State-Time Decomposition for Estimating Time-Dependent Utility *

Jun Miura and **Yoshiaki Shirai**
Dept. of Computer-Controlled Mechanical Systems, Osaka University
Suita, Osaka 565-0871, Japan
email: jun@mech.eng.osaka-u.ac.jp
URL: http://www-cv.mech.eng.osaka-u.ac.jp/~jun/

Abstract

This paper discusses the estimation of time-dependent utility that is used for decision-theoretic planning in dynamic environments. A general approach to modeling the time-dependent utility is presented which is based on the decomposition of time-dependent utility into time-dependent state description (*SEM: situation evolution model*) and state-dependent utility description (*PUM: plan utility model*). This approach makes the designing process of time-dependent utility clearer and the estimating process less costly in some cases. An example application to a robot motion planning in a dynamic environment is described. The relationships between the proposed approach and MDP-based approaches is also discussed.

Introduction

Decision-theoretic planning is an effective way to generate a plan under uncertainty of available information (Blythe 1999). Many works applied decision theory to vision and/or motion planning tasks (e.g., (Hutchinson & Kak 1989) (Cameron & Durrant-Whyte 1990) (Miura & Shirai 1997a)) of autonomous agents (robots). These works, however, assumed that the environment is static.

In a dynamic environment, the state evolves as time advances; a plan generated for the current state may not be effective in future states. For example, a plan to move towards the current target position may be useless if the target changes the position frequently. Usually an agent in a dynamic environment has to consider the balance between: (1) the increasing plan quality by obtaining more information or by searching for a better plan and (2) the decreasing plan quality by loosing a good opportunity of executing a plan. *Time-dependent utility* is an appropriate representation to be used for this balancing task in a context of decision-theoretic planning (Zilberstein 1993).

In planning in a dynamic environment, therefore, an important issue is how to represent utilities which are inherently time-dependent. A simple way is to assess the utility every time it is needed; however this approach provides no guidelines in modeling time-dependent utility; also, this could be computationally expensive. Thus we propose a systematic way to obtain time-dependent utility values, based on the concept of *state-time decomposition*. We applied the proposed approach to a mobile robot planning problem in the environment where moving obstacles exist.

*In Proc. AIPS-2000 Workshop on Decision-Theoretic Planning, pp. 64-68.

Related Works on Time-Dependent Utility

In a dynamic environment, utility of an action (or plan) changes over time. Usually the utility of a selected action decreases as time advances because the action is most suitable for a specific past state and is not the best for the current state in general.

Zilberstein (1993) defined the *time-dependent utility* as follows: a utility function $U(r, s, t)$, that measures the value of a result r in situation s at time t, is said to be *time-dependent* if:

$$\exists r, s, t_1, t_2 \quad U(r, s, t_1) \neq U(r, s, t_2).$$

Russell and Wefald (1991) discussed the notion of the *cost of time*. This reflects the loss of utility due to deliberation, sensing, or delay in action. Using the above definition of utility, the cost of time $C(t)$ is defined as:

$$C(t) = U(r_t, s_0, 0) - U(r_t, s_t, t),$$

where r_t is the result at time t and s_t is the state at t. This is the difference between the utility of r_t assuming that it is obtained now and that of r_t at the time it is actually obtained. In simple cases where the cost of computation is independent of the action, the cost of time depends only on time t; for example, if the change of the environment is slow enough compared with the agent's reasoning speed, the loss of utility due to deliberation is the time of deliberation itself. In such a case, a time-dependent utility can easily be described as:

$$U(r_t, s_t, t) = U(r_t, s_0, 0) - C(t). \quad (1)$$

The first term of the right side is often referred to as *intrinsic utility*, which corresponds to the utility in a static environment. The equation indicates how much the utility decreases from the intrinsic utility as time advances.

Horvitz and Rutledge (1991) proposed the following two types of functions to represent the time-dependent decrease of the utility:

$$\begin{aligned} u(A_i H_j, t) &= u(A_i H_j, t_0)e^{-k_a t} & \text{(exponential)} \\ u(A_i H_j, t) &= u(A_i H_j, t_0) - c_b t, & \text{(linear)} \\ & \quad (u(A_i H_j, t) \geq 0), \end{aligned}$$

where $u(A_i H_j, t)$ indicates the value of action A_i executed at time t when state H_j is true, k_a and c_b are parameter constants derived through assessments with real data.

