Computational Intelligence, Volume 14, Number 3, 1998

UTILITY MODELS FOR GOAL-DIRECTED, DECISION-THEORETIC PLANNERS

PETER HADDAWY

Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee

STEVE HANKS

Department of Computer Science and Engineering, University of Washington

AI planning agents are *goal-directed*: success is measured in terms of whether an input goal is satisfied. The goal gives structure to the planning problem, and planning representations and algorithms have been designed to exploit that structure. Strict goal satisfaction may be an unacceptably restrictive measure of good behavior, however.

A general *decision-theoretic* agent, on the other hand, has no explicit goals: success is measured in terms of an arbitrary preference model or utility function defined over plan outcomes. Although it is a very general and powerful model of problem solving, decision-theoretic choice lacks structure, which can make it difficult to develop effective plan-generation algorithms.

This paper establishes a middle ground between the two models. We extend the traditional AI goal model in several directions: allowing goals with temporal extent, expressing preferences over partial satisfaction of goals, and balancing goal satisfaction against the cost of the resources consumed in service of the goals. In doing so we provide a utility model for a goal-directed agent.

An important quality of the proposed model is its tractability. We claim that our model, like classical goal models, makes problem structure explicit. This structure can then be exploited by a problem-solving algorithm. We support this claim by reporting on two implemented planning systems that adopt and exploit our model.

Key words: planning, decision theory, utility, goals, preferences.

1. INTRODUCTION

Reasoning about and planning for an uncertain world raises both representational and algorithmic problems. We need to represent change, uncertainty, and value or utility in order to represent alternative plans of action. We further need an efficient way to generate plausible plans, anticipate their results, improve their performance, and choose the best option.

Classical AI planning algorithms provide a vocabulary and computational model for describing and solving planning problems. The definition of a planning problem includes:

1. a description of some initial *world state* (usually expressed as a formula in some logical or quasi-logical language)
2. a *goal state description* (another formula that describes a subset of the world states, one of which should hold at the end of plan execution)
3. a set of *operators* (that can effect change in the world state).

Classical planning algorithms also typically assume an omniscient agent: the initial world state is known with certainty, the operators' state changes are deterministic, and the world does not change except as a result of operator execution. As a result, the agent always has complete information about the world state while it is planning and executing.

We are particularly interested in the second component of this problem definition. The goal state description defines a set of world states that should hold when plan execution completes. Any plan that moves the world into one of these states is equally good. We will refer to this definition of the planning problem as the "simple goal satisfaction" model.[1]

Address correspondence to Peter Haddawy at the Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, Milwaukee, WI, 53201; e-mail: haddowy@cs.uwm.edu.

[1] A vast majority of planning algorithms almost unanimously define the problem in terms of simple goal satisfaction. To

Simple goal satisfaction is an attractive way to define the planning problem. In addition to being concise, the model derives power from a goal formula that makes the target world state *explicit* to the planning algorithm. The goal formula can be matched against operator postconditions, for example, to determine what operators should be added to the plan. This technique is fundamental to a diverse collection of planning algorithms, including goal regression, means-ends analysis, and backward- and forward-chaining planners.

But simple goal satisfaction has these limitations as a model:

- It defines success in terms of what holds only at the *end* of plan execution.
- Goals are regarded as all-or-nothing propositions, so failure to achieve the goal completely is equivalent to complete failure.
- It regards as equivalent any two plans that achieve the goal, without regard to how efficiently they do so.

Furthermore, while simple goal satisfaction may be a good model for agents required to achieve their goals aggressively and at all costs, it prevents us from representing situations like the following:

- Achieve the goal by noon; being slightly late is acceptable but not ideal.
- Bring me these three books from the library; if one is checked out, bring the other two anyway.
- Run these errands, but be careful to conserve fuel because it is very expensive.

None of these situations is inherently excluded from the classical planning paradigm, but few implemented algorithms can handle them.

What are the alternatives if the simple goal satisfaction model is inadequate for a particular problem domain? The decision-theoretic model of problem solving (e.g., Raiffa 1968) defines success in terms of *preferences* over the *outcomes* of executing a plan. This model permits the expression of preferences between any two outcomes instead of a binary satisfied/not satisfied criterion applied to the world state prevailing at the end of execution. The focus on complete outcomes (rather than on the end of execution only) permits the expression of preferences about any aspect of a plan's execution: how long it takes, its resource consumption, and so on. The planning problem is then to generate the plan that leads to the *most preferred* outcome. In the case of uncertainty—in which a plan generates a probability distribution over outcomes—the problem is to generate the plan with maximum *expected value*. This definition demands an *optimal* plan instead of *any* plan that successfully achieves the goal.

This general model of plan success is more than adequate to capture the situations listed above. Its problem is its generality, however. First, instead of goal formulas we must represent complete plan outcomes (all execution traces). Second, instead of implicitly defining two equivalence classes over plan outcomes (those that satisfy the goal and those that do not), we must express arbitrary pairwise preferences.

Furthermore, there is no inherent structure in this representation. Therefore, it is not clear how an algorithm like backward chaining would work. At the most extreme—when the preference model is a black-box relation that only answers whether outcome o_1 is preferable to o_2—the planner is forced to employ a generate-and-test algorithm: propose two plans,

get a sense of how researchers define planning problems, we surveyed the proceedings of the 1992, 1994, and 1996 AI Planning Systems conferences. We identified 67 papers that reported actual planning algorithms (as opposed to those that dealt with formal aspects of plan generation) and defined the problem clearly enough for us to determine the definition of plan success. Of these, 55 adopted the simple goal satisfaction model. Of the remaining 12 papers, 8 adopted the extended goal model proposed in this paper.

generate every possible outcome for each, pass these outcomes to the preference evaluator, discard the less preferred plan, and repeat. There is no guidance regarding how to repair or improve a plan, which is precisely the information supplied in the simple goal model by the goal formula and plan operators. More specifically, there is no way to evaluate how good a plan is until that plan is fully constructed and all its possible outcomes are generated. The ability to evaluate how close a candidate plan is to a solution, during its construction, is a central component of classical planning algorithms.

In this paper we seek a middle ground between the simple goal model and the arbitrary preference model. We extend the notion of a goal into a preference or utility model that includes concepts like temporal extent, partial satisfaction, and efficiency in achievement. At the same time, we retain the explicit structural information in the goal formula. Thus, planning algorithms can still use the preference model to guide the search for optimal plans by evaluating partially constructed ones.

We will use the language of decision theory to develop our model. This provides a rational basis for extending the planning problem into probabilistic domains.

The paper is structured as follows. Section 2 discusses the role of goals in the plan-generation process, and proposes and justifies a utility function for a goal-directed agent. This function is expressed in terms of individual utility functions for each of the agent's goals and for resource attributes. Section 3 introduces concepts and notation from multiattribute utility theory required to develop the model formally. Section 4 then develops a formal utility model for goal-directed agents.

The paper then addresses computational questions: How can a planning agent exploit this utility model in the process of generating plans? Section 6 reports on one formal and two algorithmic techniques for generating plans, each of which exploits explicit structural aspects of the utility model.

Three sections conclude the paper: Section 7 summarizes the model and addresses validation issues, Section 8 discusses related work, and Section 9 suggests areas for future research.

2. CLASSICAL GOAL-DIRECTED AGENTS

Before formally analyzing goals and utilities, we must define more precisely what role these concepts play in a planning system. We also examine the limitations of behavior that is strictly goal directed, and explore extensions to the simple goal model.

2.1. The Role of Goals in Automated Planning Systems

Goals typically play three roles in automated planning systems. They

1. communicate information about the planning problem. In particular, goals provide a concise definition of what constitutes a successful plan.
2. limit inference in the planning process by allowing the planner to examine and build chains of dependencies over goal formulas. Thus, goals define exactly what is *relevant* to the planning problem.
3. restrict the temporal scope of the planning problem, imposing a temporal "horizon" beyond which planning is irrelevant.

In the first case goals can be communicated more easily than utility functions, in cases two and three the goals' symbolic content can aid in the search for good plans. Thus, there is a fundamental link between the goal model and an appropriate solution algorithm.

2.2. Limitations of Goal-Directed Behavior

Suppose a problem solver is given only a symbolic goal expression, a goal formula G representing the target world state(s). Assume the problem solver then builds a plan that maximizes the probability that the world will be in a state in which G holds when plan execution ends. This is the probabilistic analogue of logical goal-directed planning, in which the problem solver constructs a plan that *provably* achieves G. What limits does this model place on the problem solver's preference structure? In other words, under what circumstances is planning to maximize expected utility equivalent to planning to maximize the probability of goal satisfaction?

A *plan* can be viewed as defining a probability distribution over *chronicles*. A chronicle is a complete specification of the world state over time, representing one possible way plan execution might play out. We can therefore define the probability that a plan $\mathcal{P}$ generates a chronicle c as: $\mathbf{P}(c|\mathcal{P})$.[2] We are interested in the time point representing the moment the plan finishes executing; we use end(c) to represent this point. The probability of success in the classical paradigm is the probability that the goal conjunction will hold when the plan finishes executing:

$$\mathbf{P}(\mathcal{P} \text{ succeeds}) \equiv \mathbf{P}(\mathsf{G}|\mathcal{P}) \equiv \sum_{\{c:\ \mathsf{G} \text{ holds at end}(c)\}} \mathbf{P}(c|\mathcal{P}). \tag{1}$$

The planner tries to find the plan that maximizes this value.

A decision-theoretic planner has the same probabilistic model as the probabilistic planner. However, it also uses a utility function over chronicles, a function $\mathbf{U}(c)$ that maps chronicles into real values. The expected utility of a plan is defined as:

$$\mathbf{EU}(\mathcal{P}) \equiv \sum_{c} \mathbf{U}(c) \cdot \mathbf{P}(c|\mathcal{P}). \tag{2}$$

The decision-theoretic planner tries to find the plan that maximizes this value.

To understand the relationship between these two models, we can ask under what circumstances the two models are equivalent; that is, for what utility models (definitions of $\mathbf{U}(c)$) is the plan that maximizes the probability of goal achievement (Eq. (1)) the same as the plan that maximizes expected utility (Eq. (2))?

The answer is that this relationship holds only for simple, two-valued utility functions—functions for which utility is a constant low value $\mathsf{U}\overline{\mathsf{G}}$ for chronicles in which the goal does not hold at the end of plan execution, and a constant high value UG for chronicles in which it does. Figure 1 shows such a function; utility is represented along the vertical axis, and the space of chronicles along the horizontal axis. G and $\overline{\mathsf{G}}$ designate the set of all chronicles in which the goal holds and does not hold, respectively.

Previous work (Haddawy & Hanks 1990) demonstrates the correspondence between these two policies, showing that the simple class of step functions pictured in Figure 1 is the *only* class of utility functions for which the relationship

$$\mathbf{P}(\mathsf{G}|\mathcal{P}_1) > \mathbf{P}(\mathsf{G}|\mathcal{P}_2) \Rightarrow \mathbf{EU}(\mathcal{P}_1) > \mathbf{EU}(\mathcal{P}_2) \tag{3}$$

holds for any plans $\mathcal{P}_1$ and $\mathcal{P}_2$. In other words, describing the desired state of the world in terms of a goal formula restricts a problem-solver's preference structures to those preferences that can be characterized by a simple step function. Haddawy and Hanks (1990) also explored

[2] McDermott (1982) defines chronicles using a temporal logic, and Hanks (1990b) and Haddawy (1991) extend the notion to a probabilistic framework.

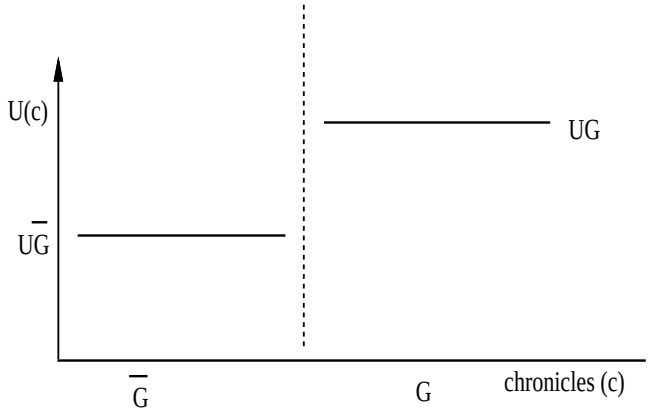

FIGURE 1. Utility function for simple goal satisfaction.

the relationship between goal satisfaction and probability maximization for other situations, such as cases in which goal expressions are combined with preference information about resource consumption.

This analysis illustrated some limitations of planning to achieve a goal conjunction, which we will address later in this paper. The model can be extended in other directions as well:

1. *Temporal extent*. Defining plan success in terms of what is true at the end of execution (i.e., defining a successful chronicle only in terms of whether the goal holds at end(c)) rules out goals such as: deadlines (have G_1 true by noon and G_2 true by midnight), maintenance (keep G true continuously between noon and midnight), and prevention (make G_1 true, but without allowing G_2 to become true in the meantime). The last is a combination of deadline and maintenance goals. *What* is accomplished is important, but so is *when* or for *how long*.
2. *Trade-offs among the goals*. The classical model dictates that the satisfaction of all goal conjuncts is necessary and sufficient for success. A reasonable extension to this model is the view that satisfying *some* conjuncts is preferable to satisfying *none*. Thus, success in achieving one conjunct could be traded off against success in achieving others. In this case each goal is still an all-or-nothing proposition; however, satisfying some subset of the goals might be preferable to satisfying none.
3. *Partial satisfaction*. Symbolic goals imply a binary success criterion: either the goal formula holds at the end of execution or it does not. This criterion is reflected in the step function: utility is either at a constant high value or at a constant low value. A realistic representation should make it possible to (1) define what it means to satisfy a goal partially, and (2) specify that satisfying G entirely is most preferred, but satisfying G partially is better than not satisfying it at all. To reason about partial satisfaction, it is also useful to introduce metric quantities into the model. For example, we could express the fact that delivering three tons of rocks to the depot is preferable to delivering two, but that delivering less than two tons is useless.
4. *Incidental costs and benefits*. Symbolic goals imply that goal attributes are the only aspects of a chronicle that are relevant to assessing utility. This rules out the possibility that plan $\mathcal{P}_1$ is preferable to $\mathcal{P}_2$ because it is as likely to achieve the goal as $\mathcal{P}_2$, but at

a lower cost. Symbolic goals provide no way to specify the "lower cost" aspect, nor do they provide the language to express the trade-off between effectiveness in achieving the goal and the cost involved in doing so.

