Temporal coordination under uncertainty: initial results for the two agents case

Emmanuel Rachelson
ONERA-DCSD
2, avenue Edouard Belin
F-31055 Toulouse, France
emmanuel.rachelson@onera.fr

Abstract

We focus on the problem of decentralized planning and coordination for two heterogeneous autonomous agents, having a common mission in an uncertain environment. For example, we consider a helicopter UAV and a ground rover cooperating in the exploration of a dangerous zone where communication is limited, which forces decentralization of planning. After proposing a framework for decentralized planning, we underline the need for a planner under uncertainty taking continuous time into account in time-dependent problems and present initial results on temporal planning under uncertainty.

Decentralized coordination of independent policies

Imagine a situation where a helicopter UAV and a rover cooperate in the search for a specific target in an unknown operation zone. Available individual actions deal with movement, observation and interaction with the environment, but these actions' efficiency can be greatly improved when agents cooperate and coordinate their strategies. For communication reasons, both agents plan separately. Since their environment is unknown and uncertain, we assume they have a Markov Decision Process (Puterman 1994) representation of their world. However, since planning is a decentralized task, no agent has authority on the other to impose a global plan. This problem can thus be seen as a decentralized planning process for Dec-MDPs (Bernstein, Zilberstein, & Immerman 2000). In order to build an efficient overall plan, the agents need to coordinate their strategies by communicating to each other some relevant information about how their current strategy affects the environment.

Therefore, following the general idea of (Chades, Scherrer, & Charpillet 2002), we imagined the communication and coordination protocol summarized in figure 1. This framework for coordination uses individual planning among agents. Initially, a set of common variables is defined between the two agents. These variables describe the problem's aspects that are common to both agents and will serve as communication variables. The agents generate their own initial policy with the initial knowledge they have of the problem. Then, using the communication variables, they send to each other a message describing the effect of their own strategy upon the common variables. This message describes the probabilistic timed evolution of the communication variables under an agent's actions. At that point, both agents integrate the message information into their model of the world and correct their plan accordingly. Then the process starts over in order to refine the obtained plans. This method doesn't guarantee to find the optimal global policy for the pair of agents but (Chades, Scherrer, & Charpillet 2002) proved it was a good heuristic in the search for an acceptable coordinated strategy.

One can notice that in the framework we present, two global strategies are actually initiated and improved through exchanges of information. These two strategies can be evaluated through the agents' individual value functions and therefore can be compared at each step. Consequently, this method can be stopped at any time during the information exchanges: the best valued policy is then chosen and applied. These information exchanges can be seen as "intention assessments" and therefore as a purely cooperative protocol for agents coordination. Refinements for communication stopping, common actions (actions that require the existence of the pair of agents as a third virtual agent) coordination and policy update can be set up on top of the existing protocol and will not be discussed here. If used during a pre-mission phase, the framework presented above is an offline coordination scheme where individual replanning is triggered by a new intention declared by the other agent. Whereas during mission execution, observations and plan changes can trigger online replanning. This behaviour is illustrated by the bottom part of figure 1.

In the above paragraphs, we assumed each agent had an individual planning algorithm able to deal with problems presenting two important aspects: uncertainty about the action's outcomes and time-dependency of the problem's data. For example, a specific place can be very dangerous for the rover to explore and therefore be associated with a very weak survival probability, but while the helicopter patrols over the zone, the survival probability becomes acceptable to undertake an exploration. Therefore, the planner needs to take into account the fact that exploring the zone is a probabilistic ac-

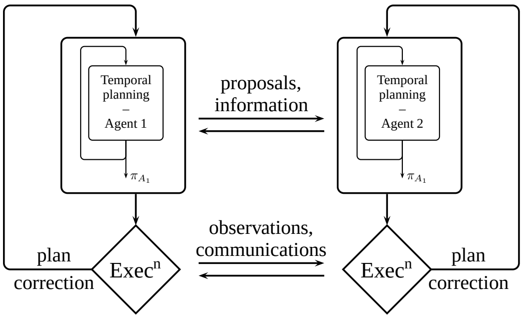

Figure 1: Decentralized coordination and cooperation

tion and that its characteristics are time-dependent (here the time-dependency comes from the intentions declared by the other agent). Similarly, the rewards obtainable can be time dependent (for example if a reward changes continuously with time). These two crucial aspects were already pointed out by (Bresina *et al.* 2002). We investigate temporal planning under uncertainty in the next section and propose two methods for approximating good policies.

Temporal planning under uncertainty

Since the problematic of time dependent problems of planning under uncertainty reaches beyond the scope of policy coordination, we investigated general models for temporal planning under uncertainty. For example, planning to take a picture at a certain time with an Earth observation satellite has a certain probability of success according to the cloud cover beneath it. Markov decision processes (MDPs) are usually used to deal with uncertainty in the result of actions (Puterman 1994). An MDP can be represented as a countable set of states S, a set of actions A, a transition function $P(s'|s,a)$ giving the probability of arriving in state s' when we undertake action a in state s, and a reward model $r(s,a)$ representing the reward associated with the transition (s,a). Solutions to MDPs, are often given as policies mapping states to actions, these policies being optimized according to a certain criterion. MDP policies can be optimized using linear or dynamic programming (Bellman 1957) algorithms such as value or policy iteration.