Modeling the time-dependent utility with the intrinsic utility and the time-dependent decrease of utility seems suitable for approximating (or inducing) time-dependent utility in the case where the mechanism of utility change is not well understood. However, if the mechanism is well modeled, a more constructive approach is effective. In addition, the change of utility may not be represented by a simple form as above.

State-Time Decomposition for Estimating Time-Dependent Utility

In this paper, we consider the case where the time dependency of the utility mainly comes from the dynamics of the environment. In such a case, the dynamics has an effect not directly on the utility but on the future state of the environment; in other words, the dynamics has an *indirect* effect on the utility via the state in the future.

To model this indirect effect from the dynamics to the utility, we propose to decompose the time-dependent utility into the state-dependent utility and the time-dependent state. Two decomposed parts are called the *state evolution model* (SEM), and the *plan utility model* (PUM), respectively (see Figure 1). The time-dependent utility is thus estimated through the two-step consideration of dependency.

This two-step approach is quite natural. However, by *explicitly* manifesting the decomposition, we think we will obtain the following two merits:

1. The designing process becomes easier. This decomposition provides a clear guideline to determine what should be modeled for estimating time-dependent utility.

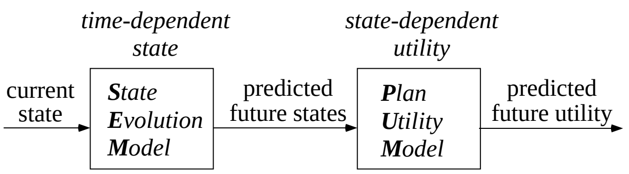

Figure 1: State-time decomposition of time-dependent utility using SEM and PUM.

2. The estimating process may become less costly. Since we make two mappings, one is from time to state and the other is from state to utility, in advance, we do not have to perform utility calculation from scratch. When two different evolution processes reach the same state, state-based utility estimation needs to be done only once.

State Evolution Model (SEM)

A state evolution model (SEM) describes how the situation evolves as time advances. It is a function of the current state $\boldsymbol{s}_0$, action (or action sequence) a to be executed, and time t. This model describes the prediction ability of an agent. Since the prediction usually includes uncertainty, the function returns a set of possible states $\{\boldsymbol{s}_i^t\}$ and their probabilities $\{P_i^t\}$. Sources of uncertainty include ones due to an agent as well as other agents or the dynamics of the environment. Action a can be omitted from the model if the agent's action does not affect the dynamics of the environment.

Plan Utility Model (PUM)

A plan utility model (PUM) describes the relationship between state-action pairs and utilities. It is a function of action (or action sequence) a and state vector $\boldsymbol{s}$, which returns a real-valued utility u. There is no restrictions on the form of the function; it may be an analytically derived closed-form function or may be a lookup table describing a mapping from action-state pairs to utilities.

Decision Making using SEM and PUM

Using SEM and PUM, planning is performed as follows. We here consider an agent that repeatedly selects and executes the next one action. The best next action is selected with a multi-step lookahead search. Evaluation of each candidate action is given by calculating the maximum expected utility of the corresponding AND/OR tree as shown in Figure 2. Possible states to be obtained by executing an action are predicted using the SEM. For each predicted state below the candidate action, the best subsequent action is selected; for a *certain* state at which the best action can be selected deterministically (this state is a leaf node of the AND/OR tree), the utility is calculated using PUM; for an *uncertain* state at whcih the best action cannot be determined without further search, the best action is recursively selected which maximizes the expected utility. The search depth may be limited using a certain threshold or using a meta-level control considering the tradeoff between planning cost and the plan quality (Miura & Shirai 1997b).

Example Problem: Mobile Robot Navigation in a Dynamic Environment

This section describes an example planning problem in mobile robot domain. Figure 3 illustrates an example

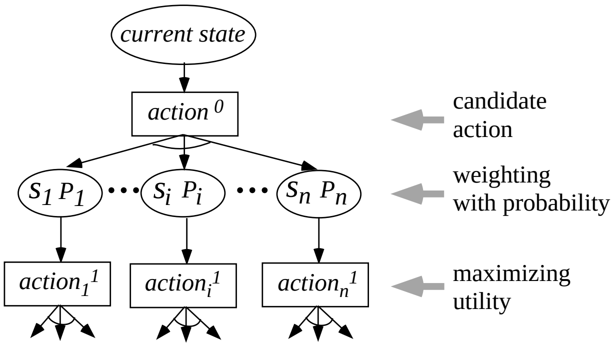

Figure 2: AND/OR tree for a candidate action.

