due to all the plans in Ψ_α^P is obtained by summing over (weighted by the individual likelihoods) all the individual profiles as $\mathcal{G}^\lambda : \mathbb{N} \rightarrow [0,1] = \{(t,g) \mid t = 1, 2, \dots, max(|\pi|)$ and $g \propto \frac{1}{|\Psi_\alpha^P|} \sum_\pi \mathcal{G}_\pi^\lambda(t) \times l(\pi) \; \forall \; \pi \in \Psi_\alpha^P\}$.

Similarly, we can define profiles over the actual groundings of a variable (shown in the lower part of Figure 3) as $\mathcal{G}_\pi^{f^\lambda} = \{(t,g) \mid t \in [1, |\pi|]$ and $f^\lambda = 1$ at step t of plan π, 0 otherwise$\}$, and the resultant usage profile due to all the plans in Ψ_α^P is obtained as before as $\mathcal{G}^{f^\lambda} = \{(t,g) \mid t = 1, 2, \dots, max(|\pi|)$ and $g \propto \frac{1}{|\Psi_\alpha^P|} \sum_\pi \mathcal{G}_\pi^{f^\lambda}(t) \times l(\pi) \; \forall \; \pi \in \Psi_\alpha^P\}$. These profiles are helpful when actions in the robot's domain are conditioned on these variables, and the values of these variables are conditioned on the plans of the other agents in the environment currently under execution.

One important aspect of this formulation that should be noted here is that the notion of "resources" is described here in terms of the subset of the common predicates in the domain of the agents ($\xi \subseteq S_\alpha \cap S_R$) and can thus be used as a generalized definition to model different types of conflict between the plans between two agents. In as much as these predicates are descriptions (possibly instantiated) of the typed variables in the domain and actually refer to the physical resources in the environment that might be shared by the agents, we will stick to this nomenclature of calling them *"resources"*. We will now look at how an autonomous agent can use these resource profiles to minimize conflicts during plan execution with other agents in its environment.

1.4 Conflict Minimization

The planning problem of the robot - given by $\Pi = \langle D_R, \pi_R, \Lambda, \{G^\lambda \mid \forall \lambda \in \Lambda\}, \{G^{f^\lambda} \mid \forall f \in \Gamma(\lambda), \forall \lambda \in \Lambda\} \rangle$ - consists of the domain model D_R and the problem instance $\pi_R = \langle \mathbb{O}_R, \mathbb{I}_R, \mathbb{G}_R \rangle$ similar to that described in section 1.3, and also the constrained resources and all the profiles corresponding to them. This is because the planning process must take into account both goals of achievement as also conflict of resource usages as described by the profiles. Traditional planners provide no direct way to handle such profiles within the planning process. Note here that since the execution of the plans of the agents is occurring in parallel, the uncertainty is evolving at the time of execution, and hence the uncertainty cannot be captured from the goal states of the recognized plans alone, and consequently cannot be simply compiled away to the initial state uncertainty for the robot and solved as a conformant plan. Similarly, the problem does not directly compile into action costs in a metric planning instance because the profiles themselves are varying with time. Thus we need a planner that can handle these resource constraints that are both stochastic and non-stationary due to the uncertainty in the environment. To this end we introduce the following IP-based planner (partly following the technique for IP encoding for state space planning outlined in (Vossen et al. 1999)) as an elegant way to sum over and minimize overlaps in profiles during the plan generation process. The following formulation finds such T-step plans in case of non-durative or instantaneous actions.

For action $a \in A_R$ at step t we have an action variable:

$$x_{a,t} = \begin{cases} 1, & \text{if action a is executed in step t} \\ 0, & \text{otherwise;} \quad \forall a \in A_R,\ t \in \{1, 2, \ldots, T\} \end{cases}$$

Also, for every proposition f at step t a binary state variable is introduced as follows:

$$y_{f,t} = \begin{cases} 1, & \text{if proposition is true in plan step t} \\ 0, & \text{otherwise;} \quad \forall f \in S_R,\ t \in \{0, 1, \ldots, T\} \end{cases}$$

Note here that the plan being computed for the robot introduces a new resource consumption profile itself, and thus one optimizing criterion would be to minimize the overlap between the usage profile due to the computed plan with those established by the predicted plans of the other agents in the environment. Let us introduce a new variable to model the resource usage graph imposed by the robot as follows:

$$g_{f,t} = \begin{cases} 1, & \text{if } f \in \xi \text{ is locked at plan step } t \\ 0, & \text{otherwise;} \quad \forall f \in \xi,\ t \in \{0, 1, \ldots, T\} \end{cases}$$