Unfortunately, MDPs represent stationary problems or discrete change problems. For the example of our satellite (or the case of our policies coordination), the evolution of the clouds is a continuous function of time, so is the probability of success of our photography. We aim at defining and approximating optimal policies for MDP problems that present a continuous-time evolution. In the agent coordination example, if one agent is able to plan in a dynamic, continuously changing environment, then it can integrate the other agent's plan effects as an evolution of its own model in order to coordinate their actions.

Temporal planning is a topic that has been widely covered in deterministic planning (with the IxTeT planner for example (Ghallab & Laruelle 1994)) and in models that included some uncertainty about action durations (Wellman, Ford, & Larson 1995). Most models that deal with temporal planning under uncertainty in the MDP framework explore the possibilities of different and uncertain action durations and asynchronous actions (Younes & Simmons 2004; Mausam & Weld 2005), or deal with continuous state spaces without emphasis on the time variable (Hauskrecht & Kveton 2006). There are two main models in the literature that take a continuous-time evolution into account in an MDP framework. The semi-Markov decision processes (SMDP) model describes transitions with a function $Q(s',t'|s,a) = P(s'|s,a) \cdot F(t'|s,a)$ indicating the probability to end up in state s' after a duration t' if we undertake a in s. SMDPs are useful to take duration costs into account but are limited by some strong hypothesis of stationarity, independence of t' and s' and the absence of exogenous events. (Boyan & Littman 2001) introduced the TMDP model in order to deal with this kind of problems.

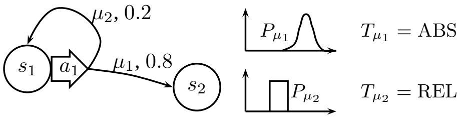

Figure 2: TMDP

The TMDP model deals with time dependent decision problem with uncertainty on the action's outcomes. A TMDP transition is described by a set of outcomes μ and an outcome realization likelihood function $L(\mu|s,a,t)$. When we undertake action a in state s at time t we have a probability $L(\mu|s,a,t)$ of realizing outcome μ; an outcome being a triplet (s'_μ, T_μ, P_μ) where s'_μ is the arrival state of the outcome and P_μ is the probability density function describing the duration (if $T_\mu = REL$) or the ending date (if $T_\mu = ABS$) of the outcome. This model is illustrated on figure 2. Formally, a TMDP is described as a discrete state space S, a discrete action space A, a set M of outcomes $\mu = (s'_\mu, T_\mu, P_\mu)$, an outcome likelihood function $L(\mu|s,a,t)$ giving the probability of realizing outcome μ when undertaking action a at t in state s, a reward function $R(\mu,t,t')$ (t and t' being the beginning and ending dates of the outcome) and a "dawdling" cost $K(s,t)$ representing the instantaneous cost at t of a virtual "wait" action in s.

Initial results for approximate temporal planning under uncertainty

We have proposed two different algorithms for approximating good policies for TMDP-like problems. Classical TMDPs can be solved exactly if P_μ are discrete probability distributions, L are piecewise constant and R is decomposable into piecewise linear additive functions. The first algorithm extends the classical TMDP representation to more general classes of functions (namely piecewise polynomial functions and probability density functions). The second algorithm, named SMDP+, is based on the search for the smallest set of dates necessary to build an efficient time-dependent policy. These two methods lead to a generalization of temporal planning under uncertainty to a model of parametric action MDPs, named XMDP, which is detailed in (Rachelson, Teichteil, & Garcia 2007).

The optimality equations for TMDPs as given in (Boyan & Littman 2001) are:

$$V(s,t) = \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta) d\theta + \overline{V}(s,t') \right) \tag{1}$$

$$\overline{V}(s,t) = \max_{a \in A} Q(s,t,a) \tag{2}$$

$$Q(s,t,a) = \sum_{\mu \in M} L(\mu | s,t,a) \cdot U(\mu,t) \tag{3}$$

$$U(\mu,t) = \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' & (*) \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' & (\dagger) \end{cases} \tag{4}$$

$$(*) \text{ if } T_\mu = \text{ABS} \qquad (\dagger) \text{ if } T_\mu = \text{REL}$$

We proved that if all P_μ, L, R and K functions were piecewise polynomial then V was piecewise polynomial and, with an approximation scheme used to reduce V's degree to keep it stable through the value iterations of equations 1 to 4, we could approximate the optimal value function with piecewise polynomial function. This first approach is currently being implemented and tested.