This list is not exhaustive, and the items are not mutually exclusive: minimizing consumption of a resource can be viewed as having the goal of consuming none of it, but allowing partial satisfaction of that goal. Our analysis is intended to demonstrate some useful extensions to the simple goal model, and hence the target of our incremental extensions to that model:

- The language of goals should be extended to represent temporally scoped goals, partial goal satisfaction, and resource-related utility.
- To use these extended goal forms effectively in planning scenarios, we must establish relationships like that shown in Eq. (3): circumstances under which planning to maximize the probability of goal satisfaction guarantees rational behavior in the decision-theoretic sense.
- We need to exploit these relationships as we build or refine plans.

We next present a utility model that extends the simple goal model's expressive power, but retains an explicit representation of the goals. Since the model uses concepts from multiattribute utility theory, we begin with a short review of the relevant concepts and terminology.

3. MULTIATTRIBUTE UTILITY THEORY

Our development of utility functions makes extensive use of several concepts related to independence properties of multiattribute utility functions, so we begin by providing definitions taken from Keeney and Raiffa (1976). We then present a simple example to clarify the concepts presented.

3.1. Concepts and Terminology

Independence properties enable the concise specification of a utility function over a set of attributes in terms of simpler functions over subsets of those attributes. The user is also referred to von Neumann and Morganstern (1947) and Raiffa (1968) for more information on the basic concepts and notation of decision theory.

We begin by formalizing the concept of an *outcome*. In decision theory an outcome is defined as a description of the consequences of a course of action, specifying the state of everything that is of value to the decision maker. This is equivalent to the notion of a *chronicle* introduced in the previous section. We will use the terms interchangeably, preferring the term *chronicle* when discussing concepts from the planning literature and the term *outcome* when discussing concepts from utility theory.

Outcomes are typically specified in terms of value assignments to a set of *attributes*. We use capital letters to designate attributes. The *basic attributes* in most of our discussions is the set $\mathbf{A} = \{A_1, A_2, \ldots, A_n\}$, where each A_i assumes a single value from its *domain*—a totally ordered, possibly infinite set. Attributes correspond roughly to the predicate symbols used to model a planning domain, but the correspondence must account for the temporal dimension as well. Therefore, the correspondence is actually between an attribute A_i and a propositional symbol evaluated at a particular point in time.

It is generally the case that attributes can be partitioned into *sets* that share preference characteristics. For example, two attributes A_i and A_j might be related because they refer

to two related propositional attributes evaluated at the same time, or because they refer to the same propositional attribute evaluated at different times. A set of attributes can also be considered an attribute. If B is a nonempty subset of the full attribute set $\mathbf{A}$, we will refer to B as "attribute B." The complement of B, $\mathbf{A} - B$, is denoted by $\overline{B}$.

An *outcome* is an assignment of values to each attribute from the appropriate domain: $a = \{a_1, a_2, \ldots, a_n\}$. When referring to the portion of an outcome that applies to a subset B of $\mathbf{A}$, we use the notation $(b, \overline{b})$ to indicate the partition of the outcome into two subsets.

A *preference order* $\succeq$ is a total ordering on the set of all outcomes. We say that outcome a_1 is *weakly preferred to* outcome a_2 if $a_1 \succeq a_2$. Given such an order, we can define notions of strict preference and indifference. For example, we say that a_1 is strictly preferred to a_2 if $a_1 \succeq a_2$ and it is not the case that $a_2 \succeq a_1$. A central result from decision theory states that if preferences satisfy certain axioms of "rationality," then there exists a real-valued *value* function v on the space of all outcomes, such that $a_1 \succeq a_2$ if and only if $v(a_1) \geq v(a_2)$.

In situations where uncertainty is involved, a preference model must capture not only the decision maker's attitude toward different outcomes, but also his attitude toward uncertainty or risk. For example, one might prefer a plan that usually offers adequate performance to one that usually offers exceptional performance if the latter leads to a catastrophe when it fails. Attitudes toward uncertain outcomes are generally assessed by eliciting preference information about *lotteries* over outcomes. A *two-outcome lottery* is an $\langle a_1,\ p,\ a_2\rangle$-tuple where a_1 and a_2 are outcomes, and p is a probability. The lottery describes a hypothetical situation in which outcome a_1 occurs with probability p, and outcome a_2 occurs with probability $(1 - p)$. Lotteries with more than two outcomes are defined similarly. When preferences over lotteries are rational, they can be captured by a real-valued *utility function* u, defined on the space of outcomes, such that $a_1 \succeq a_2$ if and only if $u(a_1) \geq u(a_2)$. Thus, a value function is a special case of the utility function in cases where there is no uncertainty about the outcome of taking action.

Defining a complete preference model for a planning agent therefore requires eliciting preferences over all lotteries over all possible outcomes, which is a daunting task. For this reason we look for regularities in the preferences that allow the model to be defined in terms of preferences over its components. For example, preferences over fuel consumption (a set of attributes) might be the same regardless of the time at which the fuel is consumed (a different set of attributes). Likewise, preferences over whether a goal is satisfied might not depend on how much fuel was consumed in achieving the goal; in this case, preferences over these two attributes could be elicited separately.

The key to discovering and exploiting these relationships is to find *independence relationships* among attributes. The following definitions clarify the concept of independence.

Definition 1 (Preferential Independence). An attribute B ($B \subset \mathbf{A}$) is preferentially independent (PI) of its complement $\overline{B}$ if the preference order over outcomes involving only changes in the level in B does not depend on the levels at which attributes in $\overline{B}$ are held fixed.

Utility independence extends the concept of preferential independence to situations where the consequences of an action are uncertain:

Definition 2 (Utility Independence). An attribute B ($B \subset \mathbf{A}$) is utility independent (UI) of its complement $\overline{B}$ if the preference order over lotteries involving only changes in the level of B does not depend on the levels at which attributes in $\overline{B}$ are held fixed.

Since outcomes are simply degenerate lotteries (lotteries where the probability is 1.0), PI immediately follows from UI. The converse is not true, however, as UI is the stronger condition.

For the remainder of this paper we focus on the concept of utility independence. Instead of saying that B is UI of its complement $\overline{B}$, we simply say that B is UI. If there are only two basic attributes, $\mathbf{A} = \{A_1, A_2\}$, we refer to them as being PI or UI of one another without reference to a set B.

Utility independence is not symmetric in the same way as is probabilistic independence. It is possible for an attribute B to be UI, but not vice versa, i.e., $\overline{B}$ is not UI. When B is UI for all $B \subset \mathbf{A}$, we say that the attributes of $\mathbf{A}$ are *mutually utility independent* (MUI).

The significance of utility independence is that the structure in the preferences over independent attributes leads to "simpler" forms of the utility function over these attributes. Simpler means that a high-dimensional utility function can be reduced to a function of other functions with lower dimensionality, resulting in fewer values to elicit. For example, if B is UI of $C = \overline{B}$, the utility function can be completely specified by two conditional utility functions for C and one conditional utility function for B:

$$u(b, c) = u(b_0, c) + u(b, c_0)[u(b_1, c) - u(b_0, c)]. \tag{4}$$

If B and C are mutually utility independent—that is, if C is also UI—then the utility function takes an even simpler form:

$$u(b, c) = k_B u_B(b, c') + k_C u_C(b', c) + k_{BC} u_B(b, c') u_C(b', c), \tag{5}$$

where b' and c' are arbitrarily chosen specific values of B and C. The joint utility function can thus be built by eliciting only the two simpler conditional utility functions, u_B and u_C, and three scaling constants.

It is natural to ask when the multiplicative term drops out of Eq. (5); that is, what additional assumptions guarantee that $k_{BC} = 0$? This case is called *additive independence* (*AI*), because the utility function is the sum of the utility functions for its component attributes.

Definition 1 (Additive Independence). The attributes of the set $A = \{A_1, A_2, \ldots, A_n\}$ are called additive independent (*AI*) if the utility function $u(a)$ can be expressed in the form:

$$u(a) = \sum_{i=1}^{n} k_i u_{A_i}(a_i).$$

An alternative definition for additive independence is to say that the preferences over lotteries on the attributes $A_1, A_2, \ldots, A_n$ depend only on their marginal probability distributions, not on their joint probability distribution (Keeney and Raiffa 1976, Th. 6.4).[3]

It is clear that additive independence implies mutual utility independence, but the converse is again not true. Refer to Keeney and Raiffa (1976) and the example that follows for a discussion of the conditions that imply additive-independent preferences.

3.2. Example of Preferential, Utility, and Additive Independence

A simple example will help to clarify the concepts of preferential independence, utility independence, and additive independence. Consider having to make decisions that affect two

[3]Note that the k_i coefficients are not formally necessary, since the u_{A_i} can themselves be scaled arbitrarily. They are usually introduced only to illustrate the fact that the u_{A_i} functions have to be scaled with respect to each other.

attributes, W (wealth) and F (fame). The former can take values rich and poor, and the latter can take values infamous, anonymous, and famous.

In the case of certainty, the effect of every action on W and F is deterministic. All that is necessary is a preference ordering over (w, f) pairs. Saying that W is preferentially independent of F amounts to saying that the preference order over the values of W does not depend on the value of F. For example suppose that (rich, f) $\succ$ (poor, f) for all values of f. Likewise, it is possible that our fame-seeking decision maker has preferences of the form (w, famous) $\succ$ (w, anonymous) $\succ$ (w, infamous), which hold for all values of w. We can then say that F and W are mutually preferentially independent, and as a result the decision maker's value function is of the form

$$v(w, f) = k_W v(w) + k_F v(f).$$

For decisions whose outcomes are uncertain, the assessment is over lotteries of the form $\langle (w_i, f_i), p, (w_j, f_j)\rangle$. However, the definition of utility independence is essentially the same: W is utility independent of F if the decision maker's preferences over lotteries of the form

$$\langle (w_i, f), p_1, (w_j, f)\rangle \succ \langle (w_i, f), p_2, (w_j, f)\rangle$$

are identical regardless of the value of f. That is, the decision maker's base preferences over W do not depend on F; in addition, his attitude toward *risky* actions involving W does not depend on the value of F, either.

Notice that MUI does not cover the case where a decision can affect both attributes simultaneously. For example, consider the case where the decision maker is currently poor and anonymous and has two actions that could change both. In order for W and F to be additive independent, the preferences toward changes in W are insensitive to symmetric changes in F. For example, the following relationship is sufficient to guarantee that W and F are additive independent

$\langle$(poor, famous), 0.5, (rich, anonymous)$\rangle \sim \langle$(poor, anonymous), 0.5, (rich, famous)$\rangle$

provided that all three of the following conditions are met:

1. W and F are also MUI
2. (poor, famous) $\not\sim$ (poor, anonymous)
3. (poor, famous) $\not\sim$ (rich, famous).

This concludes our brief introduction to multiattribute utility theory concepts and terminology. We next structure our agent's utility model according to these independence assumptions.

4. UTILITY MODELS FOR GOAL-DIRECTED AGENTS

This section develops a utility model for a goal-directed agent by describing the form of the function $\mathbf{U}(c)$ introduced in Section 2. The task is simple for the classical goal model in the previous section

$$\mathbf{U}(c) = \begin{cases} 1 & \text{if G is true at end}(c) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where we can view a chronicle as a set of attributes capturing the state of the system at all points in time. Thus, "G at end(c)" is a single attribute.

However, we want our model to capture these facts as well: that goal satisfaction can be measured at other time points and over intervals, that goals can be partially satisfied, that (partial) satisfaction of one goal can be traded off against (partial) satisfaction of other goals, and that resource costs also affect the extent to which a plan succeeds. There are thus two main properties we want to capture to make our model more robust:

1. Satisfying a goal to a greater extent is preferable to satisfying it to a lesser extent, all other things being equal.[4]
2. Success in satisfying one goal component can be traded off against success in satisfying another goal, or against consuming resources.

It is also important that symbolic information about the goal be explicit in the model so that information can be exploited by planning algorithms.

In Section 4.1 we develop a definition for the agent's top-level utility function in terms of its goals and preferences about resource consumption. Section 4.2 then discusses how the model handles goal interactions, a common topic in classical planning research.

4.1. The Agent's Utility Model

We begin by specifying the top-level functional form for the agent. We assume that the top-level utility function is composed of (1) n individual utility functions UG_i associated with the agent's n explicit goals, and (2) a utility function UR (for "residual" or "resource" attributes) that measures the extent to which the outcome produces or consumes nongoal attributes, e.g., time, fuel, wear and tear on equipment, money lost or gained.