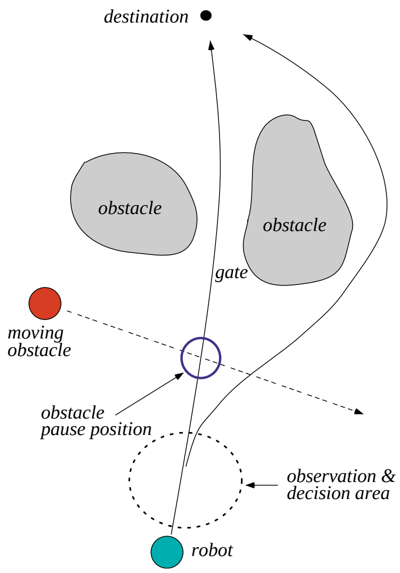

Figure 3: Simulation problem.

situation. A vision-based mobile robot, which has a rough map of the environment, is going to the destination while selecting routes. There is a short route which passes the narrow space (called the *gate*) between static obstacles; however the passability of the gate is initially unknown due to the uncertainty of visual data obtained at the initial position. The detour on the right side is known to be passable, although it is longer. A moving obstacle is going to cross the robot's route towards the gate. It will also make a pause for a predetermined period at the crossing point; this forces the robot to wait for the obstacle leaving.

There is an area in front of the robot where the robot would make observation of the gate to determine if it is passable, and based on the observation result, it would make a decision on route selection. If the gate is passable, the robot takes the left route; if the gate is impassable, it takes the right; otherwise, it observes the gate again while moving. The option the robot has is the speed inside the area, at which it moves while observing the gate. If the passability is determined, or if the robot exits from the area, the robot moves at the predetermined maximum speed. If the passability is still unknown when the robot exits the area, it takes the detour.

Tradeoff to be Considered

In this problem, the robot changes the speed in the observation & decision area. If the robot moves slowly, it can observe the gate many times, thereby obtaining more accurate information for better decision; but at the same time, the possibility of being obstructed by the moving obstacle increases. If the robot move fast, it observe the gate only a few times and more likely to give up the left route, while it is less likely to be obstructed. Therefore, the robot has to trade the increase of plan quality by obtaining more information with the decrease of plan quality by being obstructed by the moving obstacle.

State Evolution Model

The moving obstacle is the only activity in the environment. Its movement is modeled using a simple linear motion model, that is, the future position of the obstacle is predicted by assuming that it will continue moving at the current speed. No uncertainty is considered for the obstacle's motion in this paper. In a real situation, however, various sources of uncertainty exist in predicting obstacles' movement such as: the obstacle motion itself cannot be completely predictable; the observation of the obstacle position and velocity includes uncertainty. We have developed a more elaborated probabilistic model of prediction uncertainty(Miura & Shirai 2000). Such a model can be incorporated in the state evolution model.

The other factor described in the SEM is the observation result of the gate, in which vision uncertainty is considered. We adopt a simplified version of the uncertainty model of vision which we have already developed (Miura & Shirai 1997a). From this model, the passability (i.e., being passable or impassable) of the gate is determined with a certain probability, which increases as the distance from the robot to the gate is smaller. As observations are repeatedly made, the probability of the gate being passable or impassable increases. This is modeled as follows.

$$\begin{aligned} P^0_{ok} &= P^0_{ng} = 0, \\ P^0_{ud} &= 1, \\ P^i_{ok} &= P^i_{obs} * P_{true} * P^{i-1}_{ud}, \\ P^i_{ng} &= P^i_{obs} * (1 - P_{true}) * P^{i-1}_{ud}, \\ P^i_{ud} &= (1 - P^i_{obs}) * P^{i-1}_{ud}, \end{aligned}$$

where P^i_{ok} (P^i_{ng}, P^i_{ud}) is the probability that the gate's state become known to be passable (impassable, unknown) by the ith observation; P^0_{ok}, P^0_{ng}, and P^0_{ud} are the initial probabilities, P^i_{obs} is the probability that the

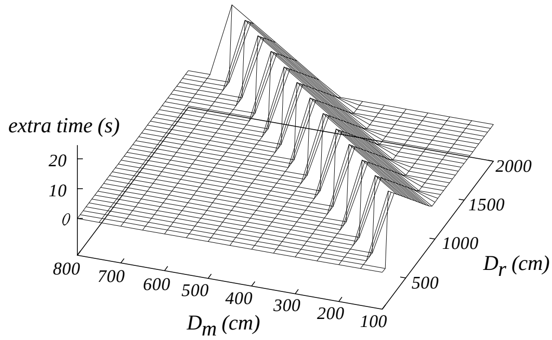

Figure 4: An example PUM.

gate's state is determined by the ith observation, which depends on the observation position; P_{true} is the actual probability of the gate being passable, which is of course not known to the robot.

