

Using Goals to Find Plans with High Expected Utility

Jak Kirman, Ann Nicholson,
Moises Lejter, Thomas Dean
Dept. of Computer Science
Brown University,
Box 1910, Providence,
RI 02912

Eugene Santos Jr.
Department of Electrical and
Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB,
OH 45433-7765

Abstract. We describe a method for planning to achieve goals under uncertainty that makes use of decision-theoretic methods to guide search. Given a probabilistic model of the world and a utility measure on world states, we wish to find plans (sequences of actions) with high expected utility. Finding a plan maximizing the expected utility is combinatorial in nature. In previous research, we coped with the combinatorics by making simplifying assumptions that sometimes led to a poor choice of plan. In this paper, we reduce the combinatorics by restricting attention to plans that are likely to achieve specific goals; we then use a successive approximation algorithm to select from these plans one with high utility. We obtain the restricted set of plans using a procedure that can, given a goal, generate candidate plans one at a time as needed; these plans are produced in decreasing order of probability of achieving their goal. This procedure is also used to obtain iteratively refined bounds on the expected utility of the candidate plans; these bounds are used in choosing a plan to be executed, hence our method produces better plans the more time it is given. We apply this method to a robotics problem addressed in our previous research and show that the goal-directed search produces better plans.

1. Introduction

Our research has focused on the problems involved in planning in uncertain domains. We view planning in terms of enumerating a set of possible sequences of actions, or plans, evaluating the outcomes of those plans to determine their expected utilities, and selecting the plan with the highest expected utility. We use Bayesian decision theory [14] as a basis for planning under uncertainty. We model the world with temporal Bayesian networks, a compact and well-understood formalism which allows us to reason about the effects of actions over time [4, 12].

We describe a method for goal-directed search in planning within such a decision-theoretic framework. Given a probabilistic model of the world and a utility measure on world states, the aim is to choose a plan with high expected utility. We guide search

by considering plans that are likely to lead to outcomes satisfying particular goals. However, the final selection of a plan is made on the basis of the expected utility of plans, rather than simply choosing the plan with the highest probability of achieving a goal, because such a plan could also lead to undesirable outcomes with some probability, thus resulting in a low expected utility.

Finding the plan with the maximum expected utility is combinatorial in nature, since we must consider all possible plans; therefore an exact solution is impractical. In our previous research [13, 1], the complexity problem was handled by making simplifying assumptions about the world; these assumptions made it feasible to consider all possible plans but sometimes led to a poor choice of plan.

In this paper, we instead direct the search through the space of plans by first restricting attention to plans that are most likely to lead to the outcome with the highest utility. If all these plans have low probability of reaching this best outcome, we consider outcomes with progressively lower utility.

In order to generate these plans we use a procedure that, given an outcome (a future world state) provides the most likely plan and intermediate world states that lead to this future state, together with the probability of this combination. (We use the term *scenario* for a plan and an outcome, together with intermediate world states; we define a scenario formally in Section 3.1.) Upon demand, the procedure iteratively provides scenarios matching this outcome in decreasing order of probability. The actual procedure we use is an application of an algorithm for Bayesian belief revision based on linear constraint satisfaction [15, 16]. This procedure can be used either to generate explanations, as above, or predictively, by taking a plan and producing the most likely scenario (including an outcome) and its probability. Again, when used predictively, the procedure generates a sequence of these scenario/probability pairs in order of decreasing probability. With such a sequence, we can iteratively refine the bounds on the expected utility of the candidate plans. Thus we further reduce the complexity of choosing a plan by avoiding the exact computation of the expected utility of any plan; furthermore this process of iterative refinement of bounds on the expected utilities allows us to choose a plan at any point during computation at which circumstances demand a response from the system.

We apply these techniques to the robot tracking example used in our previous research and show that the goal-directed search produces better plans. The domain of mobile robotics is particularly appropriate for planning research as it requires considerable complexity in terms of temporal and spatial representation, and involves time pressure and uncertainty in sensing and action.