For every resource $\lambda \in \Lambda$, the actions in the domain of the robot are divided into three sets - $\Omega_f^+ = \{a \in A_R \text{ such that } x_{a,t} = 1 \implies y_{f,t} = 1\}$, $\Omega_f^- = \{a \in A_R \text{ such that } x_{a,t} = 1 \implies y_{f,t} = 0\}$ and $\Omega_f^o = A_R \setminus (\Omega_f^+ \cup \Omega_f^-)$. These then specify respectively those actions in the domain that lock, free up, or do not affect the current use of a particular resource, and are used to calculate $g_{f,t}$ as part of the IP. Further, we introduce a variable $h_{f,t}$ to track preconditions required by actions in the generated plan that are conditioned on the plans of the other agents (e.g. position of the medkits are changing, and the action `pickup` is conditioned on it) as follows:

$$h_{f,t} = \begin{cases} 1, & \text{if } f \in \mathbb{P}_a \text{ and } x_{a,t+1} = 1 \\ 0, & \text{otherwise;} \quad \forall f \in \xi,\ t \in \{0, 1, \ldots, T-1\} \end{cases}$$

Then the solution to the IP should ensure that the robot only uses these resources when they are in fact most expected to be available (as obtained by maximizing the overlap between $h_{f,t}$ and G^{f^λ}). These act like *demand profiles* from the perspective of the robot.

We also add a new "no-operation" action $A_R \leftarrow A_R \cup a_\phi$ such that $a_\phi = \langle \mathbb{N}, \mathbb{C}, \mathbb{P}, \mathbb{E} \rangle$ where $\mathbb{N} = \texttt{NOOP}$, $\mathbb{C} = 0$, $\mathbb{P} = \{\}$ and $\mathbb{E} = \{\}$.

The IP formulation is given by:

$$\begin{aligned} &min\ k_1 \textstyle\sum_{a \in A_R} \sum_{t \in \{1,2,\ldots,T\}} \mathbb{C}_a \times x_{a,t} \\ &+k_2 \textstyle\sum_{\lambda \in \Lambda} \sum_{f \in \Gamma(\lambda)} \sum_{t \in \{1,2,\ldots,T\}} g_{f,t} \times G^\lambda(t) \\ &-k_3 \textstyle\sum_{\lambda \in \Lambda} \sum_{f \in \Gamma(\lambda)} \sum_{t \in \{0,1,\ldots,T-1\}} h_{f,t} \times G^{f^\lambda}(t) \end{aligned}$$

such that

$$y_{f,0} = 1\ \forall f \in \mathbb{I}_R \setminus \xi \tag{1}$$

$$y_{f,0} = 0\ \forall f \notin \mathbb{I}_R \text{ or } f \in \xi \tag{2}$$

$$y_{f,T} = 1\ \forall f \in \mathbb{G}_R \tag{3}$$

$$x_{a,t} \leq y_{f,t-1}\ \forall a\ s.t.\ f \in \mathbb{P}_a, \forall f \notin \xi, t \in \{1, \ldots, T\} \tag{4}$$

$$h_{f,t-1} = x_{f,t}\ \forall a\ s.t.\ f \in \mathbb{P}_a, \forall f \in \xi, t \in \{1, \ldots, T\} \tag{5}$$

$$\begin{aligned} &y_{f,t} \leq y_{f,t-1} + \textstyle\sum_{a \in add(f)} x_{a,t} \\ &\text{s.t. } add(f) = \{a | f \in eff^+(a)\}, \forall f, t \in \{1, \ldots, T\} \end{aligned} \tag{6}$$

$$\begin{aligned} &y_{f,t} \leq 1 - \textstyle\sum_{a \in del(f)} x_{a,t} \\ &\text{s.t. } del(f) = \{a | f \in eff^-(a)\}, \forall f, t \in \{1, \ldots, T\} \end{aligned} \tag{7}$$

$$\textstyle\sum_{a \in A_R} x_{a,t} = 1, t \in \{1, 2, \ldots, T\} \tag{8}$$

$$\textstyle\sum_{a \in \Omega_f^+} \sum_t x_{a,t} \leq 1\ \forall f \in \xi, t \in \{1, 2, \ldots, T\} \tag{9}$$