Plan Utility Model

Plan utility model provides the utility value for a pair of state and action. Since the effect of the dynamics of the environment to the utility is additive, we represent a plan utility model as function $t_{ext}(D_r, D_o; v_o)$ which returns the estimated extra time imposed by the obstacle; D_r (D_o) is the distance of the robot (the obstacle) to the crossing at the time when the robot makes the commitment to the action of taking the left route. v_o is the speed of the obstacle.

Figure 4 shows an example t_{ext} for the case where the velocity of the obstacle $v_o = 10.0(cm/s)$, the velocity of the robot is $30.0(cm/s)$, and the pause time of the obstacle is $20(s)$. Once this kind of table is generated, the utility can be calculated by simply looking it up.

Simulation Results

Figure 5 shows a simulation result. The robot moved at the speed by which the robot can observe four times in the observation & decision area. After making four observations, the gate ahead became known to be passable; the robot then moved towards the goal at the maximum speed.

Figure 6 shows how the utility (or loss in this case) and the best robot action change as the dynamics of the environment changes. When the obstacle speed is low, the robot moves relatively slowly to observe the gate several times to collect enough information for decision; since enough information is collected, the resultant behavior has a low expected time (high utility). When the obstacle speed is high, the robot moves fast so that it can pass the crossing point before the obstacle reach there. The resultant behavior has a relatively hige expected time (low utility) because enough information has not been collected and, therefore, the robot sometimes takes the detour even when the gate is actually passable. The obstacle speed is much higher, the effect of obstacle becomes very small again, and

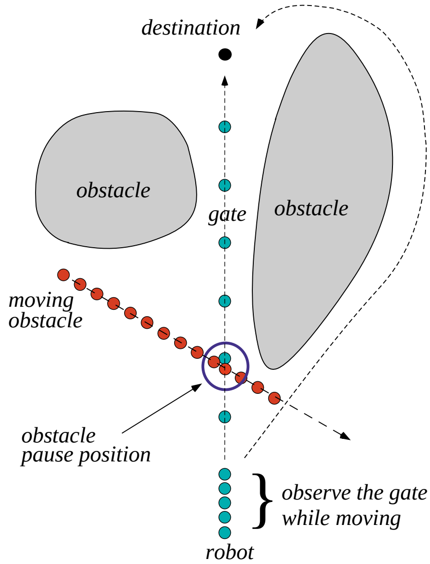

Figure 5: A simulation result. $v_m = 10\,[cm/s]$.

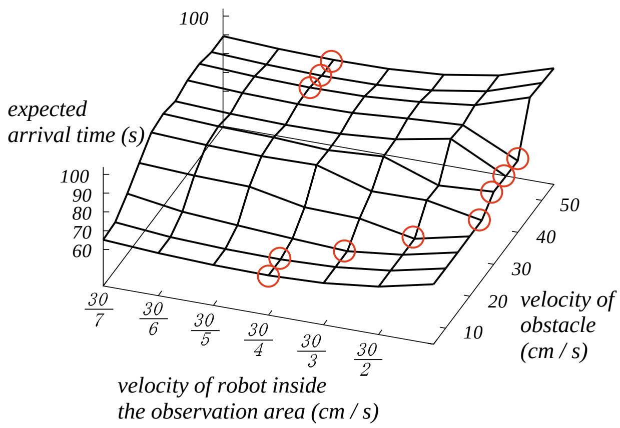

Figure 6: Simulation results for various obstacle velocity. Circles indicate the best speed of the robot for each obstacle velocity.

the robot moves slowly while observing the gate several times. The figure shows the robot can trade the increase of plan quality by obtaining more information with the decrease of plan quality by being obstructed by the moving obstacle.

Figure 6 shows the effect of the dynamics of the environment (change of obstacle velocity) is different for each action (a robot velocity). Thus, a single model (e.g., exponential decreasing model) cannot be adopted; the state-time decomposition approach is effective.

Relation with MDP-Based Approaches

Markov decision processes have been recognized as a useful tool for describing and solving planning problems under uncertainty (Dean *et al.* 1993). There exist efficient algorithms to find optimal policies for MDPs; however such *standard* algorithms has a difficulty to be applied to a large-sized MDP.