2. Background

A *Bayesian network* is a directed acyclic graph, where vertices correspond to random variables, often referred to as *chance nodes*. The relationship between any set of state variables can be specified by a joint probability distribution. The arcs in the network define the causal and informational dependencies between the random variables. In the model described in this paper, chance nodes are discrete-valued variables that encode

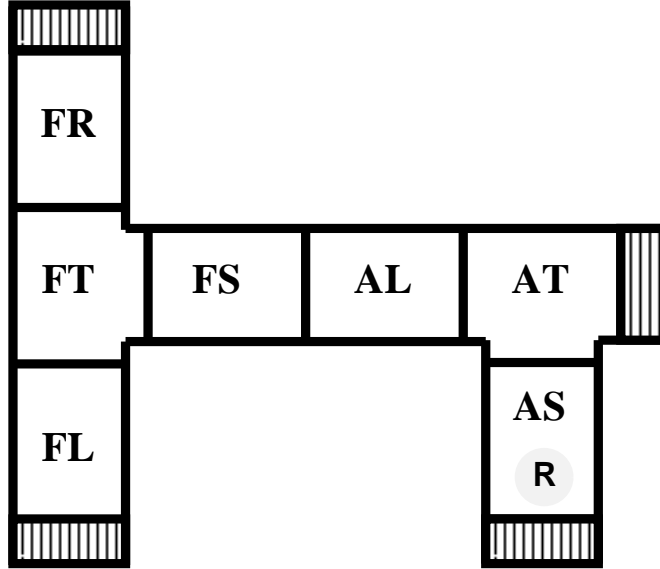


Figure 1: Global map of the environment supplied to the robot.

states of knowledge about the world. There is a probability distribution for each chance node: if the node has no predecessors then this is its marginal probability distribution; otherwise, it is a conditional probability distribution dependent on the states of its immediate predecessors. Evidence can be specified about the state of any node in the network; this evidence is propagated through the network affecting the overall joint distribution, and producing posterior probability distributions for each node. In this paper we use a specialization of Bayesian networks, a model called *temporal belief networks* [4].

We apply our method to the same domain used in our previous work, that of Mobile Target Localization (MTL), with a mobile robot navigating and tracking a moving target in a cluttered environment. The robot’s task is to detect and track moving objects, reporting their location in the coordinate system of a global map (see Figure 1). The robot is provided with sonar and visual sensory capabilities, and supplied with a global map. The robot must balance the often competing aims of tracking and position estimation, in order not to become lost.

Our decision model for the MTL problem is represented in a network model as follows. The world is tessellated into a set of regions \mathcal{L} , based on the sensor capabilities, representing the possible locations of both the robot and its target. Let L_R and L_T be variables representing the possible locations of the robot and the target respectively. Let A_R be a variable representing the action taken by the robot. At any given point in time, the robot can make certain observations regarding its position with respect to the global map, and the target’s position with respect to the robot, represented by O_R and O_T respectively. Our temporal network model uses a discrete time representation, with a chance node for each variable at each time point. Figure 2 shows the temporal belief network for the MTL model over 4 time slices. (See [13] for a more detailed description of the original MTL model.) We have modified the model slightly to prevent the robot from attempting actions that are physically impossible (such as moving forward when facing a wall); these constraints are represented by the arc from L_R to A_R . A

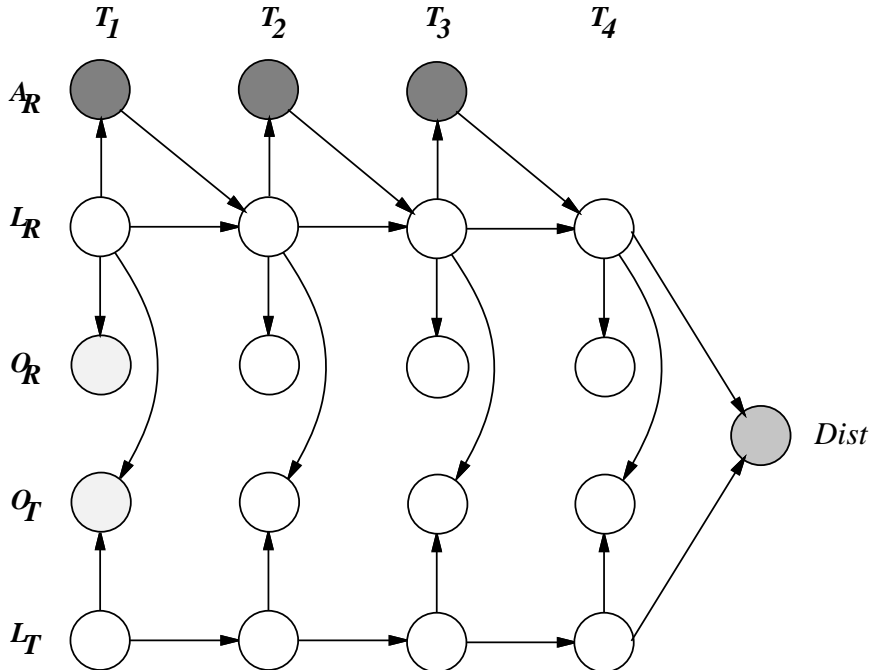


Figure 2: Probabilistic model for the MTL problem

further addition to the original model is the variable $Dist$, which represents the distance between the robot and the target, and allows us to use proximity to the target as a desirable outcome. The lighter shaded O_R and O_T nodes correspond to observation evidence. In the algorithm we describe in this paper, the medium shaded nodes will be instantiated to desirable outcomes, in order to generate the restricted set of candidate plans. The darkest shaded A_R action nodes will be instantiated by our algorithm in the process of refining the bounds of the expected utility of a plan.