$$\begin{aligned} &g_{f,t} = \textstyle\sum_{a \in \Omega_f^+} x_{a,t} \\ &+(1 - \textstyle\sum_{a \in \Omega_f^+} x_{a,t} - \sum_{a \in \Omega_f^-} x_{a,t}) \times g_{f,t-1} \\ &\forall f \in \xi, t \in \{1, \ldots, T\} \end{aligned} \tag{10}$$

$$h_{f,t} \times G^{f^\lambda}(t) \geq \epsilon\ \forall f \in \xi, t \in \{0, 1, \ldots, T-1\} \tag{11}$$

$$y_{f,t} \in \{0,1\} \; \forall f \in S_R, t \in \{0,1,\ldots,T\} \tag{12}$$

$$x_{a,t} \in \{0,1\} \; \forall a \in A_R, t \in \{1,2,\ldots,T\} \tag{13}$$

$$g_{f,t} \in \{0,1\} \; \forall f \in S_R, t \in \{0,1,\ldots,T\} \tag{14}$$

$$h_{f,t} \in \{0,1\} \; \forall f \in S_R, t \in \{0,1,\ldots,T-1\} \tag{15}$$

where k_1, k_2, k_3 are constants (set manually) that determine the relative importance of each of the optimization criteria and ϵ is a small constant.

Here, the objective function minimizes the sum of the cost of the plan and the overlap between the cumulative resource usage profiles of the predicted plans and that imposed by the current plan of the robot itself while maximizing the validity of the demand profiles. Constraints (1) through (3) model the initial and goal conditions, while the value of the constrained variables are kept uninitialized (and are determined by their profiles). Constraints (4) and (5), depending on the particular predicate, enforces the preconditions, or produces the demand profiles respectively, while (6) and (7) enforces the state equations that maintain the add and delete effects of the actions. Constraint (8) imposes non concurrency on the actions, and (9) ensures that the robot does not repeat the same action indefinitely to increase its utility. Constraint (10) generates the resource profile of the current plan, while (11) maintains that actions are only executed if there is at least a small probability ϵ of success. Finally (12) to (15) provide the binary ranges of the variables.

2 Modulating the Behavior of the Robot

The IP-planner has been implemented on the IP-solver `guorbi` and integrates Ramirez *et. al.* (Ramrez and Geffner 2010) and `fast-downward` (Helmert 2011) respectively for goal recognition and plan prediction for the recognized goals. We will now go through a simplified use case, and illustrate how the resource profiles can be used to produce different behaviors of the robot by appropriately configuring the objective function and the length of the planning horizon of the IP formulation.

2.1 Compromise vs Opportunism

Let us look back at the setting in Figure 2. Consider that the robot recognizes that the goal of the commander is to perform triage in `room1`, computes his optimal plan (which ends up using `medkit1` at time steps 7 through 12) and updates the resource profiles accordingly. If now, it has its own goal to perform triage in `hall3`, the plan that it comes up with given a 12 step lookahead is shown below. Notice that the robot now opts to use the other medkit (`medkit2` in `room3`) even though its plan now incurs a higher cost in terms of execution. The robot thus can adopt a policy of *compromise* if it is possible for it to preserve the commander's (expected) plan.

```
01 - MOVE_ROBOT_ROOM1_HALL1
02 - MOVE_ROBOT_HALL1_HALL2
03 - MOVE_ROBOT_HALL2_HALL3
04 - MOVE_ROBOT_HALL3_HALL4
05 - MOVE_REVERSE_ROBOT_HALL4_ROOM4
06 - MOVE_REVERSE_ROBOT_ROOM4_ROOM3
07 - PICK_UP_MEDKIT_ROBOT_MK2_ROOM3
08 - MOVE_ROBOT_ROOM3_ROOM4
09 - MOVE_ROBOT_ROOM4_HALL4
10 - MOVE_REVERSE_ROBOT_HALL4_HALL3
11 - CONDUCT_TRIAGE_ROBOT_HALL3
12 - DROP_OFF_ROBOT_MK2_HALL3
```

Notice, however, that the commander is actually bringing the medkit to `room1` as predicted by the robot, and this is a favorable change in the world, because robot can use this `medkit` once the commander is done and incur a much lower cost of achieving its goal. The robot, indeed, realizes this once we give it a bigger time horizon to plan with, as shown below. Thus, in this case, the robot shows *opportunism* based on how it believes the world state will change.

```
01 - NOOP
02 - NOOP
03 - NOOP
   ...
12 - NOOP
13 - NOOP
14 - PICK_UP_MEDKIT_ROBOT_MK1_ROOM1
15 - MOVE_ROBOT_ROOM1_HALL1
16 - MOVE_ROBOT_HALL1_HALL2
17 - MOVE_ROBOT_HALL2_HALL3
18 - CONDUCT_TRIAGE_ROBOT_HALL3
19 - DROP_OFF_ROBOT_MK1_HALL3
```