A planning problem in our domain (i.e., robot motion and observation planning in a dynamic and uncertain world) often has a large state-space. For example, we used a simplified state description on the gate's state (passable, impassable, or unknown) in the presented problem. But for the unknown cases, we can use a more elaborated description such as the estimate of the probabilistic distribution of the gate width (e.g., by a pair of the mean and the variance (Miura & Shirai 1997a)); this classifies the unknown state more precisely, thereby, enabling the robot to select better actions. We can also elaborate the description of the state of the moving obstacle. If we introduce such more expressive descriptions, the state-space will be large and it will be unrealistic to enumerate all states and to specify all state transitions.

Many techniques have been developed to reduce the complexity of solving large MDPs (e.g., (Boutilier, Dearden, & Goldszmidt 1995)); they mainly focus on obtaining optimal policies (state-action rules for every state). As Blythe (1999) has described, however, a search-based approach like ours has more interest in obtaining the best action (or action sequence) for a given state. For a *time-separable* sequential decision problem (Dean, Basye, & Lejter 1990), we can obtain the optimal policy for an initial state using dynamic programming; since this is almost the same as what is done in action selection using an AND/OR search tree (see Fig. 2), our approach is not computationally inexpensive as compared with MDP-based methods.

In real applications, many problem specific techniques have been developed to predict the future state of the environment. Our approach may have an advantage that such techniques can be straightforwardly adopted, without explicitly considering the state-space of planning problem.

Execution Monitoring using PUM

If the situation changes (unexpectedly) drastically, the current plan being executed (or the current planning process) may be no longer effective. In such a case, the agent has to terminate the current execution to initiate another execution or planning according to the new situation as soon as possible. Such an execution management is important for agents in the real world.

The plan utility model (PUM) can be a useful tool for execution management. By referring to the PUM, an agent can determine if the current plan is still effective in the coming future state, and if not, the agent can terminate it to release the computing resources to a new planning process or other activities. We are now investigating such a use of PUM in planning in dynamic environments. Hansen and Zilberstein (1996) proposes to use time-dependent utility for decision-theoretic control of execution monitoring.

Concluding Remarks

This paper has presented an approach to estimating time-dependent utility used for decision-theoretic planning in dynamic environments. The approach is based on the *state-time decomposition*, based on which the time-dependent utility is estimated from both the time-dependent state (state evolution model) and the state-dependent utility (plan utility model). This approach is expected to provide both a clear guideline for designing the time-dependent utility and an efficient utility estimation. We have applied the approach to a mobile robot navigation problem in a dynamic environment. We also discussed the relation between the proposed approach and the MDP-based approaches.

References

Blythe, J. 1999. Decision-theoretic planning. *AI Magazine* 20(2):37–54.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In *Proceedings of the Fourteenth Int. Joint Conf. on Artificial Intelligence*, 1104–1111.

Cameron, A., and Durrant-Whyte, H. 1990. A bayesian approach to optimal sensor placement. *Int. J. of Robotics Res.* 9:70–88.

Dean, T.; Basye, T.; and Lejter, S. 1990. Planning and active perception. In *Proceedings of 1990 DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, 271–276.

Dean, T.; Kaelbling, L.; Kirman, J.; and Nicholson, A. 1993. Planning with deadlines in stochastic domain. In *Proceedings of AAAI-93*, 574–579.

Hansen, E., and Zilberstein, S. 1996. Monitoring the progress of anytime problem-solving. In *Proceedings of AAAI-96*.

Horvitz, E., and Rutledge, G. 1991. Time-dependent utility and action under uncertainty. In *Proceedings of the 7th Conf. on Uncertainty in Artificial Intelligence*, 151–158.

Hutchinson, S., and Kak, A. 1989. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Trans. on Robotics and Automat.* RA-5(6):765–783.

Miura, J., and Shirai, Y. 1997a. Vision and motion planning for a mobile robot under uncertainty. *Int. J. of Robotics Research* 16(6):806–825.

Miura, J., and Shirai, Y. 1997b. Vision-motion planning for a mobile robot considering vision uncertainty and planning cost. In *Proceedings of the 15th Int. Joint Conf. on Artificial Intelligence*, 1194–1200.

Miura, J., and Shirai, Y. 2000. Modeling motion uncertainty of moving obstacles for robot motion planning. In *Proceedings of the 2000 IEEE Int. Conf. on Robotics and Automation.* (to appear).

Russell, S., and Wefald, E. 1991. *Do The Right Thing.* The MIT Press.

Zilberstein, S. 1993. *Operational Rationality through Compilation of Anytime Algorithm.* Ph.D. Dissertation, University of California at Berkeley.