The method presented in this paper for guiding the search for a plan with high expected utility relies on a procedure developed by Santos [15, 16] to perform belief revision in Bayesian networks, based on linear constraint satisfaction. A system of linear constraints is generated automatically to represent a given Bayesian network. A cost function is defined such that minimizing this function corresponds to finding the most probable assignment to the variables in the original network. Techniques from operations research such as linear programming can be used to perform this minimization.

3. The Approach

3.1. Notation

Let V be a random variable and let Ω_V be the finite set of possible values for V (called the *domain* of V). Given a finite collection of random variables $\mathcal{V} = \{V_1, V_2, \dots, V_n\}$, we define the *assignment space* of \mathcal{V} as

$$\Omega_{\mathcal{V}} = (\Omega_{V_1} \cup \{\perp\}) \times (\Omega_{V_2} \cup \{\perp\}) \times \dots \times (\Omega_{V_n} \cup \{\perp\}) \quad (1)$$

where \perp is distinct from the Ω_{V_i} s and denotes “not assigned” or “don’t care”. Each element ω in $\Omega_{\mathcal{V}}$ is called an *assignment* on \mathcal{V} . We can represent ω as a vector $[v_1, v_2, \dots, v_n]$ corresponding to the cross-product where $v_i \in \Omega_{V_i} \cup \{\perp\}$.

A *complete assignment space* for \mathcal{V} is a subset of $\Omega_{\mathcal{V}}$ defined as

$$\Omega_{\mathcal{V}}^c = \Omega_{V_1} \times \Omega_{V_2} \times \dots \times \Omega_{V_n}. \quad (2)$$

Hence, assignments ω in $\Omega_{\mathcal{V}}^c$ are called *complete assignments* (also called *scenarios*) since all random variables are given a value. We denote complete assignments for \mathcal{V} by $\omega_{\mathcal{V}}^c$ or simply ω^c when the random variable set is understood.

Given our notion of complete assignments, we consider any assignments found in $\Omega_{\mathcal{V}}$ to be *partial assignments*. They are denoted by $\omega_{\mathcal{V}}^p$ or ω^p .

Given assignments $\omega^c = [v_1, \dots, v_n]$ and $\omega^p = [v'_1, \dots, v'_n]$ for \mathcal{V} , we say that ω^c *agrees with* ω^p if for all $i = 1, \dots, n$, either $v_i = v'_i$ or $v'_i = \perp$. We also call ω^c an *extension* of ω^p .

A *temporal network* TN is a 4-tuple $(\mathcal{V}, p, \mathcal{O}, U)$ where

- \mathcal{V} is a collection of random variables.
- p is a probability function on $\Omega_{\mathcal{V}}^c$.
- \mathcal{O} is a subset of \mathcal{V} called the *outcome variables*.
- U is a *utility function* from $\Omega_{\mathcal{O}}^c$ to \mathfrak{R} . Since only the outcomes are used in determining utility, we define the utility of a scenario to be the utility of the values of its outcome nodes. Hence, we can extend our function to $U(\omega_{\mathcal{V}}^c) = U(\omega_{\mathcal{O}}^c)$ where $\omega_{\mathcal{V}}^c$ agrees with $\omega_{\mathcal{O}}^c$ by simply extending $\omega_{\mathcal{O}}^c$ to a partial assignment for \mathcal{V} . The maximum and minimum utility bounds are given by

$$U_{min} = \min_{o \in \Omega_{\mathcal{O}}^c} U(o), \quad U_{max} = \max_{o \in \Omega_{\mathcal{O}}^c} U(o)$$

The *scenario set* for a partial assignment $\omega^p \in \Omega_{\mathcal{V}}$, denoted $S_{\mathcal{V}}(\omega^p)$, is the set of all complete assignments in $\Omega_{\mathcal{V}}^c$ that agree with ω^p :

$$S_{\mathcal{V}}(\omega^p) = \{\omega^c \in \Omega_{\mathcal{V}}^c | \omega^c \text{ agrees with } \omega^p\}$$