2.2 Negotiation

In many cases, the robot will have to eventually produce plans that will have potential points of conflict with the expected plans of the commander. This occurs when there is no feasible plan with zero overlap between profiles (specifically $\sum g_{f,t} \times G^{\lambda}(t) = 0$) or if the alternative plans for the robot are too costly (as determined by the objective function). If, however, the robot is equipped with the ability to communicate with the human, then it can *negotiate* a plan that suits both. To this end, we introduce a new variable $H^{\lambda}(t)$ and update the IP as follows:

$$\begin{aligned} &min\; k_1 \textstyle\sum_{a \in A_R} \sum_{t \in \{1,2,\ldots,T\}} \mathbb{C}_a \times x_{a,t} \\ &+k_2 \textstyle\sum_{\lambda \in \lambda} \sum_{f \in \Gamma^{-1}(\lambda)} \sum_{t \in \{1,2,\ldots,T\}} g_{f,t} \times H^{\lambda}(t) \\ &-k_3 \textstyle\sum_{\lambda \in \Lambda} \sum_{f \in \Gamma^{-1}(\lambda)} \sum_{t \in \{0,1,\ldots,T-1\}} h_{f,t} \times G^{f^{\lambda}}(t) \\ &+k_4 \textstyle\sum_{\lambda \in \Lambda} \sum_{t \in \{0,1,\ldots,T\}} ||G^{\lambda}(t) - H^{\lambda}(t)|| \end{aligned}$$

$$y_{f,T} \geq h_{f,t-1} \; \forall \; a \; s.t. \; f \in \mathbb{P}_a, \forall f \in \xi, t \in \{1,\ldots,T\} \tag{5a}$$

$$H^{\lambda}(t) \in [0,1] \; \forall \lambda \in \Lambda, t \in \{0,1,\ldots,T\} \tag{16}$$

$$H^{\lambda}(t) \leq G^{\lambda}(t) \; \forall \lambda \in \Lambda, t \in \{0,1,\ldots,T\} \tag{17}$$

Constraint (5a) now complements constraint (5) from the existing formulation, by promising to restore the world state every time a demand is made on a variable. The variable $H^{\lambda}(t)$, maintained by constraints (16) and (17), determine the desired deviation from the given profiles. The objective function has been updated to reflect that overlaps are now measured with the desired profile of usage, and there is a

T		Number of Observations								
		1	2	3	4	5	6	7	8	9
T=10	C	9	9	9	9	9	9	8.84	8.81	8.84
	U	0.39	0.417	0.394	0.399	0.406	0.35	0.36	0.484	0.429
	S	1	1	1	1	1	1	0.96	0.955	0.96
T=15	C	5.5	5.23	5.26	5.27	5.3	5.38	5.2	5.39	5.41
	U	0.007	0.008	0.007	0.008	0.006	0.002	0.008	0.01	0.009
	S	0.5	0.467	0.456	0.464	0.453	0.442	0.457	0.55	0.508
T=20	C	5.34	5.23	5.26	5.27	5.3	5.2	4.86	5.09	5.19
	U	0.004	0.004	0.004	0.004	0.003	0.001	0.004	0.008	0.006
	S	0.46	0.467	0.457	0.464	0.453	0.412	0.394	0.495	0.465
T=25	C	5.28	5.16	5.21	5.207	5.24	5.2	4.857	5.095	5.15
	U	0.003	0.003	0.003	0.003	0.002	0.001	0.003	0.007	0.004
	S	0.46	0.458	0.455	0.455	0.444	0.412	0.397	0.499	0.459

Table 1: Performance metrics w.r.t. number of observations

Time		Size of the Hypothesis Goal Set $\|\Psi_\alpha\|$					
		1	2	3	4	5	6
T=10	C	9	8.95	8.74	8.95	8.75	8.73
	U	0.345	0.364	0.54	0.228	0.55	0.42
	S	1	0.988	0.93	0.98	0.94	0.93
T=15	C	7.32	6.34	5.26	5.65	3.65	4.44
	U	0.015	0.004	0.012	0.005	0.011	0.004
	S	1	0.683	0.45	0.365	0.322	0.27
T=20	C	7.32	6.34	5.1	5.14	3.35	4.07
	U	0.009	0.002	0.007	0.003	0.006	0.002
	S	1	0.68	0.431	0.255	0.27	0.192
T=25	C	7.32	6.18	5.1	5.14	3.35	4.07
	U	0.006	0.002	0.005	0.002	0.004	0.002
	S	1	0.663	0.432	0.255	0.27	0.192