The second approach is based on the idea that, in a given state s, the optimal time-dependent policy can be represented as a finite (and supposedly small) set of "time interval, action" pairs. Thus, finding the best policy turns out to be the task of finding the best partitioning of the time axis and the best action to undertake on each time interval. The SMDP+ algorithm we developed uses the t-Bellman error as a heuristic to find the best partitioning dates. Details are provided in (Rachelson *et al.* 2006) and the algorithm is illustrated on figure 3. The SMDP+ algorithm follows four main steps. First, it discretizes the continuous problem written as a TMDP or a generalized SMDP, using the initial partitioning of the time axis (this partitioning may be trivial for initialization) adding a state variable corresponding to the current time interval. Then it finds an optimal policy $\tilde{\pi}$ for this discretized MDP $\tilde{M}$. It defines a policy π on the continuous time variable by identification to $\tilde{\pi}$ and calculates the t-Bellman error of this policy π. The t-Bellman error is, per state, the function of time giving the measure of how much we can improve the current value function by performing a single Bellman backup. We find the t that maximizes this error and use it as a heuristic to repartition the time axis. We adapt the discretization and iterate back to the first step. As the cache of decision dates grows, we insert a simplification step between step 2 and 3 in order to merge any consecutive intervals where the action specified by $\tilde{\pi}$ is the same. This way, we maintain, per state, a minimal cache of decision dates.

This SMDP+ algorithm is quite close to policy iteration approaches, future research will involve investigating the link between them. The SMDP+ algorithm is the next step in our implementation and testing process. Since we deal with state variables and factored representations we wish to exploit the properties of factored MDPs and the techniques of Approximate Linear Programming for evaluation of the value functions in the algorithm.

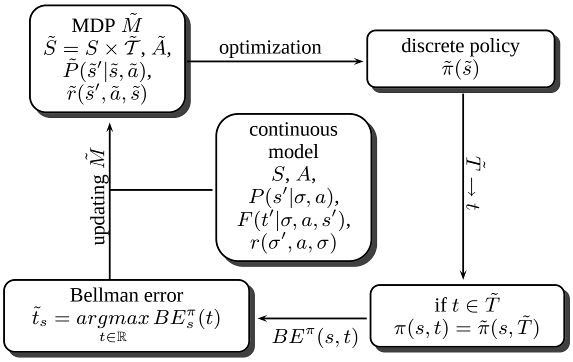
MDP $\tilde{M}$
$\tilde{S} = S \times \tilde{\mathcal{T}}$, $\tilde{A}$,
$\tilde{P}(\tilde{s}'|\tilde{s},\tilde{a})$,
$\tilde{r}(\tilde{s}',\tilde{a},\tilde{s})$
optimization
discrete policy
$\tilde{\pi}(\tilde{s})$
continuous model
S, A,
$P(s'|\sigma,a)$,
$F(t'|\sigma,a,s')$,
$r(\sigma',a,\sigma)$
updating $\tilde{M}$
$\tilde{T} \rightarrow t$
Bellman error
$\tilde{t}_s = \underset{t \in \mathbb{R}}{argmax}\, BE^\pi_s(t)$
$BE^\pi(s,t)$
if $t \in \tilde{T}$
$\pi(s,t) = \tilde{\pi}(s,\tilde{T})$

Figure 3: The SMDP+ algorithm

Future research will deal with improving the current techniques for temporal probabilistic planning, comparing them to existing results, and finally integrating them into the global coordination framework we defined initially.

References

Bellman, R. 1957. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey.

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of markov decision processes. In *16th Conf. on Uncertainty in AI.*

Boyan, J. A., and Littman, M. L. 2001. Exact solutions to time dependent MDPs. *Advances in Neural Information Processing Systems* 13:1026–1032.

Bresina, J.; Dearden, R.; Meuleau, N.; Ramkrishnan, S.; Smith, D.; and Washington, R. 2002. Planning under continuous time and resource uncertainty: A challenge for AI. In *18th Conference on Uncertainty in AI.*

Chades, I.; Scherrer, B.; and Charpillet, F. 2002. A heuristic approach for solving decentralized-POMDP: assessment on the pursuit problem. In *ACM symposium on Applied computing.*

Ghallab, M., and Laruelle, H. 1994. Representation and control in IxTeT a temporal planner. In *AIPS 94.*

Hauskrecht, M., and Kveton, B. 2006. Approximate linear programming for solving hybrid factored MDPs. In *9th Intl Symp. on AI and Mathematics.*

Mausam, and Weld, D. 2005. Concurrent probabilistic temporal planning. In *ICAPS.*

Puterman, M. L. 1994. *Markov Decision Processes.* John Wiley & Sons, Inc.

Rachelson, E.; Fabiani, P.; Farges, J.; Teichteil, F.; and Garcia, F. 2006. Une approche du traitement du temps dans le cadre MDP : trois méthodes de découpage de la droite temporelle. In *Journées Françaises Planification Décision Apprentissage.* F. Garcia, G. Verfaillie editors.

Rachelson, E.; Teichteil, F.; and Garcia, F. 2007. XMDP : un modèle de planification temporelle dans l'incertain à actions paramétriques. In *Journées Françaises Planification Décision Apprentissage.* F. Garcia, G. Verfaillie editors.

Wellman, M.; Ford, M.; and Larson, K. 1995. Path planning under time-dependent uncertainty. In *11th Conf. on Uncertainty in AI*, 532–539.

Younes, H. L. S., and Simmons, R. G. 2004. Solving generalized semi-markov decision processes using continuous phase-type distributions. In *AAAI04.*