In essence, we are simply extending the partial assignment to all possible complete assignments by replacing the \perp s. Furthermore, the probability of a partial assignment is simply the sum of the probabilities of the scenarios in its scenario set:

$$p(\omega^p) = \sum_{\omega^c \in S_{\mathcal{V}}(\omega^p)} p(\omega^c)$$

Let \mathcal{A} be a set of random variables from \mathcal{V} called the *action variables*. A complete assignment to \mathcal{A} , denoted $\pi \equiv \omega_{\mathcal{A}}^c$, is called a *plan*. Given action variables \mathcal{A} , we define the *expected utility* EU on a plan π to be the mapping from $\Omega_{\mathcal{A}}^c$ to \mathfrak{R} where

$$EU(\pi) = \sum_{\omega^c \in S_{\mathcal{V}}(\pi)} U(\omega^c) p(\omega^c | \pi) = \sum_{\omega^c \in S_{\mathcal{V}}(\pi)} U(\omega^c) \frac{p(\omega^c)}{p(\pi)}$$

The *scenario sequence*, $\text{SSEQ}_{\mathcal{V}}(\omega^p)$, for a partial assignment ω^p on \mathcal{V} is a sequence of ordered pairs, each pair consisting of a scenario and its probability, and the sequence is ordered by the probabilities. Formally,

$$\text{SSEQ}_{\mathcal{V}}(\omega^p) = \{(\omega_n^c, p(\omega_n^c))\}_{n=1}^N$$

is a sequence where $N = |S_{\mathcal{V}}(\omega^p)|$, $\omega_n^c \in S_{\mathcal{V}}(\omega^p)$, and $p(\omega_n^c) \geq p(\omega_{n+1}^c)$ for $n = 1, \dots, N-1$.

(From this point on, our choice of \mathcal{V} is fixed and information involving scenario sets and sequences are solely based on \mathcal{V} . Hence, we will drop the subscripts to S and SSEQ .)

3.2. Method

Given \mathcal{A} , our goal is to choose a sequence of actions, or plan, $\pi^* \in \Omega_{\mathcal{A}}^c$ with high expected utility. Computing the expected utility of action sets is combinatorial in nature; therefore a correct solution is impractical. Our approach reduces the complexity of the decision procedure in two ways. First, we consider only a limited number of plans, $\mathcal{P} \subseteq \Omega_{\mathcal{A}}^c$. Second, for each plan under consideration, instead of computing its exact expected utility, we compute bounds on the expected utility of that plan.

We begin by considering the outcome o_{max} in $\Omega_{\mathcal{O}}^c$ with the highest utility, i.e. $U(o_{max}) = U_{max}$. We then start generating the plans from the scenario sequence $\text{SSEQ}(o_{max})$ (where each scenario includes a plan). Depending on the bounds computation, which we describe next, we may later include plans for outcomes with successively lower utilities.

Once we have a set of candidate plans, \mathcal{P} , we iteratively refine the upper and lower bounds on the expected utility of every plan $\pi \in \mathcal{P}$. Initially, we have no information about the expected utilities, so the bounds are simply the maximum and minimum utilities:

$$\begin{aligned} \text{EU}_{min}^0(\pi) &= \sum_{\omega^c \in S(\pi)} U_{min} \frac{p(\omega^c)}{p(\pi)} = U_{min} \\ \text{EU}_{max}^0(\pi) &= \sum_{\omega^c \in S(\pi)} U_{max} \frac{p(\omega^c)}{p(\pi)} = U_{max} \end{aligned}$$

Consider the set of all scenario probability pairs in the scenario sequences for the plans being considered, \mathcal{P} :

$$Sseq = \bigcup_{\pi \in \mathcal{P}} \text{SSEQ}(\pi)$$

We generate pairs $(\omega^c, p(\omega^c))$ from $Sseq$ and use them to update the boundaries as described by equations (3) and (4):

$$\text{EU}_{min}^{i+1}(\pi) = \begin{cases} \text{EU}_{min}^i(\pi) + \frac{p(\omega^c)}{p(\pi)}[U(\omega^c) - U_{min}] & \text{if } \omega^c \text{ agrees with } \pi \\ \text{EU}_{min}^i(\pi) & \text{otherwise} \end{cases} \quad (3)$$

$$\text{EU}_{max}^{i+1}(\pi) = \begin{cases} \text{EU}_{max}^i(\pi) + \frac{p(\omega^c)}{p(\pi)}[U(\omega^c) - U_{max}] & \text{if } \omega^c \text{ agrees with } \pi \\ \text{EU}_{max}^i(\pi) & \text{otherwise} \end{cases} \quad (4)$$

Note that every scenario generated from $Sseq$ affects exactly one expected value interval since

$$\forall \pi_1, \pi_2 \in \mathcal{P}, \pi_1 \neq \pi_2 \Rightarrow SSEQ(\pi_1) \cap SSEQ(\pi_2) = \emptyset$$

Also note that the scenario and probability pairs generated for each plan in the first stage, allows us to refine the corresponding plan's expected utility bounds as described above.