Table 2: Performance metrics w.r.t. size of the goal set

cost associated with the deviation from the real one. The revised plan now produced by the robot is shown below.

```
01 - MOVE_ROBOT_ROOM1_HALL1
02 - MOVE_ROBOT_HALL1_HALL2
03 - MOVE_REVERSE_ROBOT_HALL2_ROOM2
04 - PICK_UP_MEDKIT_ROBOT_MK1_ROOM2
05 - MOVE_ROBOT_ROOM2_HALL2
06 - MOVE_ROBOT_HALL2_HALL3
07 - CONDUCT_TRIAGE_ROBOT_HALL3
08 - MOVE_REVERSE_ROBOT_HALL3_HALL2
09 - MOVE_REVERSE_ROBOT_HALL2_ROOM2
10 - DROP_OFF_ROBOT_MK1_ROOM2
```

Notice that the robot restores the world state that the human is believed to expect, and can now communicate to him *"Can you please not use* `medkit1` *from time 7 to 9?"* based on how the real and the ideal profiles diverge, i.e. t such that $H^\lambda(t) < G^\lambda(t)$ for each resource λ.

3 Evaluation

We ran our scenario (with one human and one robot, and two medkits) on 400 problem instances, randomly generated by varying the specific (as well as the number of probable) goals of the human, and evaluated how the planner behaved with the number of observations it can start with to build its profiles. To generate the test cases, we first fix the domain description, location and goal of the agents, and the position of the resources. Then we consider 10×6 randomly generated hypothesis goal sets each of size 1 through to 6. The goals of the commander were assumed to be known to be triage related, but the location of the triage was allocated randomly, and one of the possible goals were again picked at random as the real goal. Finally for each of these problems, we generate 1-9 observations for each of these problems by simulating the commander's plan over the real goal, and plan with these observations known*a priori* the robot's plan generation process. The experiments were conducted on a Intel Xeon(R) CPU E5-1620 v2 3.70GHz$\times$8 processor with a 62.9GiB memory. The planner is available at http://bit.ly/1QHt21Q.

3.1 Scaling Up

Note that our primary contribution is the IP-formulation for planning with resource profiles, while the goal recognition component can be any off-the-shelf algorithm, and as such

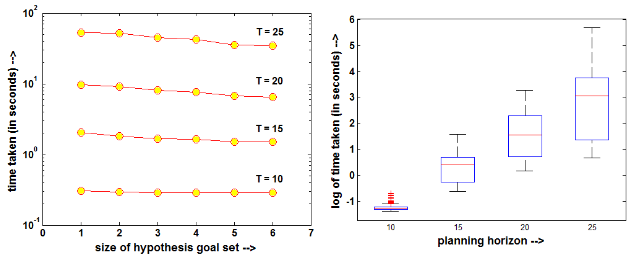


Figure 4: Performance of the planner with increasing number of possible goals and with increasing planning horizon.

we compare the scalability with respect to the planning component only. Indeed, our planner only consumes 0.2-27% (for $T = 10$ to 25 steps) of the total CPU time.

w.r.t. the Number of Agents and the Size of the Hypothesis Goal Set The IP formulation is independent of the number of agents being modeled. In fact, this is one of the major advantages of using abstractions like resource profiles in lieu of actual plans of each of the agents. On the other hand, the time spent on recognition, and on calculating the profiles, is significantly affected. However, observations on multiple agents are asynchronous, and goal recognition can operate in parallel, so that this is not a huge concern beyond the complexity of a single instance. Similarly the performance is also largely unaffected by the number of possible goals in Ψ_α, as shown in Figure 4.

w.r.t. Length of the Planning Horizon The performance of the planner with respect to the length of the planning horizon is shown in Figure 4 in terms of a box plot. This is the biggest bottleneck in the computation due to the exponential growth in the size of the IP.