We further assume that the n top-level goals and the resource attributes are mutually utility independent. The separation of residual utility from goal utility and the MUI assumption provides strong computational leverage; it allows dominance among plans to be shown by demonstrating dominance along each of the individual goal and resource dimensions. For example, the PYRRHUS planner discussed in Section 6.2 calculates upper bounds on the expected utility of a plan by calculating upper bounds on the expected utility for each UG_i and for UR. Similarly, the qualitative filtering technique discussed in Section 6.3.2 can be used to establish stochastic dominance relationships among plans by establishing those relationships among the individual goal and residual utility functions.

Under these assumptions, the top-level utility function has the following multiplicative form.[5] For simplicity of notation in the equation below, we define $\mathsf{UG}_{n+1} = \mathsf{UR}$.

$$\begin{aligned}\mathbf{U}(c) \;=\; & \sum_{i=1}^{n+1} k_i \mathsf{UG}_i(c) + k \sum_{i=1,\, j>i}^{n+1} k_i k_j \mathsf{UG}_i(c)\mathsf{UG}_j(c) \\ & + k^2 \sum_{i=1,\, j>i,\, l>j}^{n+1} k_i k_j k_l \mathsf{UG}_i(c)\mathsf{UG}_j(c)\mathsf{UG}_l(c) \\ & + \cdots \\ & + k^n k_1 k_2 \ldots k_{n+1} \mathsf{UG}_1(c)\mathsf{UG}_2(c)\ldots\mathsf{UG}_{n+1}(c),\end{aligned} \tag{7}$$

[4] See Wellman and Doyle (1991) for a more general interpretation of this criterion.

[5] See Keeney and Raiffa (1976, Ch. 6) for details of this and later derivations.

where k is a scaling constant that is the solution to:

$$1 + k = \prod_{i=1}^{n+1}(1 + kk_i).$$

Equation (7) reduces to the following additive form when $\sum_{i=1}^{n+1} k_i = 1$, $k = 0$:

$$\mathbf{U}(c) = \Sigma_{i=1}^{n+1} k_i \mathsf{UG}_i(c), \tag{8}$$

which can be rewritten as:

$$\mathbf{U}(c) = \Sigma_{i=1}^{n} k_i \mathsf{UG}_i(c) + k_{\mathrm{r}} \mathsf{UR}(c). \tag{9}$$

When $\sum_{i=1}^{n+1} k_i \neq 1$, Eq. (7) can be rewritten as:

$$k\mathbf{U}(c) + 1 = (kk_r \mathsf{UR}(c) + 1) \prod_{i=1}^{n}(kk_i \mathsf{UG}_i(c) + 1). \tag{10}$$

Verifying that top-level goals and the resource attributes are MUI seems daunting, since it would require us to test utility independence for all subsets of the goals and resources. Fortunately, there are various simpler conditions that imply MUI. For example, Keeney and Raiffa (1976, Th. 6.2) show that for a set of attributes $\{X_i\}$, if $\{X_i, X_{i+1}\}$ is UI for $i = 1, 2, \ldots, n-1$, $n \geq 3$, then the X_i are MUI. For many problems, even this simplified condition would be impossible to verify; therefore, in practice MUI is typically determined by asking the decision maker for preferences at a few points, which are then verified by asking general questions about the independence of preferences over the attributes. Thus, for any planning situation, the validity of the MUI assumption (1) is subject to verification for the particular problem, and (2) will depend on the choice of goals and attributes used to model the problem. We return to this second point below.

Note that at the top level, the model explicitly represents trade-offs between satisfying individual goals as well as the trade-off between goal satisfaction and resource consumption. For the additive form, for example, this dependency is defined in terms of the $n + 1$ numeric parameters $k_1, k_2, \ldots, k_n, k_r$, only n of which are formally necessary. The ratio of any two of these numbers indicates the amount the agent is willing to sacrifice in satisfaction of one goal in order to satisfy another, or the amount of satisfaction in resource consumption the agent is willing to "spend" in order to increase the extent to which a goal is satisfied.

4.2. Goal Interactions

The MUI restriction on top-level goals seems troubling on the surface. This is because AI planning research has focused mainly on interactions among multiple goals, and our assumption seems to rule out these interactions. In particular, the MUI assumption seems to run counter to the analysis of conjunctive-goal planning presented in Section 2. Equation (7) means, for example, that we cannot represent conjunctive goals such as "have the truck fueled and loaded by 7 a.m." as two separate top-level goals because satisfying either one without the other presumably affords low utility but their conjunction affords high utility. This is indeed the case, and the solution is to represent these two conjuncts not as two goals, but as two interacting components of a *single* goal. In Section 5.2 we provide a model that allows interactions among symbolic components that can interact to comprise a goal. We demonstrate how to represent traditional conjunctive goals of this form: "top-level conjunctive goal G is satisfied if and only if its components $\mathsf{g}_1 \ldots \mathsf{g}_n$ are all fully satisfied."

We should also note that the assumption of utility independence does not imply that goals are *probabilistically* independent. One might object that two goals—"have the truck at the depot by noon" and "have the truck clean"—interact strongly if the only road to the depot is muddy, and there is no way to wash the truck once it arrives at the depot. In particular, there might be no plan that would make them both true. But this situation means only that the two goals are not *probabilistically* independent; it does not constitute a violation of *utility* independence. The MUI assumption demands only that the *utility* derived from satisfying one top-level goal does not depend on the extent to which the other goals are achieved. It does not address the likelihood of achieving either goal in isolation, or both simultaneously. The likelihood of achieving goals is properly reflected in the probabilistic model of the domain and the operators. It is not a reflection of the decision maker's fundamental preferences.

The main implication of mutual utility independence is that the decision maker must structure his preferences, identifying those that are utility independent and those that are not. The former are divided into separate goals. The latter become components of individual goals, which are allowed to interact.

We now turn to the question of how to express the goal information—the $\mathsf{UG}_i(c)$. We temporarily ignore the top-level utility function $\mathsf{U}(c)$ and the residual utility function $\mathsf{UR}(c)$, to concentrate on how to build utility functions for individual goals.

5. UTILITY FUNCTIONS FOR INDIVIDUAL GOALS

In classical planning algorithms, goals consist of a symbolic expression to be achieved. Our paper extends this idea in several directions, including giving goals a temporal extent and allowing them to be partially satisfiable. Partial satisfaction should be allowed for both numeric goals ("deliver 10 tons of cargo") and for more traditional symbolic goals ("have all the parts painted and on the loading dock by noon"). Partial satisfaction should extend to the temporal component as well: if the deadline is noon, finishing by 12:05 might be better than finishing the next morning.

We begin by breaking a goal into *static* and *temporal* components. The former indicates *what* is to be achieved, the latter *when* it is to be achieved. The goal "deliver two tons of material to the depot by noon" has (1) a static component, described by the variable "number of tons delivered to the depot," and (2) a temporal component representing the deadline of noon and information about preferences for deliveries that occur after the noon deadline.

We define two types of temporal goals: *deadline goals* and *maintenance goals*.[6] A deadline goal says that some condition should be true by a deadline point. Two examples of deadline goals are:

- Have block A on block B and block B on block C by noon.
- Have two tons of rocks at the depot by 2:00 this afternoon.

A maintenance goal represents a desire to keep a condition true over an interval of time. Two examples of maintenance goals are:

- Keep the temperature between 65 and 75 degrees from 9 a.m. until 5 p.m.
- Keep all tools in the warehouse between midnight and 8 a.m.

[6]These particular extended goal forms—temporal deadline points and maintenance intervals—are not uncommon in the literature (see, e.g., McDermott (1982), Drummond (1989), Allen et al. (1991)), though this work does not address the problem of reasoning about uncertainty or partial satisfaction.

The remainder of this section describes possible ways to formalize deadline and maintenance goals in terms of temporal and static components and discusses the problem of eliciting utility functions for these goals. We show that under certain utility independence assumptions, the temporal/static separation can be preserved so we can express preferences over partial satisfaction of both components. In particular, we show how to specify the UG_i function for a deadline goal in terms of one utility function for the static component and a set of temporal weighting coefficients. We also demonstrate how to specify the UG_i function for a maintenance goal in terms of one utility function over pairs consisting of a value for the static component and the length of the maintenance interval.

5.1. Deadline Goals

We now develop our utility model $\mathsf{UG}_i(c)$ for deadline goals. Let $\{A_i\}$ denote the set of variables representing the static goal component, so that A_i represents the value of the static goal component at time i. The A_i are attributes as defined in Section 4, but they are constrainted to share the same domain (e.g., "number of tons delivered to the depot"). Thus, for the utility function UG(c), we have a set of basic attributes A_1, A_2, ... tracking the state of the static component over time.

Let the deadline point be d, the earliest time at which satisfying the static component is of value.[7] We also assume that the deadline has a *horizon* h ($0 < d \leq h$), a time after which there is *no* benefit in achieving the static goal component. In other words, temporal attributes $A_0, A_1, \ldots, A_{d-1}$ and $A_{h+1}, A_{h+2}, \ldots$ have no effect on the utility function UG. Theoretically, this is a restrictive assumption, omitting cases such as those in which the effects on utility of temporal attributes A_t asymptotically approach 0 when $t \to \infty$. For most practical problems, however, it is possible to identify *some* time point after which satisfying the goal does not matter. Thus, for purposes of this discussion, a chronicle c is simply the sequence $c = (a_d, \ldots, a_h)$.

The task of specifying the utility function UG(c) is equivalent to specifying the preference order over lotteries with $|A_i|^{h+1}$ possible outcomes (where $|A_i|$ denotes the cardinality of the domain of A_i). Since this is not practical, we must seek techniques to decompose the problem. We assume mutual utility independence in order to decompose UG into simpler components, as we did with the top-level utility function. Mutual utility independence gives rise to a multiplicative form similar to that of Eq. (7). For simplicity of exposition, we make the stronger assumption of additive independence; we note, however, that all results concerning exploitation of the utility function's structure to facilitate computation apply to the case of mutual utility independence, as well.

The rest of this section proceeds as follows. First, we describe a simple model for eliciting the utility function when additive independence and some structural assumptions apply. Next, we discuss situations in which the structural assumptions of our model are violated. We then discuss a situation where *AI* is not directly applicable, but can still be achieved by transforming the static attribute. An extended example of assessing a utility function follows, and the section finishes with a discussion of goals with symbolic attributes and a discussion of maintenance goals.

5.1.1. When AI is Directly Applicable. First, consider the case when *AI* can reasonably be assumed. In this case, the utility function UG for the deadline goal can be expressed in

[7] Note that this does not coincide with the the conventional use of the term "deadline" but is chosen for technical reasons.

the following additive form:[8]

$$\mathsf{UG}(c) = \sum_{i=d}^{h} k_i u_i(a_i). \tag{11}$$

In this equation, the deadline utility function UG, the coefficients k_i, and the utility functions u_i satisfy the following structural assumptions. We assume that the common set of values of the A_i is bounded and totally ordered. We use a_i^- and a_i^+ to denote the least and most preferred values for A_i, respectively. Due to the *AI* assumption, the notions of least and most preferred values are well defined because preferences over values of any A_i are independent of the values of the other A_j. For all i, the utility function $u_i(a_i)$ is the conditional utility function obtained from UG by holding all temporal attributes other than A_i at their least preferred values. Both UG and u_i are scaled from 0 to 1 by setting:

$$\begin{aligned} \mathsf{UG}((a_0^-, \ldots, a_h^-)) &= 0 \\ \mathsf{UG}((a_0^+, \ldots, a_h^+)) &= 1 \\ u_i(a_i^-) &= 0 \\ u_i(a_i^+) &= 1. \end{aligned}$$

The coefficients k_i, which must sum to 1, are expressed as:

$$k_i = \mathsf{UG}((a_d^-, \ldots, a_{i-1}^-, a_i^+, a_{i+1}^-, \ldots, a_h^-)).$$

They can be determined from the probability k_i for which the decision maker is indifferent between the lottery $\langle (a_d^+, \ldots, a_h^+), k_i, (a_d^-, \ldots, a_h^-) \rangle$ and the certain outcome $(a_d^-, \ldots, a_{i-1}^-, a_i^+, a_{i+1}^-, \ldots, a_h^-)$. Since the k_i sum to one and $u_i(a_i^+) = 1$, we have $\mathsf{UG}(a_1^+, a_2^+, \ldots, a_n^+) = 1$.

We can further simplify the utility function by assuming that preferences over values of the attributes A_i are stable over time. Equation (11) can then be simplified by substituting u for u_i:

$$\mathsf{UG}(c) = \sum_{i=0}^{h} k_i u(a_i). \tag{12}$$

The conditional utility function $u(a_i)$ can be assessed at any convenient time point (for example, at the deadline).

A common type of goal that fits this model maximizes some quantity over time. For example, consider the problem of delivering tomatoes to a warehouse. Suppose our goal is to deliver as many tomatoes as possible, with the constraint that our maximum delivery capacity is 10 tons per hour (in other words, $a_i^- = 0$, $a_i^+ = 10$). Suppose our deadline (the earliest point at which delivering tomatoes is of value) is 8 a.m., and we are indifferent as to when deliveries occur between 8 a.m. and noon. As a result, the coefficients k_i for these time points must be equal. Suppose further that during the afternoon, deliveries have linearly decreasing utility until midnight, which is the horizon point after which all utilities are 0; we then set $d = 0$ and $h = 24 - 8 = 16$.

Finally, assume that we are indifferent between delivering some constant amount at every time point between 8 a.m. and noon and delivering that amount at every time point after noon. In this case the sum of the k_i for the time points 0 to 4 must equal the sum for the time points 5 to 16. The coefficients k_i representing these preferences are:

$$k_i = \begin{cases} \frac{1}{10} & \text{if } 0 \le i \le 4 \\ \frac{17-i}{156} & \text{if } 5 \le i \le 16. \end{cases} \tag{13}$$

[8] See Keeney and Raiffa (1976, Ch. 6) for details of this derivation.