It is easy to show that $EU_{max}^i(\pi) \geq EU_{min}^i(\pi)$ and furthermore that, in the limit, the expected utility bounds for each plan converge to its actual expected utility.

$$\lim_{i \rightarrow N} EU_{max}^i(\pi) = \lim_{i \rightarrow N} EU_{min}^i(\pi) = \sum_{\omega^c \in S(\pi)} U(\omega^c) \frac{p(\omega^c)}{p(\pi)} = EU(\pi)$$

When we are required to choose a plan, there are several possible strategies. We select the plan whose expected utility interval has the highest midpoint, which is optimal if the actual expected utilities of the plans are equally likely to fall anywhere within the interval. If avoiding bad outcomes is an important concern, a better strategy is to choose the plan with the maximum EU_{min} , i.e. the greatest lower bound.

3.3. Guiding the Search

We described above an incremental algorithm to select a plan with highest expected utility. In the limit, that is, if we use that algorithm to consider all possible plans and all possible outcomes for each plan, we will obtain a correct selection. However, such a strategy would be impractical, due to the combinatorics involved. The description above suggests a strategy that would alleviate this problem: constraining the search only to plans likely to lead to desirable outcomes. In this section we describe a more powerful strategy to guide that search, by additionally providing a mechanism that allows us to choose when to consider new plans, and which plans to concentrate our efforts on, in terms of refining its expected utility bounds.

In the algorithm described above, at any given time we have two possibilities. The first is to refine the bounds on a plan π in \mathcal{P} , by producing the next scenario probability pair for π , $(\omega_{n+1}^c, p(\omega_{n+1}^c))$, where ω_{n+1}^c agrees with π . The second is to add another candidate plan to \mathcal{P} , by producing the next most probable scenario for some desirable outcome. We present a heuristic procedure for making this choice, using a metric on the state of the decision process which intuitively corresponds to the amount of discrimination between the expected utility intervals for the plans. At each stage in our algorithm, we estimate the value of the metric after each of the possible next steps (refine a plan $\pi_i \in \mathcal{P}$, or generate a new candidate), and choose the step with the highest estimate.

At any given time i , the procedure has a set IS_i of intervals bounding the utility of the plans in the candidate set,

$$IS_i = \{[EU_{min}^i(\pi), EU_{max}^i(\pi)] \mid \pi \in \mathcal{P}\}$$

Let $CP(IS_i)$ be the plan we choose given IS_i . If our selection criterion is the plan whose expected utility interval has the highest midpoint, then

$$EU_{min}(CP(IS_i)) + EU_{max}(CP(IS_i)) \geq EU_{min}(\pi) + EU_{max}(\pi)$$

for all $\pi \in \mathcal{P}$.

We define the function M on IS_i to be the difference between the expected utility of the plan we choose and the highest expected utility of the candidate plans.

$$M(IS_i) = EU(CP(IS_i)) - \max_{\pi \in \mathcal{P}} EU(\pi)$$

M is a measure of the usefulness of the set of intervals in making a choice of plan. Since the values of $EU(\pi)$ are not known, we cannot compute M directly. However, we can estimate it by assuming that the true expected utility for each plan π is equally likely to take any value in the interval $[EU_{min}(\pi), EU_{max}(\pi)]$. For $\mathcal{P} = \{\pi_1, \dots, \pi_n\}$, suppose π_j is the plan chosen by our criterion $CP(IS_i)$. Our measure estimate $\hat{M}(IS_i)$ is given by:

$$\hat{M}(IS_i) = \frac{\int_{EU_{min}(\pi_1)}^{EU_{max}(\pi_1)} \int_{EU_{min}(\pi_2)}^{EU_{max}(\pi_2)} \dots \int_{EU_{min}(\pi_n)}^{EU_{max}(\pi_n)} (e_j - \max(e_1, \dots, e_n)) de_n \dots de_1}{\prod_{k=1}^n (EU_{max}(\pi_k) - EU_{min}(\pi_k))}$$

For each of the plans π in \mathcal{P} , we estimate the $M(IS_{i+1})$ corresponding to refining the set of bounds for π . Equations (3) and (4) describe how to compute this estimated value; to use them, we must obtain estimates on the probability and utility of the next scenario that would be generated for each plan π . We use order statistics on the decreasing sequence of scenario probabilities previously generated for π to obtain an estimate for the probability of its next scenario, $\hat{p}(\omega_n^c + 1)$. We use the mean of the utility of scenarios previously generated for π to obtain an estimate $\hat{U}(\omega_n^c + 1)$ for the utility of this next scenario. Once we obtain these, computing estimated refined interval bounds $\hat{EU}_{min}^{i+1}(\pi)$ and $\hat{EU}_{max}^{i+1}(\pi)$ for plan π is straightforward. The same technique is used to compute $\hat{M}(IS_{i+1})$ corresponding to adding a new plan to \mathcal{P} .