3.2 Quality of the Plans Produced

Tables 1 and 2 point out some interesting aspects of planning with resource profiles. We define the U as the average conflict per step of the plan when a demand on a resource is placed by the robot, and S as the success probability per plan step that the demand is met. Notice that the average conflict goes down with increasing planning horizon T, which indicates opportunistic behavior on the part of the robot, while the average cost C of the plans is higher for lower T, which indicates that the robot has to compromise towards higher cost plans in case of conflicts. Also note how the algorithm is quite robust with respect to the number of observations available *a priori*, indicating that the robot need not wait long to find good plans. Further, U falls drastically with higher T, which indicates that given longer plan lengths the robot is able to effectively identify lower conflict time steps to act. However, S also falls with higher T which might seems unintuitive at first, but it really means that with lesser options the robot chooses safer plans at a higher execution cost. Indeed the exact tradeoff in this behavior can be modulated by appropriately configuring the objective function of the planner.

4 Conclusions

In this paper we look at how plans may be affected by conflicts on shared resources in an environment cohabited by humans and robots, and introduce the concept of resource profiles to model the usage of such resources. We also propose a general formulation to plan in such scenarios and provide a complete framework of obtaining and using these profiles in conjunction with this planner. Finally, we show how the planner can be used to model different types of behavior of the autonomous agents. One interesting research question would be to extend the current formulation to consider nested beliefs on the agents; after all, humans are rarely completely aloof of other agents in its environment. Also, currently we only assume non-durative actions, and completely known models of the human and completely observable worlds, which we hope to relax in future works.

Acknowledgments

This research is supported in part by the ARO grant W911NF-13-1-0023, and the ONR grants N00014-13-1-0176, N00014-13-1-0519 and N00014-15-1-2027.

References

[Beaudry, Kabanza, and Michaud 2010] Beaudry, E.; Kabanza, F.; and Michaud, F. 2010. Planning with concurrency under resources and time uncertainty. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, 217–222. Amsterdam, The Netherlands, The Netherlands: IOS Press.

[Blum and Furst 1995] Blum, A., and Furst, M. L. 1995. Fast planning through planning graph analysis. In *IJCAI*, 1636–1642.

[Cavallo et al. 2014] Cavallo, F.; Limosani, R.; Manzi, A.; Bonaccorsi, M.; Esposito, R.; Di Rocco, M.; Pecora, F.; Teti, G.; Saffiotti, A.; and Dario, P. 2014. Development of a socially believable multi-robot solution from town to home. *Cognitive Computation* 6(4):954–967.

[Cirillo, Karlsson, and Saffiotti 2009] Cirillo, M.; Karlsson, L.; and Saffiotti, A. 2009. Human-aware task planning for mobile robots. In *Proc of the Int Conf on Advanced Robotics (ICAR)*.

[Cirillo, Karlsson, and Saffiotti 2010] Cirillo, M.; Karlsson, L.; and Saffiotti, A. 2010. Human-aware task planning: An application to mobile robots. *ACM Trans. Intell. Syst. Technol.* 1(2):15:1–15:26.

[Helmert 2011] Helmert, M. 2011. The fast downward planning system. *CoRR* abs/1109.6051.

[Koeckemann, Pecora, and Karlsson 2014] Koeckemann, U.; Pecora, F.; and Karlsson, L. 2014. Grandpa hates robots - interaction constraints for planning in inhabited environments. In *Proc. AAAI-2010*.

[Kuderer et al. 2012] Kuderer, M.; Kretzschmar, H.; Sprunk, C.; and Burgard, W. 2012. Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of Robotics: Science and Systems*.

[Ramrez and Geffner 2010] Ramrez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *In Proc. AAAI-2010*.

[Sisbot et al. 2007] Sisbot, E.; Marin-Urias, L.; Alami, R.; and Simeon, T. 2007. A human aware mobile robot motion planner. *Robotics, IEEE Transactions on* 23(5):874–883.

[Talamadupula et al. 2014] Talamadupula, K.; Briggs, G.; Chakraborti, T.; Scheutz, M.; and Kambhampati, S. 2014. Coordination in human-robot teams using mental modeling and plan recognition. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2957–2962.

[Tomic, Pecora, and Saffiotti 2014] Tomic, S.; Pecora, F.; and Saffiotti, A. 2014. Too cool for school adding social constraints in human aware planning. In *Proc of the International Workshop on Cognitive Robotics (CogRob)*.

[Vossen et al. 1999] Vossen, T.; Ball, M. O.; Lotem, A.; and Nau, D. S. 1999. On the use of integer programming models in ai planning. In Dean, T., ed., *IJCAI*, 304–309. Morgan Kaufmann.