The denominator 156 comes from the constraint that $\sum_{i=5}^{16} = 1/2$.

5.1.2. Violation of the Structural Assumptions. In the preceding discussion, we assessed the coefficients k_i by setting up lotteries with outcomes $(a_d^-, \ldots, a_h^-)$ and $(a_d^+, \ldots, a_h^+)$. But one or both of these hypothetical outcomes might not be physically possible. Asking a decision maker to entertain a lottery with impossible outcomes may produce an inaccurate assessment. Two possible solutions are to assess the function directly or to choose different outcomes as reference outcomes. We discuss the former approach here and the latter approach in the next section.

This situation commonly arises in goals of *simple attainment*. For example, suppose that we want to deliver a single object to a warehouse and the time at which that object is delivered is significant. Our static attributes might then be "delivery-made-at-this-time," taking values True and False. We can have a chronicle in which the value is False at every time point, but we cannot have one in which the value is True at every time point. In fact, we also have no feasible chronicles in which the value is True at some time then False at all later times, which was used in the construction of the assessment example. Fortunately, in this case we can assess the utility function directly: for horizon h, we have only $h + 1$ possible chronicles representing all possible different times the delivery could occur.

5.1.3. When AI is Indirectly Achievable. One common type of goal in which the attributes A_i may violate the *AI* assumption is an *aspiration-level* goal.[9] An aspiration-level goal is one in which we want to attain a particular level or accumulation of a quantity.

For example, suppose that in the preceding tomato delivery example our goal was to deliver 100 tons and deliveries in excess of this amount have no value. If the static attributes A_i are defined as "the amount of tomatoes delivered at time i," then *AI* no longer holds. If 10 tons of tomatoes were delivered each hour between 9 a.m. and 6 p.m., reaching a total of 100 tons by 6 p.m., then *AI* would require the decision maker to be indifferent regarding the amount of tomatoes delivered after 6 p.m. The violation of *AI* becomes apparent when one realizes that the decision maker's preferences over an additional delivery at 7 p.m. must depend on whether 100 tons or 0 tons had been delivered prior to that time.

It is important to note that such violations of *AI* occur only when the value of the static attributes exceed the aspiration level. Violations can therefore be avoided by defining static attributes so they do not have a threshold level that can be attained and exceeded.

Techniques for choosing the "right" attributes over which to express preferences (so that independence assumptions can be made) have been studied extensively (Bell 1977, Ch 18; Keeney & Raiffa 1976, Ch. 2). One of the commonly used methods is to transform the original attributes into new ones for which the desired independence property does apply.

For our example, we define new attributes $B_i, i = 0, 1, \ldots, 16$ as "the number of tons delivered at time i that does not cause the total of all deliveries to exceed the aspiration level of 100 tons." Formally, the B_i are derived from the A_i as follows:

$$B_0 = \min\{A_0, 100\}$$
$$B_i = \min\left\{A_i, 100 - \sum_{j=0}^{i-1} B_i\right\}, \qquad 1 \le i \le 16.$$

Through this transformation we have grouped outcomes over the A_i attributes into equivalence classes, each of which has a particular pattern of values of A_i up to the time that the

[9] Recall that we are discussing an individual deadline goal here, so the A_i attributes refer to the evolution of the static goal component over time. Violation of AI in this context means that preferences over *when* the goal is achieved are not independent.

aspiration level is met. These are also equivalence classes with respect to preference: the decision maker is indifferent among all the elements of each class.

Note that in this case, assessments of the coefficients k_i and the utility functions u_i are accomplished differently. The reason is that outcomes in which each B_i is 10 are impossible, and using impossible outcomes in lotteries for assessment can be problematic.

To solve this problem we realize that a single-attribute utility function, such as u_i, can be assessed by choosing *any* two attribute values as references as long as one is preferred over the other. The assessment is typically easiest for the decision maker if the least and most desirable outcomes are chosen, but this is not strictly necessary. We address the issue of getting the maximum "spread" between reference values in the assessment example below.

For this example we can set b_i^+ to five, and thus ensure that $\sum_{i=0}^{16} b_i^+ = 85 < 100$. Since $u_i(5)$ is set to one, $u_i(10)$ will be greater than one. This is perfectly acceptable since utility functions are unique only up to a positive linear transformation.

5.1.4. Extended Assessment Example. Suppose our goal is to deliver four tons of tomatoes to a warehouse by a deadline 240 minutes from now and that our planning horizon is 280 minutes from now. We represent time using a discrete set of points denoting one-minute intervals from 240 to 280: $t = 1, 2, \ldots, 40$. Since we have a goal with an aspiration level, we take the static attributes to be "the number of tons delivered that does not cause the total deliveries to exceed four tons" at each of the time points 1 to 40. We assume that preferences over values of these attributes are additive independent and stable, and so can be captured by the additive utility function of Eq. (12). Once again, this analysis holds for MUI attributes as well, and we are making the *AI* assumption for expository purposes only.

To specify the utility function, we must first choose the reference values a_i^- and a_i^+. To afford the maximum possible "spread" between outcomes for the assessment of the k_i, the obvious choice is the least and most preferred values that can possibly be attained at all time points in the outcome. Under this choice $a_i^- = 0$, and a_i^+ is the outcome that would have us delivering the four tons equally at each of the 40 time points: $a_i^+ = \frac{4}{40} = 0.1$.

Next, we must assess the temporal coefficients k_i and the $u(a_i)$ function. To assess k_i values, we elicit a few points and fit a curve to them, keeping in mind that the k_i must sum to one and that the curve must be monotonically nonincreasing in i. We ask the decision maker:

> For what probability p_1 are you indifferent between the lottery $\langle (.1, .1, \ldots, .1),\ p_1,$ $(0, 0, \ldots, 0)\rangle$ and the outcome $(.1, 0, \ldots, 0)$?

Suppose he responds with the value $\frac{2}{41}$. After asking him for more of the k_i values and fitting a curve to the points, we find that the values of k_i form a linearly decreasing function of i. Since the index i ranges from 1 to 40, $k_i = \frac{2}{41} - \frac{(i-1)}{820}$, and our utility function takes the form:

$$\mathsf{UG}(c) = \sum_{i=1}^{40} \left[\frac{2}{41} - \frac{(i-1)}{820} \right] u(a_i). \tag{14}$$

Now we must determine the function $u(a_i)$. To assess the single-attribute utility function, we must first choose two values as reference points. We then ask the decision maker about his preferences over lotteries using the chosen reference values as outcomes. For ease of assessment, we would like to choose the extreme values 0 and 4 as reference points, but we have already committed to $a_i^- = 0$ and $a_i^+ = .1$, so $u(0) = 0$ and $u(.1) = 1$. To solve this problem, we can assess a utility function $u'(a_i)$ using 0 and 4 as the reference values, then scale it by dividing by $u'(.1)$ to obtain a function $u(a_i)$ for which $u(.1) = 1$. Utility functions are unique up to a positive linear transformation, so this scaling preserves the

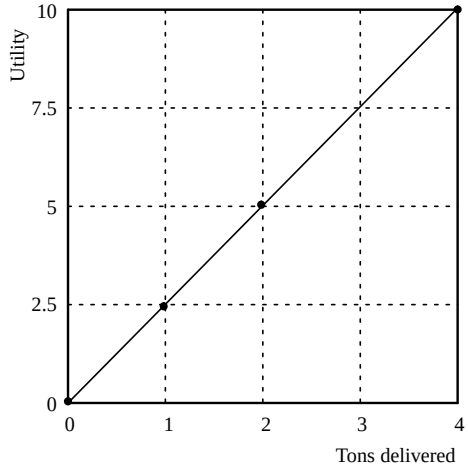

FIGURE 2. Example utility function for the static attribute.

assessed preference order, and the functions are *strategically equivalent*. Therefore, we let $u'(0) = 0$ and $u'(4) = 1$ and choose the deadline $t = 240$ as the time at which to assess the utility function.

We begin by asking the decision maker:

> For what amount of delivery x are you indifferent between the outcome $(x, 0, 0, \ldots, 0)$ and the lottery $\langle (4, 0, 0, \ldots, 0),\ 0.5,\ (0, 0, \ldots, 0)\rangle$?

Suppose the decision maker is risk neutral and responds with the value 2. Then we know that $u'(2) = .5u'(4) + .5u'(0) = .5$. Now we can ask him:

> For what amount of delivery x are you indifferent between the outcome $(x, 0, 0, \ldots, 0)$ and the lottery $\langle (2, 0, 0, \ldots, 0),\ 0.5,\ (0, 0, \ldots, 0)\rangle$?

Suppose he responds with the value 1. Then we have $u'(1) = .25$. We can continue asking these questions to obtain as many points along the utility function as we like. Once we feel we have enough data points, we can fit a curve. We can obtain u by dividing u' by $u'(.1) = 1/40$. Thus, $u(x) = 10x$. The curve for u is shown in Figure 2.

5.1.5. Comparing Plans. We can now use the assessed utility function to compare two plans, $\mathcal{P}_1$ and $\mathcal{P}_2$. Suppose that $\mathcal{P}_1$ has a 50% chance of resulting in the chronicle $c_1 = (1, 1, 0, \ldots, 0)$ and a 50% chance of resulting in the chronicle $c_2 = (2, 0, 0, \ldots, 0)$. Further, suppose that $\mathcal{P}_2$ has a 60% chance of resulting in the chronicle $c_3 = (0, 2, 0, \ldots, 0)$ and a 40% chance of resulting in the chronicle $c_4 = (2, 0, 0, \ldots, 0)$. Referring to the graph of $u(a_i)$, we find that $u(1) = 10$ and $u(2) = 20$. Substituting these values into Eq. (14) above yields $u(c_1) = 39.5/41$, $u(c_2) = 40/41$, $u(c_3) = 39/41$, and $u(c_4) = 40/41$. Therefore, the expected utility of $\mathcal{P}_1$ is $39.75/41$, and the expected utility of $\mathcal{P}_2$ is $39.4/41$. Hence, $\mathcal{P}_1$ is preferred to $\mathcal{P}_2$. This preference is plausible, since both plans deliver an equal amount of tomatoes, and $\mathcal{P}_1$ has a better chance of making the deliveries at the deadline.

5.2. Symbolic Static Attributes

Our discussion of eliciting UG_A functions has to this point dealt only with *numeric* static attributes A_i, for which we assessed the $u(a_i)$ function by fitting a curve to the range of numeric values that the A_i could assume. In many planning applications, however, the goal has a more symbolic nature, e.g., a conjunction of atomic propositions. In these cases, the static attribute has a discrete set of values.

We now consider how to define a function specifying partial satisfaction of symbolic-attribute goals. Consider, for example, the (deadline) goal of having block B on top of a red block by noon. We want to represent situations like one in which it is important to have B on an object, but perhaps less important that the object be red.

The static component utility function $u(a_i)$ for a symbolic-attribute goal is defined in terms of a sequence S of mutually exclusive and exhaustive formulas $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ such that:

- σ_n is the most preferred formula (representing complete satisfaction)
- σ_i is preferred over σ_j if $i > j$.

The $u(a)$ value is then assigned to each formula, with the condition that $u(\sigma_1) = 0$ and $u(\sigma_n) = 1$. For the preceding example, S might be

i	σ_i	$u(\sigma_i)$
1	$\neg\exists x \mathrm{On}(B, x)$	0.0
2	$\exists x \mathrm{On}(B, x) \wedge \neg\mathrm{Red}(x)$	0.7
3	$\exists x \mathrm{On}(B, x) \wedge \mathrm{Red}(x)$	1.0

As we suggested, some partial satisfaction is accrued from making the *On* relationship true even without the *Red*, but no satisfaction is accrued if *Red* is true without *On*.

The simplest such function would be one that admits no partial satisfaction of the goal. Recall the example in Section 4.2, of having the truck loaded and fueled by 7 a.m., where accomplishing one goal without the other yields no utility. The $u(a_i)$ function for that conjunction is:

i	σ_i	$u(\sigma_i)$
1	$\neg$(truck-loaded $\wedge$ truck-fueled)	0.0
2	truck-loaded $\wedge$ truck-fueled	1.0

The rest of the analysis for quantitative goals—the independence assumptions, elicitation of temporal weighting coefficients, and definition of the UG_A function by combining k_i with $u(a_i)$—now applies. The $u(a_i)$ value in this case is the one associated with the (unique) σ_i value that is true at the time the function is evaluated.

5.3. Maintenance Goals

A maintenance goal maintain-over(ϕ, t_b, t_e) specifies that ϕ must hold throughout an interval $[t_b, t_e]$, where ϕ is a formula that expresses some constraints on the value of a variable A, which is defined at all time points during the interval. We assume that A changes value at a finite number n of time points $t_1, t_2, \ldots t_{n-1}$ (where $t_b = t_0 < t_1 < \cdots < t_{n-1} < t_n = t_e$). The interval $[t_{k-1}, t_k]$ is denoted by I_k, for $k = 1, 2, \ldots, n$, and the value of A during I_k is denoted by a_k. The duration of I_k is denoted by d_k: $d_k = t_k - t_{k-1}$.