3.4. Performance Considerations

The dominant performance consideration involves the scenario generation procedure. This procedure is based on a linear constraint satisfaction approach using integer linear programming techniques. It is well known that its core of linear programming can be solved in roughly quadratic time with respect to the number of constraints and variables involved. The size of the constraint systems generated for Bayesian networks by the algorithm we rely on is linear with respect to the size of the network's conditional tables. In addition, the sparseness of those tables can contribute a great deal to further reduce the size of the constraint system. Techniques such as the Simplex method and Karmarkar's projective scaling algorithm are designed to solve linear programming problems which contain tens of thousands of variables and constraints. Our class of problems generate constraint systems well within the capabilities of these techniques. In addition, there are many other possible optimizations based on the knowledge of the domain, such as the rigid effects from clamping any single random variable, which reduce the size of the resulting constraint systems.

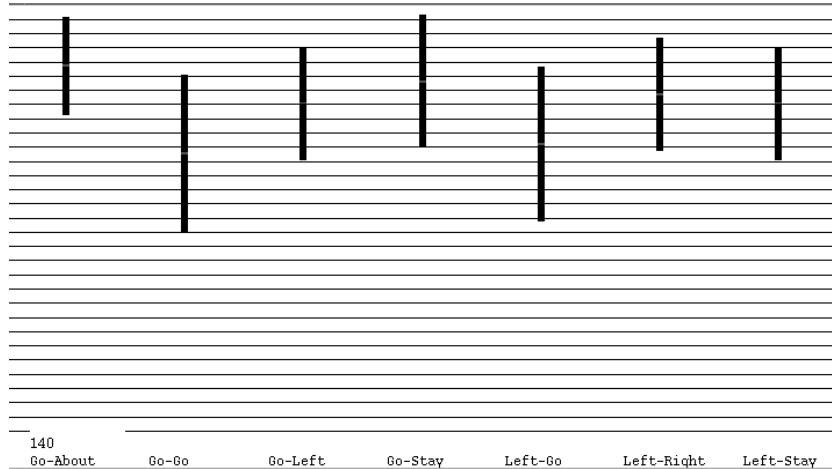


Figure 3: Intervals after first set of scenarios

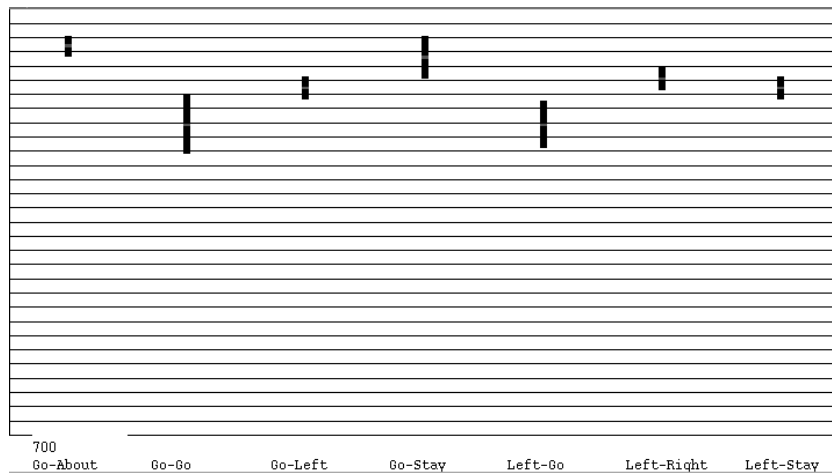


Figure 4: Intervals after further refinement

4. Results for the MTL Model

Our experimental results are given for the MTL problem over 3 time slices, with outcome nodes, $\mathcal{O} = \{Dist, O_T(T_3)\}$. The utility function is proportional to the negation of the distance to the target, slightly modified by the observations at the last time step, i.e. “better” observations are preferred.

Experimental results for the interval refinement are shown in Figures 3, 4, and 5. The initial position of the robot is known (in the corner, location **AT**, facing west), and the target is likely to be directly in front of the robot in the corridor (location **AL**), with a tendency to be stationary. All three figures show the expected utility intervals for the set of 7 candidate plans. The utility is shown on the Y-axis, mapped onto the interval 0 to 1. Each of the candidate plans consists of two actions, $A_R(T_0)$ and $A_R(T_1)$, where $A_R(T_i) \in \{\mathbf{Go}, \mathbf{About}, \mathbf{Left}, \mathbf{Right}, \mathbf{Stay}\}$; the candidate plans are shown as **action1-action2**, below the X-axis. Figure 3 shows the expected utility intervals after considering the first 20 scenarios for each candidate plan, Figure 4 after 100 scenarios for each, and Figure 5 shows the actual expected utility. (The total

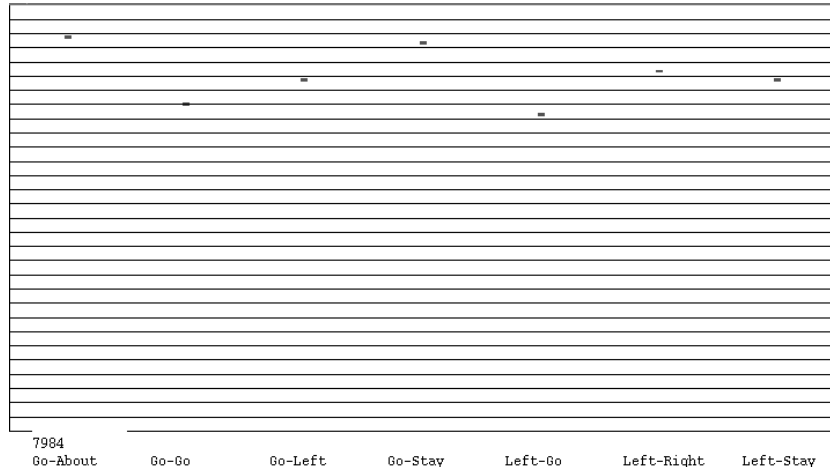


Figure 5: Actual expected utilities

number of scenarios across all candidate plans for each figure is shown in the bottom left corner.)

For the MTL problem, a good plan is clearly one which would put the robot in the same region as the target. The plans with high expected utilities which would be chosen by our system for the situation described above all display this feature. In particular, the top 2 candidate plans begin with first action **Go**, which would have the effect of moving the robot to location **AL**. In contrast, when we applied our earlier techniques to this example, the simplifying assumptions made about the world resulted in the robot displaying a distinct disinclination to move until the target was further away.

In this example, the rate of interval refinement is similar because we use the same number of scenarios for each and also because they have similar probabilities and utilities. (The latter is largely a function of the domain.) When we apply the heuristics for selectively refining the intervals described in this section, the rates of refinement vary significantly, concentrating on achieving greater discrimination among intervals sooner.

5. Related Work

In earlier work [13] we used essentially the same model. The decision process then was correct in the sense that we examined all possible plans, and for each plan, examined all possible outcomes of the plan. For each outcome, we estimated the probability of the outcome, and used this value to weight the utility of the outcome. Summing over outcomes provided us with a measure of the expected utility of the plan. We made several approximations to make the computations feasible in a reasonable amount of time. First, the model of the world was simplified to reduce the number of states each variable could take. Second, we approximated the probability of an outcome given a plan by assuming that future observations were independent of each other. Third, when determining the probability of the outcomes from a plan, we discounted the intermediate observations. Finally, instead of computing a complete policy (which would allow the system to condition subsequent actions on intermediate observations) we computed the expected utility of the plans assuming that they would be executed unconditionally.

Although these approximations were sufficiently accurate to produce reasonable results in many cases, we found the system would sometimes select poor plans.

In the method described in this paper, we use the same approach to simplifying the model of the world, but we no longer make the independence assumptions described above, since at each step in the algorithm we are considering a complete assignment, along with its true probability. Thus the bounds on the expected utility of the action sequences are correct with respect to the world model and the utility measure. Note, however, that the heuristics we describe here to choose a plan from several whose expected utility intervals overlap do not allow us to make assertions about the correctness of the choice, though we can provide lower bounds on the expected utility of the chosen action. Finally, we no longer consider actions as if they were to be executed without consideration of intermediate observations; we now encode information in our model of the world to represent the fact that the choice of action can depend on the current observations. This is still not as powerful as a complete policy, which would allow us to condition actions on the predictions made by the decision model, but provides a substantial improvement over previous models.

Satisficing approaches [19] and goal-directed search methods provide alternatives to the notion of maximizing utility dominant in decision theory. However, actually defining what constitutes a goal or a satisficing solution to a given planning problem requires taking into account an agent's preferences over outcomes [6]. Of late there have appeared a number of proposals for reconciling goal-directed and utility-directed decision making strategies [5, 9, 20]. This paper provides a particular method for using goals to direct search while at the same time using expected utility to select among plans that have been determined to achieve goals with some probability.