If the decision maker's preferences over attribute/interval $\langle A_k, I_k \rangle$ pairs are additive

independent, we can express the maintenance goal utility function in an additive form:

$$\mathsf{UG}(c) = \sum_{k=1}^{n} u(a_k, I_k). \tag{15}$$

Furthermore, if the utility function $u(a_k, I_k)$ depends only on the duration of the interval I_k, then the utility function for the maintenance goal takes the following form:

$$\mathsf{UG}(c) = \sum_{k=1}^{n} u(a_k, d_k). \tag{16}$$

We can normalize u by setting $u(a_k^-, t_e - t_b) = 0$ and $u(a_k^+, t_e - t_b) = 1$.

A more detailed account of maintenance goals is provided in Haddawy and Hanks (1993).

5.4. Summary

This concludes our formal development of the utility model. We made two main structural assumptions: (1) mutual utility independence among top-level goals, and (2) for each goal, mutual utility independence among elements of the temporal sequence of static attribute values it generates. The result is a model that is decomposable into individual utility functions for each goal, and within each goal decomposable into temporal and static component preferences.

We have already shown how this decomposition aids in the elicitation process. We now turn to computational matters, showing how the decomposition lets an algorithm use the model's structure to identify attributes and values that lead to high-utility outcomes, and to bound the expected utility of partially constructed plans so that unpromising candidates can be quickly eliminated from consideration.

6. USING UTILITY FUNCTIONS TO GENERATE PLANS

One of the main goals of this paper is to use information in the utility function's structure to guide the building of good plans, which generally involves demonstrating that one plan is preferable to another. At worst, establishing this relationship involves computing the expected utility of both plans, a process that requires generating and evaluating all possible outcomes for each. The two main problems with this approach are that: (1) generating and evaluating all possible outcomes of a plan can be prohibitively expensive (Hanks 1990a) and (2) a plan cannot be evaluated, and thus alternative plans cannot be compared, until it is *complete* (i.e., fully generated or refined). We need ways to compare plans which are cheaper than computing expected utilities over all possible outcomes and which can be applied to plans as they are being constructed.

This section suggests three approaches to exploiting the utility function. The first is formal: it demonstrates relationships between partial plans that involve only the plans' abilities to satisfy goals, but that guarantee a dominance relationship between the two plans. The idea is that we can compare two partial plans on their relative abilities to achieve a goal, such that if certain conditions hold, we can be sure that *any* completion of one plan will dominate *any* completion of the other. Therefore, the inferior plan can immediately be eliminated from consideration.

The other two approaches to exploiting utility information take the form of implemented systems. We report on two planners that exploit the ability to use the model to bound the

expected utility of a partial plan and use that information to prune the search for optimal plans. The first, PYRRHUS, extends a classical plan-space planner to represent partial goal satisfaction and resource-related utility; it uses branch-and-bound techniques to guide the planning process. The second, DRIPS, views the plan-generation process as one of *abstraction refinement*: the planner begins with an abstract plan and repeatedly replaces parts of it with more specific instructions. An abstract plan defines a certain expected-utility *interval*, and each refinement narrows the interval. Whenever the intervals for two partial plans do not overlap, the inferior plan can be removed from consideration.

6.1. Abstract Relationships among Partial Plans

Here is an abstract characterization of the relationship we establish between two partial plans:

> Suppose that one of the agent's goals is G and that there are two formulas, ϕ and ψ, relevant to whether G is achieved. More particularly, if ϕ is true, then G is mostly satisfied (ϕ has a *high* goal-related utility $u(a_i)$), whereas if ψ is true, G is mostly *unsatisfied* (ψ has a *low* goal-related utility).
>
> Further suppose that there are two alternative plans, $\mathcal{P}_1$ and $\mathcal{P}_2$. Both are still under construction, but we can establish some characteristics of each. $\mathcal{P}_1$ is a promising candidate: if it is executed, the probability that ϕ is true by some time l is at least α. $\mathcal{P}_2$ is currently unpromising: if it is executed, the probability that ψ is true at all times before m is at least β. The question is under what circumstances we can say that $\mathcal{P}_1$ dominates $\mathcal{P}_2$ and as a result prune $\mathcal{P}_2$ from the search space.
>
> We show that dominance holds if α exceeds a particular function of five parameters: β, the lowest $u(a_i)$ consistent with ϕ, the highest $u(a_i)$ consistent with ψ, l, and m. Under these circumstances, $\mathcal{P}_1$'s expected utility *must* exceed $\mathcal{P}_2$'s, regardless of how they are subsequently refined.

Having established a relationship of the above form, the planner need only establish two probability bounds associated with formulas ϕ and ψ in order to eliminate $\mathcal{P}_2$ from further consideration.

Two advantages accrue from this technique:

1. It reduces the general problem of expected-utility calculation to the more specific task of deciding whether a particular probability exceeds a particular threshold.
2. It allows us to perform the expected-utility analysis incrementally. At each stage of the planning process, we can eliminate some plans from consideration, again limiting the amount of inference needed to choose a good course of action.

The first advantage is especially important, since it allows the planner to focus its attention on two subsets of the plan's possible outcomes: those in which the threshold is met and those in which it is not. Hanks (1990b) shows how the ability to reduce the plan-evaluation problem to one of computing the probability of a proposition with respect to a threshold allows efficient evaluation of probabilistic plans. We demonstrate these relationships for one class of goals: those with temporal deadlines and symbolic static attributes.

6.1.1. Deadline Goals. We now compare two plans on the basis of their ability to achieve a goal whose temporal component is a deadline t_d and whose static component is symbolic (see Section 5.2). Assume that time is discrete and that we have the additive UG

function of Eq. (12). Formula ϕ indicates a high level of goal satisfaction and formula ψ a low level. Consider the case in which $\mathcal{P}_1$ is likely to achieve ϕ at or close to the deadline, and $\mathcal{P}_2$ is likely to do no better than ψ until some time well past the deadline. Under what circumstances can we say that $\mathcal{P}_1$ has higher expected utility than $\mathcal{P}_2$?

Suppose the goal's $u(a_i)$ function is defined as follows:

i	σ_i	$u(\sigma_i)$
1	$\neg$On(B, C)	0.0
2	$\neg$On(A, B) $\wedge$ On(B, C)	0.5
3	On(A, B) $\wedge$ On(B, C)	1.0

In other words, we accrue partial satisfaction by achieving On(B,C) alone, but no partial satisfaction by achieving On(A,B) alone. In this case the two formulas might be:

$$\phi = \text{On(B, C)} \qquad \psi = \neg\text{On(B, C)}.$$

Let holds(ϕ, t) indicate that formula ϕ is true at time t. Suppose that for plan $\mathcal{P}_1$, we can find some time point $l \geq t_d$ such that:

$$\mathbf{P}(\forall t' \ (l \leq t' \leq n) \rightarrow \text{holds}(\phi, t')|\mathcal{P}_1) \geq \alpha.$$

For plan $\mathcal{P}_2$, suppose we can find some time point $m \geq t_d$ such that:

$$\mathbf{P}(\forall t' \ (1 \leq t' < m) \rightarrow \text{holds}(\psi, t')|\mathcal{P}_2) \geq \beta.$$

Under what conditions can we say that plan $\mathcal{P}_1$ is preferable to plan $\mathcal{P}_2$? We must determine the minimum attainable value of $\mathsf{EU}(\mathcal{P}_1)$ and the maximum attainable value of $\mathsf{EU}(\mathcal{P}_2)$ consistent with these two constraints. Let σ_ϕ be the formula of lowest $u(\sigma_i)$ consistent with ϕ (in the example, $\sigma_\phi = \sigma_2$). We are guaranteed the existence of such a formula since the σ_i are exhaustive. The expected utility of plan $\mathcal{P}_1$ is minimized if:

1. With probability α, σ_ϕ is achieved at time l, and the goal is not partially achieved at any time earlier than l.
2. With probability $1 - \alpha$, the goal is completely unsatisfied.

So by Eq. (12) we have:

$$\mathsf{EU}(\mathcal{P}_1) \geq \alpha \cdot \sum_{i=l}^{n} k_i u(\sigma_\phi) \ + \ (1 - \alpha) \cdot 0.$$

Now let σ_ψ be the formula of *highest* $u(\sigma_i)$ consistent with ψ ($\sigma_\psi = \sigma_1$). The expected utility of plan $\mathcal{P}_2$ is highest if:

1. With probability β, σ_ψ is achieved by the deadline, and the goal is completely achieved immediately after time m.
2. With probability $1 - \beta$ the goal is completely satisfied at the deadline.

Again by Eq. (12):

$$\mathsf{EU}(\mathcal{P}_2) \leq \beta \left[\sum_{i=1}^{m} k_i u(\sigma_\psi) + \sum_{i=m+1}^{n} k_i \right] + \ (1 - \beta).$$

And we therefore know that $\mathcal{P}_1$'s expected utility is higher than $\mathcal{P}_2$'s if:

$$\alpha > \frac{\beta[\sum_{i=1}^{m} k_i u(\sigma_\psi) + \sum_{i=m+1}^{n} k_i] + (1-\beta)}{\sum_{i=l}^{n} k_i u(\sigma_\phi)}. \quad (17)$$

The values for σ_ϕ and σ_ψ and the times l and m determine the usefulness of our probability bounds. A time l close to the deadline and a ϕ that is inconsistent with low-utility σ_i values give us a high lower bound on $\mathsf{EU}(\mathcal{P}_1)$. Similarly, a time m well after the deadline and a ψ inconsistent with high utility σ_i values give us a low upper bound on $\mathsf{EU}(\mathcal{P}_2)$.

Additional structure in the static component can make dominance relationships easier to derive. For example, we might consider a common static component that has an additional structural feature: an ordered conjunction of expressions $g_1 \wedge g_2 \wedge \cdots \wedge g_n$ in which each conjunct dominates each subsequent conjunct. Thus any outcome that satisfies g_1 is preferable to *every* outcome that does not satisfy g_1, even if it satisfies all the others. For example, if our conjuncts are g_1, g_2, and g_3, our strictly ordered static utility function would meet the constraint that satisfying g_1 is preferable to not satisfying it, regardless of whether g_2 or g_3 is satisfied. In this case two plans can be compared incrementally: first on their respective abilities to achieve g_1, then on the basis of g_2, and so on. The ability to compare plans incrementally in this way can result in computational savings when establishing dominance relationships.

6.1.2. Maintenance Goals. Relationships between two plans that try to achieve maintenance goals can be developed in a similar manner. Haddawy and Hanks (1993) performs an analysis for maintenance goals—those in which the temporal component requires that the static component hold over an interval. The general relationship established is between: (1) a plan $\mathcal{P}_1$ that is likely to keep a condition ϕ true over a long subinterval of the maintenance interval, where ϕ has a high $u(a_i)$ value, and (2) a plan $\mathcal{P}_2$ that is likely to keep a condition ψ true over a long subinterval of the maintenance interval, where ψ has a low $u(a_i)$ value. In this case, if ϕ has sufficiently high utility compared to ψ, and if the intervals over which ϕ and ψ are maintained are sufficiently long, and if $\mathcal{P}_1$ and $\mathcal{P}_2$ are sufficiently likely to achieve ϕ and ψ, respectively, then we can prove that $\mathcal{P}_1$ dominates $\mathcal{P}_2$. The result is a formula analogous to Eq. (17), relating the relative value of ϕ versus ψ and the amount of time each formula is maintained.

6.2. Least-Commitment Optimal Planning: PYRRHUS

Several factors limit the ability of classical planning algorithms to produce "good" plans (as measured by a utility model like the one we propose). Of these factors, two are primary. First is the inability to reason about the agent's uncertainty as to the state of the world and the effects of its actions. Second is the inability to rank plans on the basis of the consumption and production of resources, which is how we model residual utility. Generally, these algorithms employ a *satisficing* success criterion, searching for any plan that provably achieves the goal.

The PYRRHUS planner (Williamson and Hanks 1994; Williamson 1996) addresses the problem of building *optimal* plans, using a restricted form of the utility model developed in this paper. PYRRHUS, like classical planners, assumes certainty in the world state and in the effects of actions. It extends the UCPOP generative planner (Penberthy and Weld 1992) to handle quantitative (integer or real-valued) attributes, like fuel level or monetary position. Action preconditions can be expressed in terms of equality and inequality constraints on these as well as symbolic attributes; actions can produce or consume specific amounts of these quantitative attributes. For example, taking a trip from point A to point B might require the vehicle to be in working order (a symbolic attribute), cause the vehicle to be at point

B (another symbolic attribute), consume five gallons of fuel, and cost $10 in tolls (both quantitative attributes). Refueling the vehicle consumes one quantitative attribute (money) but produces another (fuel).

Using this extended action representation, PYRRHUS can build a plan that is optimal according to a restricted form of our utility model. PYRRHUS allows goals to have a symbolic static content and a temporal deadline. Preferences over the consumption and production of resources are expressed by identifying certain quantitative attributes as resources, then providing a function that maps the joint level of these resources into a numeric value representing preferences over the joint asset position. (This function can be unbounded if the domain offers an opportunity to generate net wealth.)

The question PYRRHUS addresses is whether the resulting optimization problem can be solved efficiently using classical planning techniques. More specifically, PYRRHUS addresses whether structure in the utility model can be exploited to find an optimal plan without searching the entire space of possible solution plans. An analogous question is whether heuristic search-control knowledge for the classical satisficing problem (i.e., a planning problem in which there is a symbolic goal, no temporal deadline, and zero-cost resources) can be effectively employed to solve the extended optimization problem.

PYRRHUS uses a combination of plan-space search and branch-and-bound pruning to solve the planning problem. Classical plan-space search views the problem as a search through a space of *partial plans*, each a partially ordered and partially instantiated set of actions. The initial plan contains no steps. At each stage of the search, more structure (i.e., new actions, new step orderings, new variable bindings) is added to the plan. The search terminates when a plan is found that satisfies the goal (a *complete* plan). Heuristic search-control knowledge is brought to bear at each iteration of the search to choose new structure that will quickly lead to a complete plan.

PYRRHUS uses this basic algorithm along with heuristic search-control knowledge developed for classical goal-satisfaction problems in a transportation planning domain. But PYRRHUS cannot stop whenever it finds a plan that satisfies the goal, because there might be a better plan—one that satisfies the goal closer to the deadline or at a lower cost—in an unexplored region of the search space.

The concern then becomes whether significant parts of the search space can be pruned without exploring them exhaustively. This amounts to the question of whether a significant number of partial plans can be rejected as suboptimal without refining them completely. PYRRHUS evaluates a partial plan by computing upper and lower bounds on its value, representing the minimum and maximum values of any completion of that plan. An upper bound can be placed on the value of satisfying each goal (in that the earliest possible time point at which the goal could be satisfied can be determined) and on the value associated with resource consumption and production (in that bounds can be placed on the net consumption of every resource in all completions of the plan). The bounds are then compared to the value of the current best alternative. If the new alternative's upper bound is less than the current best lower bound, the partial plan can be rejected; if the lower bound is greater than the current best upper bound, the partial plan becomes the current best alternative. This is the deterministic equivalent of applying Eq. (17) to establish dominance relationships between partial plans. These bounds also can be used for heuristic guidance of the search, biasing the search in favor of alternatives with promising lower bounds.

Two interesting empirical questions arise: (1) Do classical goal-oriented search-control heuristics provide significant guidance to the extended optimization problem? and (2) Does the branch-and-bound algorithm actually allow sufficient pruning of the search space to make the optimization problem feasible to solve?

The analysis in Williamson and Hanks (1994) used a suite of 13 goal-satisfaction problems

in a cargo-delivery domain; each of these problems generates a class of optimization problems for various deadlines, partial-satisfaction functions for deadlines, consumption functions for resources, and relative weighting factors among the goal and resource attributes. Six search-control heuristics were developed for the goal-satisfaction problem that allowed all 13 problems to be solved efficiently.[10]

Experiments in Williamson and Hanks (1994) demonstrated that performance on the optimization problem clearly benefited from classical goal-oriented search-control rules. In fact, the speedup in the optimization case was generally more dramatic than in the classical goal-satisfaction problem. Intuitively, this is because effective search control led quickly to a plan that satisfied the goal (and thus had a high lower utility bound), which became a good benchmark against which other candidates could be compared. Unpromising candidates could then be rapidly pruned.

The numeric parameters associated with a particular instance of the optimization problem determined whether the optimization problem was easier or harder than the classical goal-satisfaction problem. One interesting qualitative property we discovered was that the optimization problem turned out to be significantly *easier* to solve than the satisficing problem when the temporal deadline was set early; this meant that all feasible solutions to the problem would miss the deadline to some degree. The optimization problem was consistently *harder* to solve than the satisficing problem when the temporal deadline was set late, which meant that *any* minimal feasible solution to the problem met the deadline. In less extreme cases the time to solve the problem varied significantly with the particular numeric parameters. See Williamson and Hanks (1994) for a more detailed analysis and Williamson and Hanks (1996) for a discussion of heuristic search-control techniques common among classical plan-space planners which can be used to produce optimal plans as well.

We can conclude that an optimal course of action can be found efficiently, at least in some interesting cases. Furthermore, the ability to generate optimal plans depends *crucially* on the structure of the utility model. This structure is exploited in two phases of the planning process. First, the symbolic expression representing complete satisfaction of the static component is used in the plan-generation process to choose appropriate actions to insert into the plan (using the classical "backchaining" technique for plan generation). Second, the numeric attributes and associated parameters are used to compute bounds on a plan's value, thus allowing unpromising partial plans to be pruned and promising partial plans to be favored. Neither of these algorithmic techniques could be employed without a utility model that made this structure explicit.

6.3. Efficient Refinement Planning under Uncertainty: DRIPS and Qualitative Filtering

This section presents quantitative and qualitative ways to eliminate suboptimal classes of plans without explicitly examining all plans in that class. It also describes how these techniques exploit the structure of the utility function to facilitate computation. While qualitative techniques can be used to filter out classes of obviously bad plans, at some point one must resort to quantitative reasoning about expected utility in order to evaluate tradeoffs. We first present the DRIPS decision-theoretic refinement planning system (Haddawy and Suwandi, 1994; Haddawy, Doan, & Goodwin, 1995) which uses hierarchical abstraction to reduce the computational cost of quantitatively reasoning about plans. We then describe qualitative techniques that can be used as a filter to reduce the space of plans that DRIPS must search.

[10] The time required to solve the problem increased linearly with the number of steps in the solution plan.

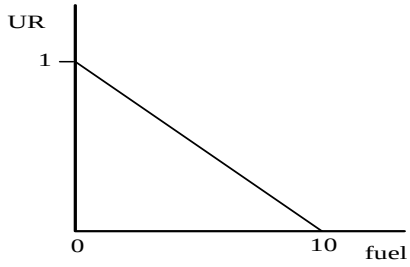

FIGURE 3. Specification of residual utility (UR) function.

6.3.1. Decision-Theoretic Refinement Planning. The DRIPS planner is capable of finding the optimal plan under uncertainty for the deadline and maintenance goal utility functions presented in this paper, as well as for arbitrary nontemporal utility functions. DRIPS searches through the space of possible plans by using action descriptions organized into an abstraction/decomposition network. The planner builds abstract plans, compares them, and refines only those plans that might be refinable to the optimal plan.

We describe DRIPS by demonstrating how it handles a sample problem. Suppose we want to find the best plan for delivering four tons of tomatoes from a farm to a warehouse within 240 minutes, with a deadline horizon of 280 minutes. We use the UG function from Section 5.1.4 and take the top-level utility function to be

$$\mathbf{U}(c) = \mathsf{UG}(c) + (.02)\mathsf{UR}(c),$$

where UR is a function of the amount of fuel consumed. The UR function is shown in Figure 3.

The delivery plan consists of driving a truck from the depot to the farm, repeatedly loading the truck, driving the loaded truck to the warehouse, and returning to the farm as many times as necessary. The truck's capacity is two tons. Descriptions of the available actions are shown in Figure 4. Each action is described with a tree structure, in which the first-level branches are labeled with conditions, the second-level branches are labeled with probabilities, and the leaves are labeled with the effects of the action. Each branch represents a conditional probability: the probability of the effect given the action and the condition. The conditions on the branches are required to be mutually exclusive and exhaustive, and the probabilities for any condition must sum to one. An action description need not include probabilities or conditions. If the probability is omitted, it is taken to be 1, and if the condition is omitted, it is taken to be true. Deterministic actions are labeled with a single outcome.

The conditions on the branches specify conditions that must be true just before the action is executed. The effects of the action are specified in terms of the action's duration and the attributes that it changes. For example, in the top branch of the "Go to farm on road B" action, $time = time + 15$ indicates that the duration is 15, and $fuel = fuel - 1/2$ means that the fuel level after the action is one half gallon less than before it. We assume that the only changes to the world are those explicitly mentioned in the action descriptions. Thus, in this particular branch the world remains in its state prior to the action until $time + 15$. Our action representation is similar to that introduced by Hanks (1990b).

There are two possible routes we can take from the depot to the farm: road A and road B. Road A is longer but has no delays, while travel along road B might be delayed due to

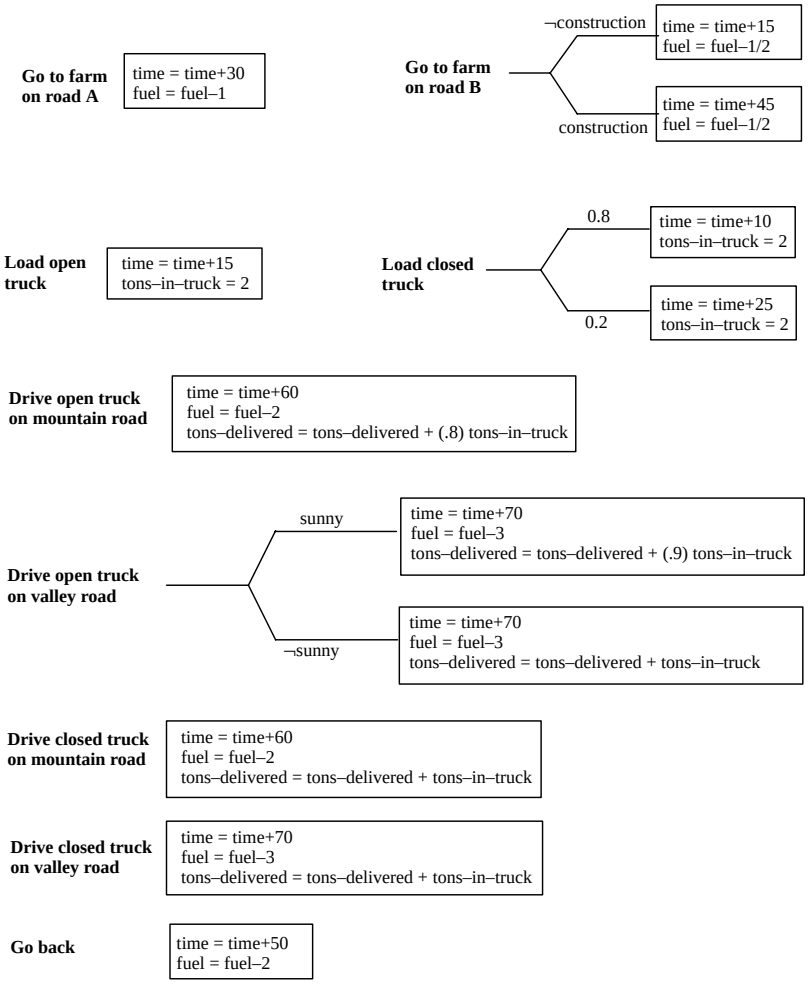

FIGURE 4. Action descriptions for the delivery example.

construction. The probability that construction is taking place is 0.2. These options are represented by the first two action descriptions in Figure 4.

Once at the farm we must load the truck. We have a choice between two trucks at the depot: an open truck and a closed, cushioned truck. The open truck is easy to load; there is an 80% chance that the closed truck can be loaded quickly and a 20% chance that loading it will take longer. The next two diagrams in Figure 4 depict these actions.

Once the truck is loaded, we must drive it to the warehouse. We have a choice between two routes: the mountain road and the valley road. The mountain road is shorter but bumpy. If we drive the open truck on the mountain road, the bottom 20% of the tomatoes will be crushed. If we drive the open truck on the valley road and the sun is shining, the top 10% of the tomatoes will be spoiled by the sun. This combination of options results in the next four action descriptions in the figure. Finally, the action of returning to the farm is depicted at the bottom of Figure 4.

The DRIPS system searches for the optimal plan using an abstraction/decomposition net-

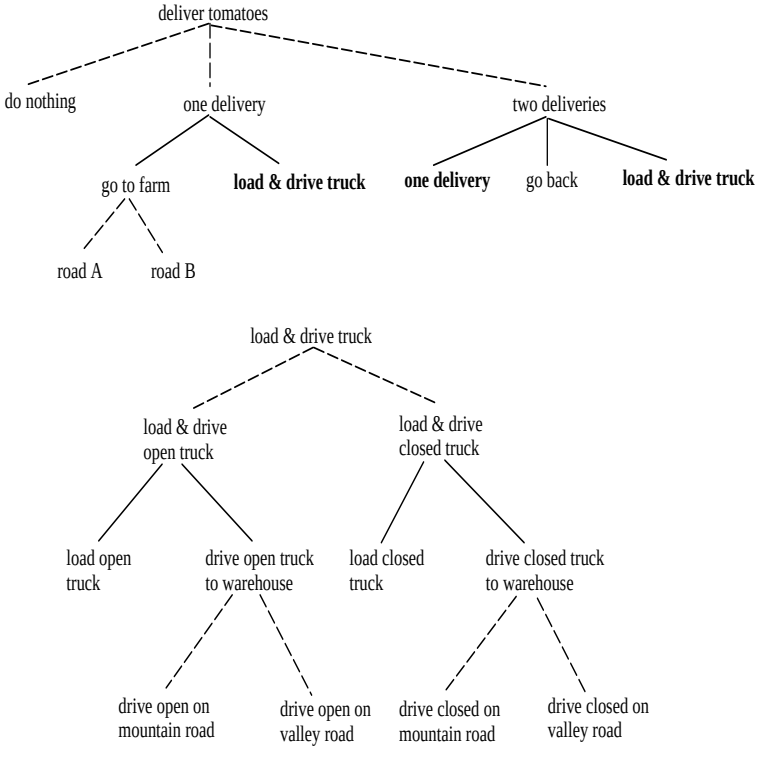

FIGURE 5. Abstraction/decomposition network. Abstraction relations are shown with dashed lines, and decomposition relations are shown with solid lines. Actions shown in bold have decompositions or abstractions that are displayed elsewhere in the figure.

work that describes the space of possible plans and their abstractions. The network for this problem is shown in Figure 5. An abstract action has one or more subactions, which themselves may be abstractions or primitive actions. A decomposable action has one or more subplans that must all be executed in sequence. For example, the task "deliver tomatoes" can be instantiated into any of the three lower level plans: "do nothing," "one delivery," and "two deliveries." The decomposable action "one delivery" consists of the sequence "go to farm" then "load & drive truck."