At each point in execution, our method chooses a plan, executes the first step in that plan, throws away the plan, and then starts all over again. This is in contrast with methods that compute a policy [18] or a highly conditional plan [11, 17] that need only be computed once and will serve for any situation in which the agent finds itself. We are primarily concerned with choosing a good plan under time pressure [3]. Our method is similar in its motivation to that of Drummond and Bressina [8] in that it addresses the issues of uncertainty and time-criticality in planning. It is different from that work in that it adopts a more precise model for prediction in the form of Markov and semi-Markov processes [7, 12] and in that it recomputes a new plan at each step. Our use of intervals to cope with time-criticality in uncertain reasoning is in keeping with the work of Horvitz et al. [10] on bounding probabilities in updating Bayesian networks and Breese and Fertig [2] on propagating intervals in evaluating influence diagrams. Our use of intervals to bound expected utility estimates is purely a concession to complexity.

6. Conclusion

Decision-theoretical models provide a convenient and well-understood framework for modeling the world and representing the effects of actions over time. However a major problem with this approach is the combinatorial nature of finding the plan with the maximum expected utility, which manifests itself in two ways: (1) considering all possi-

ble plans; and (2) computing the exact expected utility for each plan. In this paper we have presented a method for finding plans with high expected utility, which addresses both these aspects, by using goal-directed search to consider only plans with desirable outcomes and by iteratively refining bounds on the expected utility of those plans. We have successfully applied this method to a problem in the robotics domain previously explored using other techniques.

Acknowledgements

This work was supported in part by a National Science Foundation Presidential Young Investigator Award IRI-8957601, by the Air Force and the Advanced Research Projects Agency of the Department of Defense under Contract No. F30602-91-C-0041, and by the National Science foundation in conjunction with the Advanced Research Projects Agency of the Department of Defense under Contract No. IRI-8905436.

References

- [1] Kenneth Basye, Thomas Dean, Jak Kirman, and Moises Lejter. A decision-theoretic approach to planning, perception, and control. *IEEE Expert*, 7(4):58–65, 1992.
- [2] John S. Breese and Kenneth W. Fertig. Decision making with interval influence diagrams. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 122–129, July 1990.
- [3] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proceedings AAAI-88*, pages 49–54. AAAI, 1988.
- [4] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.
- [5] Thomas Dean and Michael Wellman. On the value of goals. In Josh Tenenber, Jay Weber, and James Allen, editors, *Proceedings from the Rochester Planning Workshop: From Formal Systems to Practical Systems*, pages 129–140, 1989.
- [6] Thomas Dean and Michael Wellman. *Planning and Control*. Morgan Kaufmann, San Mateo, California, 1991.
- [7] Thomas L. Dean, R. James Firby, and David P. Miller. Hierarchical planning involving deadlines, travel time and resources. *Computational Intelligence*, 4(4):381–398, 1988.
- [8] Mark Drummond and John Bresina. Anytime synthetic projection: Maximizing the probability of goal satisfaction. In *Proceedings AAAI-90*, pages 138–144. AAAI, 1990.
- [9] Peter Haddawy and Steve Hanks. Issues in decision-theoretic planning: Symbolic goals and numeric utilities. In *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pages 48–58. DARPA, 1990.
- [10] Eric J. Horvitz, H. Jacques Suermondt, and Gregory F. Cooper. Bounded conditioning: Flexible inference for decisions under scarce resources. In *UW89*, pages 182–193, 1989.
- [11] Leslie Pack Kaelbling. Goals as parallel program specifications. In *Proceedings AAAI-88*, pages 60–65. AAAI, 1988.
- [12] Keiji Kanazawa. *Reasoning about Time and Probability*. PhD thesis, Brown University, Providence, RI, 1991.

- [13] Jak Kirman, Kenneth Basye, and Thomas Dean. Sensor abstractions for control of navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2812–2817, 1991.
- [14] Howard Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard University Press, 1961.
- [15] Eugene Santos, Jr. On the generation of alternative explanations with implications for belief revision. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, 1991.
- [16] Eugene Santos, Jr. *A Linear Constraint Satisfaction Approach for Abductive Reasoning*. PhD thesis, Department of Computer Science, Brown University, 1992.
- [17] Marcel J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings IJCAI 10*, pages 1039–1046. IJCAII, 1987.
- [18] Ross D. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.
- [19] Herbert A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99–118, 1955.
- [20] Michael P. Wellman and Jon Doyle. Preferential semantics for goals. In *Proceedings AAAI-91*. AAAI, 1991.