The abstract actions used by DRIPS are formed by grouping together a set of analogous actions, which are considered to be alternative ways of realizing the abstract action. The set is characterized by the features common to all the actions in it. We can then plan with the abstract action and infer properties of a plan involving any of its instances. We call this type of abstraction *interaction abstraction*. For an exposition of our theory of action abstraction see Doan and Haddawy (1996).

Given this representation of the planning problem, we evaluate plans at the abstract level, eliminate suboptimal plans, then further refine remaining candidate plans. An abstract plan's outcomes are sets of outcomes of more concrete plans. Since different probability and utility values can be associated with each specific outcome, in general a probability range and a utility range will be associated with each abstract outcome. Thus, the expected utility of an

abstract plan is represented by an interval that includes the expected utilities of all possible instantiations of that abstract plan. Refining the plan, that is, choosing an instantiation, can only narrow the interval. We can stop refining an abstract plan when its upper expected utility bound is less than the lower bound of some other plan.

Forty-one possible plans are implicitly encoded in the abstraction/decomposition network, and we want to choose the one that maximizes expected utility. DRIPS explicitly computes the expected utility of 12 plans, thus giving a pruning rate of 71%, and returns the plan "go on road B," "load closed," "drive closed on mountain," "go back," "load closed," "drive closed on mountain" as the optimal plan. Although performance of the planner on this small example problem is not overwhelmingly impressive, the efficiency of the planner tends to increase with an increase in problem size. For example, in a medical domain containing over 6,000 possible plans, we obtained a pruning rate of 89%, and the planner significantly outperformed a standard branch-and-bound algorithm (Haddawy, Doan, & Kahn 1996).

To evaluate plans, DRIPS must compute the expected utilities of all outcomes resulting from projecting the plans. This computation is facilitated by the structure of the utility function. In DRIPS an outcome is represented by a series of states in chronological order, specifying changes caused by actions. In the delivery example, we specified an outcome in terms of the amount of fuel in the tank, tons of tomatoes in the truck, and the total tons of tomatoes delivered at each point in time. Since the only changes to the world are due to the agent's actions, we can compute the value of the static attributes (current number of tons delivered) in the UG function by taking the difference between the current value of *tons-delivered* and its previous value. For example, we could have an outcome specified by *tons-delivered* = 1.8 at time 235 and *tons-delivered* = 3.8 at time 260. This means the stream of deliveries consists of 1.8 tons delivered at timepoint 240 (deliveries before the deadline are counted as occurring at the deadline); $3.8 - 1.8 = 2$ tons delivered at timepoint 260; and, for all other timepoints from 240 to 280, zero tons delivered. We can simply compute the utility by computing the contributions to utility of the two times and adding them.

6.3.2. Qualitative Filtering. For two cumulative probability distributions F and G, we say that F *stochastically dominates* G if for any given value b we have $F(b) \leq G(b)$.[11] An equivalent definition is that for all monotonically increasing functions ϕ

$$\int \phi(b)\, dF(b) \geq \int \phi(b)\, dG(b).$$

The above dominance is referred to in the literature as *first-order stochastic dominance* (Whitmore and Findlay 1978).

Consider two plans B_1C and B_2C, where B_1, B_2, and C are action sequences. Projecting B_1 produces a cumulative probability distribution F over the space of utility values (replacing each outcome resulting from executing B_1 with its utility value). Projecting B_2 yields G. Now suppose that projecting C on the outcomes that result from projecting B_1 and B_2 acts as a monotonically increasing function on the utilities of those outcomes, and thus is order-preserving. It follows from the definition of first-order stochastic dominance that if F stochastically dominates G, then the cumulative probability distribution resulting from projecting B_1C dominates the distribution resulting from projecting B_2C. Consequently, the second plan has a lower expected utility value than the first and can be eliminated from further consideration.

Unfortunately, action sequence C is unlikely to be order-preserving in most planning domains. In other words, given that $u(c_1) \geq u(c_2)$, where c_1 and c_2 are two outcomes

[11] Intuitively, this means that F puts more probability on larger values of b.

resulting from projecting B_1 or B_2, it is unlikely that C will transform c_1 (c_2) into c_1' (c_2') such that $u(c_1') \geq u(c_2')$. We can, however, expect that C will be order-preserving for a subset of the outcome space, such as the sets $D = \bigcup_i \langle X_i, Y_0 \rangle$, where $\langle X_i, Y_0 \rangle$ denotes an outcome whose utility attributes are XY, and attribute Y is fixed at value Y_0. It is easy to see that if C is order-preserving for X, and X is preferentially independent of Y, then C is order-preserving for D. This observation can be easily exploited in an algorithmic fashion to yield efficient qualitative filtering. We demonstrate this by discussing how it can be exploited in the delivery example.

A brief inspection of the problem description shows that any plan involving more than two deliveries is not optimal, since any delivery other than the first two would always miss the deadline horizon. Consider now the two plans "go to farm" and "do nothing." The first is dominated by the second, since it consumes more fuel for the same amount of tons delivered (0). Knowing that fuel is preferentially independent of tons delivered, and that less fuel is preferred to more fuel, is sufficient to establish the dominance relation. In a similar fashion we could rule out many other obviously nonsensical plans. This was done implicitly when we constructed the abstraction/decomposition network in Figure 5.

Consider two actions, B_1 "drive closed on mountain road" and B_2 "drive closed on valley road." They are analogous actions, and from any plan $p_1 = AB_1C$ we can construct plan $p_2 = AB_2C$, where A and C are action sequences. Let P_1 and P_2 be the probability distributions (over the outcome space) resulting from projecting AB_1 and AB_2, respectively. For each outcome o in P_1, there is an outcome o' in P_2 with the same probability such that fuel consumption in o' is greater than in o and streams of deliveries in both outcomes are identical, but deliveries occur later in o' than in o. Since earlier deliveries are preferred to later ones, the utility of o is greater than that of o'. This inference is sanctioned by the mutual utility independence of residual and goal utility and by the properties of the deadline utility function. Notice that we need not know the exact utility function to make this inference. The structure of the top-level and goal utility functions — along with the knowledge that we prefer less fuel consumption to more, and larger deliveries to smaller — is sufficient. From the preceding analysis, we can conclude that P_1 dominates P_2.

Whatever action sequence C represents will be applied in a similar fashion to both AB_1 and AB_2, since conditions for actions in C do not depend on anything that can be changed by AB_1 or AB_2. Applying the transformation represented by C to o of P_1 and o' of P_2, we obtain o'' and o''', respectively. Since all effects in our action descriptions are additive, no matter what action sequence C contains, it will transform c and o' in a linear fashion and will therefore be order-preserving for fuel and time. Also o'' and o''' will contain the same stream of deliveries, modulo differences in delivery times. So the utility of o'' is greater than that of o'''. It follows that the probability distribution resulting from projecting p_1 dominates the one resulting from projecting p_2; and the expected utility of p_1 is greater than that of p_2. Hence, the action "drive closed on valley road" can be eliminated from the domain. In a similar fashion the action sequence "load open truck" then "drive open on valley road" is always dominated by the sequence "load closed truck" then "drive closed on mountain road" and can therefore be eliminated from further consideration. The domain consists now of 13 plans, instead of the original 41 plans. The new filtered abstraction/decomposition network is shown in Figure 6.

This reasoning process can be easily done in an algorithmic fashion, exploiting utility and domain regularities to perform filtering. We are currently working on developing a comprehensive algebra of filtering similar to that of Wellman (1990).

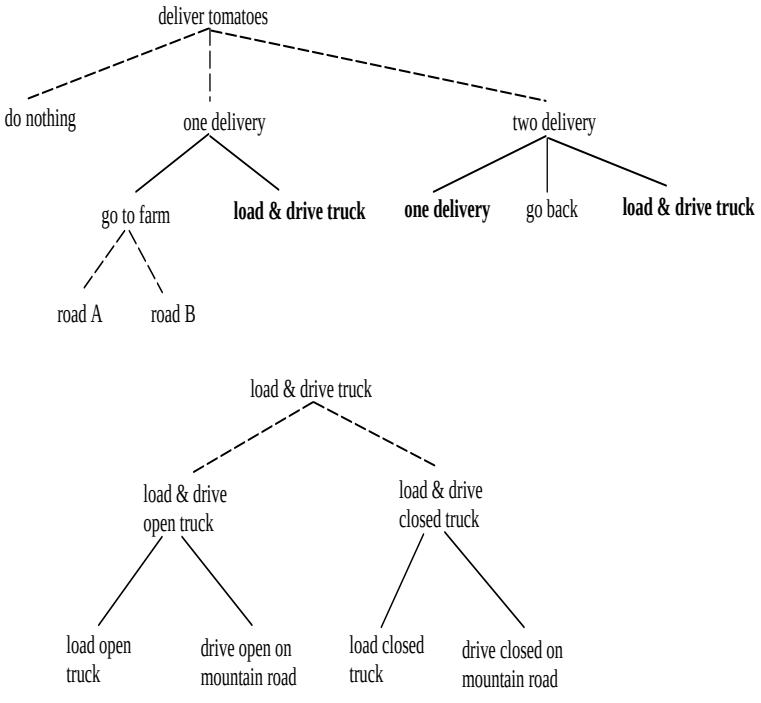

FIGURE 6. Abstraction/decomposition network after filtering.

7. VALIDATION ISSUES AND SUMMARY

Having described the proposed model, we now confront the question of how to evaluate it. Our main claim is that the proposed model provides a significant extension to classical goal models, while at the same time allowing effective assessment and generation of plans. This section makes our claims explicit and defends them. In doing so we also summarize the structure and assumptions underlying our model. We make the following five claims about the proposed model, which we discuss in turn:

1. The proposed model provides a significant increase in expressive power over preference models based only on the satisfaction or nonsatisfaction of symbolic goals.
2. The independence assumptions we make about the agent's preferences are reasonable.
3. Given these assumptions, the agent's preferences can be represented by a utility function of the form we developed.
4. The parameters needed to assess this resulting model can reasonably be obtained.
5. The resulting model can be used effectively by plan-generation algorithms.

The first and second claims are inherently subjective. We noted in Section 2 our three extensions to goal expressions:

- We added a temporal component to goals, permitting the decision maker to express preferences about *when* the goal is to be satisfied.

- We represented partial satisfaction of goals, allowing the decision maker to express preferences over outcomes in which the goal is not fully achieved. The model allows reasoning both about partial satisfaction of the goal expression itself (e.g., numeric goals that are achieved only to some partial level or symbolic subgoals, only some of which are achieved) and the goal's temporal component (e.g., soft deadlines).
- We allowed reasoning about resource attributes, permitting the decision maker to balance the cost of achieving the goal against the benefit of achieving it.

The second claim argues that the assumptions about the decision maker's preferences made in Sections 4 and 5 are reasonable. These assumptions follow:

- The decision maker's preferences over plan outcomes are fully specified by n top-level goals and by a set of resource attributes that describe how efficiently these goals are achieved.
- The decision maker's preferences over his top-level goals and over the production and consumption of resources are all mutually utility independent. In effect, this means that the level at which goal i is satisfied does not affect preferences over partial satisfaction of goal $j \neq i$, and further that preferences of partial satisfaction of goals do not depend on the consumption or production of resources. In making this assumption—essentially that preferences among levels of satisfaction for different goals are only weakly interdependent—we made the distinction between top-level goals and interacting subgoals and provided a mechanism for representing the interaction among components of a single goal.
- For each goal, preferences over the sequence of static attribute values generated by a plan are mutually utility independent over time. This assumption essentially means that preferences over *when* a goal is (partially) achieved can be considered separate from preferences over the *extent to which* the goal is achieved. We noted that in some cases (e.g., for aspiration-level goals) this assumption does not hold and we discussed techniques for transforming the attributes into a form in which MUI does hold.

The question of whether these assumptions are reasonable is subjective and depends on the particular decision maker and the particular problem. Sections 4 and 5 discussed the independence assumptions in some detail, showing how the model could represent common situations, like interacting subgoals. It is also important to keep in mind that *determining* whether the assumptions hold is a crucial part of the modeling process.

The third claim—that the decision maker's preferences can be represented by a particular utility function—follows directly from the independence assumptions noted above. Our analysis builds on results in Keeney and Raiffa (1976) which establish, for example, a correspondence between mutually utility independent preferences for the values of the goal's static component over time and a utility function that decomposes into a static utility function and a set of temporal weighting coefficients.

Fourth, we claim that the utility function so defined can be reasonably assessed. Recall the parameters of the model:

- two functions for each goal, describing preferences over temporal and static satisfaction
- one or more functions for the resource attributes, describing preferences over production and consumption of resources
- $n + 1$ numeric parameters, describing the relative value of the n goals and the value of goal satisfaction relative to resource consumption.

Once again, this claim must be evaluated subjectively. We should note, however, that the model's parameters are determined by the independence assumptions. So an objection that the model is too complex must be met with an investigation into what additional structure in the decision maker's preferences can be exploited (and what expressive power must be sacrificed) to further reduce the parameters required to complete the model.

The last claim is empirical—that planning algorithms will be able to exploit the structure in the model to build optimal plans efficiently. To substantiate this claim, we presented two implemented algorithms. The first extended classical least-commitment planning methods to handle soft deadlines and resource consumption, but it did not handle uncertainty. The second implemented the full decision-theoretic semantics of the model but used a more restricted definition of plan-generation: that of searching a space of possible refinements to abstract planning operators. In both cases we presented preliminary evidence that the planning process could be conducted efficiently and further showed that the ability to exploit structure in the model was crucial to doing so.

In conclusion, we have revealed a wide spectrum of utility models for planning problems, varying in complexity and structure. At one end we could adopt as a utility measure an arbitrary function over chronicles—an unstructured function with arbitrary expressive power. At the other end of the spectrum is the simple goal expression—a function with considerable structure that can be exploited in the planning process, but one that captures a limited class of preferences. We chose a point nearer the middle of the spectrum, trying to address some commonly accepted limitations of goals while maintaining the possibility of computational leverage. Even if our structural assumptions are inappropriate for a particular problem domain, there is value in understanding the trade-offs involved and thus how a more appropriate structure might be built.

8. RELATED WORK

A discussion of related work should begin with multiattribute utility theory, especially Keeney and Raiffa (1976). We have built a multiattribute utility model for goal-oriented planning problems that feature partial goal satisfaction and deadlines.

8.1. Goals and Utility Models

In the AI literature the work closest to our own is that of Wellman and Doyle (1991, 1992), which also analyzes the relationship between goals and preference structures. Their work confronts the question of what it means to say that an agent has some goal. They take the stance that relative preference over possible results of a plan constitutes the fundamental concept underlying the objectives of planning. However, they observe that working with unconstrained utility functions incurs too great a computational expense. As a solution they propose providing goals with a preferential semantics that preserves the validity of some common goal operations performed in planning. This semantic account provides a criterion for verifying the design of goal-based planning strategies, with goals serving as a computationally useful heuristic. The formal semantics also provides a basis for integrating goals with other types of preference information.

In contrast, rather than trying to retain the traditional concept of goal, we extend the concept to incorporate some of the expressiveness of utility theory while retaining some of the heuristic value of goals. While Wellman and Doyle define a goal as a proposition that is preferred to its opposite, all other things being equal, we explore the idea of extending the definition of goal to allow partial satisfaction. We do this in a way that imposes structure on

the form of the utility function, which we show provides computational leverage and eases the burden of elicitation.

8.2. Extended Goal Forms in Classical Planning

A few efforts have been made in the classical planning literature to extend the form of goal expressions: Drummond (1989) introduces a form of maintenance goals, allowing the constraint that a proposition must remain true throughout execution of a plan. Vere (1983) implements a concept related to deadline goals: a temporal *window*, or interval within which an action must be executed. Dean, Firby, & Miller (1988) handle deadlines and actions with duration. None of these efforts incorporates uncertainty or partial satisfaction into the representation, nor do they consider partial satisfaction of the goal's static component.

8.3. Decision-Theoretic Planning and Control

Decision-theoretic techniques have been applied to the planning problem by Feldman and Sproull (1975) and Dean and Kanazawa (1989). They have also been applied to the problem of controlling reasoning—choosing among computational actions and actions that make physical changes to the world (Russell and Wefald 1991; Boddy 1991; Etzioni 1991; Horvitz, Cooper, & Heckerman 1989).

Of these, Etzioni's (1991) model is most similar to ours, admitting both partial satisfaction of goals and also the idea that the value of achieving a goal will tend to change over time. Both of these elements are supplied directly to his model in the form of three functions:

1. a function $i(g)$ that measures the "intrinsic value" of goal g
2. a function $d(s, g)$ that measures the extent to which goal g is satisfied in state s
3. a function $F(i(g)d(s, g), s)$ that measures the extent to which the benefit of goal g should be realized in state s.

The first two functions correspond roughly to our static component, the third to the temporal weighting coefficient. There is no analogue in his model to our discussion of maintenance goals. He makes the same assumption we do about the utility independence of top-level goals.

8.4. MDP Models for Decision-Theoretic Planning

A substantial body of work is emerging in the decision-theoretic planning literature that uses formal models and computational techniques based on the Markov decision process (see Puterman 1994 for an overview). Some recent examples in the AI literature are Koenig (1992), Dean et al. (1993), Boutilier and Dearden (1994), Boutilier and Puterman (1995), Dean and Lin (1995), and Tash and Russell (1994). Two assumptions made in this literature deserve attention here: full observability and a time-separable value function.

A large majority of the MDP work assumes full observability,[12] which in effect means that the agent is given immediate and perfect information about what state it is in every time it takes an action. Under this assumption, the solution plan (or *policy*) is a mapping from states[13] into actions. Algorithms for building these policies typically use some form of

[12] See Littman, Cassandra, and Kaelbling (1995), Parr and Russell (1995), and Boutilier and Poole (1996) for some exceptions in the AI literature.

[13] A state is a complete description of the world at a single point of time. In the notation used in this paper, it corresponds to a value being assigned to each static attribute.

dynamic programming: computing a globally optimal policy involves computing a policy that is optimal under the assumption that k actions are left to perform, then using that information to extend to the case where $k + 1$ actions are left. These techniques can be used to compute a fixed-horizon policy, where the plan is to consist of exactly N actions. Alternatively, they can be continued until the policy converges, which solves the infinite-horizon case in which the plan is to be executed indefinitely.

The benefit of full observability is that at each stage of computation the algorithm must iterate only over the set of all world states. If perfect run-time information is not available, the algorithm must consider the set of *probability distributions* over world states.

The representational framework developed in this paper is not limited to a particular model of observability, since it comments on the value or preference structure of the domain and not on the system's dynamics. We are currently exploring ways in which our representational framework could be applied to fully or partially observable MDP problems. In doing so we should note the difference between the preference model we developed and the model commonly adopted by work in the MDP literature. The latter assumes that a numeric value representing reward or cost can be represented by a triple of the form $v(s_i, a, s_j)$, the value associated with being in state s_i, and taking action a, which causes a transformation to state s_j. The expected value of executing a policy is then typically taken to be the sum of the values accrued by all the states visited during that execution. When the policy is to be executed over an infinite horizon, the rewards accrued in later stages are often discounted.

Thus, the maximum expected value of executing an infinite-horizon policy given a current state of s can be expressed as

$$v^*(s) = \sum_{i=1}^{N} \mathbf{P}(s_i|s, a^*(s))[v(s, a^*(s), s_i) + \gamma v^*(s_i)],$$

where N is the number of states, $a^*(s)$ is the action dictated by the policy to be taken when in state s, $\mathbf{P}(s_i|s, a^*(s))$ is the probability that executing that action will cause a transition to state s_i, $v(s, a^*(s), s_i)$ is the reward or cost accrued if that transition actually takes place, and $0 \leq \gamma < 1$ is a discount factor that diminishes the impact of actions taken far in the future.

Although there is no analogue in our framework to a discount factor, this model can capture many of the concepts we develop in our paper, such as rewards for achieving a goal and costs for consuming resources. Other concepts, most particularly temporal deadlines, aspiration levels, and maintenance intervals, are more difficult to represent, since they tend to be defined in terms of *sets* or *sequences* of states and not in terms of a single state.

Of course, the MDP state can often be reengineered to accommodate these sorts of goals. For a simple temporal deadline, the state can be augmented to store the total elapsed time of plan execution so far. For a simple maintenance interval, the state can be made to store the total amount of time the maintenance condition has held. For a delivery goal, the state can be made to store the sum of all the deliveries made to this point not in excess of the aspiration level. However, in each case the transformation can be awkward and lead to an explosion in the number of states and thus in the time required to compute a policy.

For cases where most of the value associated with action is accrued by achieving a goal, algorithms based on the model developed in this paper offer alternatives to dynamic-programming approaches that iterate over the entire state space. In this sense our aims are similar to Boutilier, Dearden, and Goldszmidt (1995b), though we are addressing the more specific case of goal-achievement planning. The relationship between problem structure and computational leverage is discussed more thoroughly in Boutilier, Dean, and Hanks (1995a).

8.5. Optimal Planning in the Classical Framework

The work by Perez (1995) also explores algorithms for generating high-quality plans within the classical planning paradigm and is thus similar to the PYRRHUS work discussed previously. Her work is devoted to the problem of learning control rules that will improve subsequent planning episodes. Its utility model resembles those mentioned above—the cost of the plan is evaluated as the sum of the costs of its component steps. Therefore, our prior comments about the difficulty of expressing deadlines and resource consumption trade-offs also apply.

8.6. Fuzzy Decision Theory

The notion of partially satisfied goals and their role in the decision-making process appears prominently in the literature on fuzzy mathematics and decision analysis. In particular, our notion of a degree of satisfaction function closely resembles a fuzzy-set membership function. The seminal paper in this area is Bellman and Zadeh (1980); also see the papers in Zimmerman, Zadeh, and Gaines (1984), of which the most relevant to this paper is Dubois and Prade (1984), who discuss the role of *aggregation operators* in the decision-making process. In the language of fuzzy-set theory, a goal may be expressed as a fuzzy set, and a plan's membership function with respect to that set indicates the extent to which the plan satisfies that goal. An aggregation operator combines membership functions for individual goals into an aggregate membership function, which is an indicator of global success; this is called the *decision set*. A decision maker then selects an alternative that is "strongly" a member of the decision set. Dubois and Prade categorize and analyze various aggregation functions. However, they do not address the computational issues associated with plan evaluation or generation.

Our analysis is similar to the efforts in fuzzy decision making in that it emphasizes the representation problems associated with expressing partial satisfaction of goals. Fuzzy sets may be a more appropriate representation than the static attribute utility function when the latter (a numeric function) cannot reasonably be assessed. Fuzzy-set methodology provides a way to incorporate vague satisfaction measures like "reasonably well satisfied," "utter failure," and "complete success" into a precise analysis. As such, it is complementary to our analysis.

9. CONCLUSION AND FUTURE WORK

Our purpose in this work was to take the concept of goals as they have been used in symbolic planning systems and simultaneously extend their functionality and recast the intuitions in a form that can be used effectively by a plan-generation algorithm.

Our framework involved building a utility model by first identifying the agent's top-level goals plus the residual attributes that measure resource consumption and production in service of those goals. The assumption of utility independence among these attributes means that their interactions can be summarized by $n + 1$ numeric parameters, representing the relative weights for the n goals and residual attributes.

We then extended the notion of a goal to one that involves both a temporal component (deadline or maintenance interval) and a static component (a formula to be achieved or an attribute value to be maximized). We discussed various forms for the static goal component: symbolic and numeric attributes, conjunctions of attributes, and ordered conjunctions.

For both deadline and maintenance goals the user supplies two components that describe preferences over partial-satisfaction scenarios: the static component utility function and the

temporal weighting coefficient. The former defines what it means to satisfy the goal's static component, either partially or fully. The latter indicates how utility declines as a function of missing the deadline or violating the maintenance interval.

We next showed how the model's information, both numeric and symbolic, could be exploited to compare plans: to decide whether one plan's expected utility was greater than another, or to generate bounds on the quality of partial plans so that one partial plan can be chosen over an alternative. The general form of these relationships was to consider a plan $\mathcal{P}_1$ that was likely to achieve at worst a high level of satisfaction, and a plan $\mathcal{P}_2$ that was likely to achieve at best a low level of satisfaction. The result was a function of the respective formulas, their likelihoods, and their times, ensuring that $\mathcal{P}_1$'s expected utility was greater than $\mathcal{P}_2$'s, regardless of the two plans' other effects.

Finally, we demonstrated how the utility model is exploited by the two existing planning algorithms, PYRRHUS, which extends classical least-commitment algorithms, and DRIPS, which plans by refining abstract actions.

The formal model can be extended to cover more types of goals. The representation in this paper covers only goals that mention facts, but goals can also refer to events. An example might be "flip the switch at noon." Goals mentioning events must be deadline goals, since it makes no sense to maintain an event over an interval of time. Deadline goals involving events could be represented similarly to goals with a symbolic static component. But the current definition must be changed, because we defined the utility model of the goal's static component in terms of formulas that hold at time points, whereas events occur over time intervals.

The most important area for future work is to incorporate our model into additional decision-theoretic planning algorithms. Candidates are the state-space planners based on Markov decision processes, probabilistic nonlinear planners, and influence-diagram-based planners.

ACKNOWLEDGMENTS

Thanks to Tony Barrett, Denise Draper, Vu Ha, Dan Weld, Mike Wellman, and Mike Williamson for many useful comments, to AnHai Doan for many helpful discussions and for providing the results on plan filtering, and to Sandy Kaplan for editing and proofreading. Haddawy is supported in part by NSF grant IRI–9509165 and by a grant from Rockwell International. Hanks is supported in part by NSF grant IRI–9523649 and in part by ARPA contract F30602-95-1-0024 administered by Rome Labs.

REFERENCES

ALLEN, J., H. KAUTZ, R. PELAVIN, and J. TENENBERG. 1991. Reasoning about Plans. Morgan Kaufmann, San Francisco.

BELL, D. E. 1977. A decision analysis of objectives for a forest pest problem. *In* Conflicting Objectives in Decisions. *Edited by* D. E. Bell, R. L. Keeney, and H. Raiffa. Wiley, New York, ch. 18, pp. 389–421.

BELLMAN, R. E. and L. A. ZADEH. 1980. Decision-making in a fuzzy environment. Management Science, **17**:B141–B164.

BODDY, M. 1991. Solving time-dependent problems: A decision-theoretic approach to planning in dynamic environments. PhD thesis, Brown University, Department of Computer Science, May.

BOUTILIER, C., T. DEAN, and S. HANKS. 1995. Decision theoretic planning: Structural assumptions and computational leverage. *In* Proceedings of the Second European Workshop on Planning.

BOUTILIER, C., and R. Dearden. 1994. Using abstractions for decision-theoretic planning with time constraints. *In* Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge, pp. 1016–1022.

BOUTILIER, C., R. DEARDEN, and M. GOLDSZMIDT. 1995. Exploiting structure in policy construction. *In* Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal. Morgan Kaufmann, San Francisco, pp. 1104–1111.

BOUTILIER, C., and D. POOLE. 1996. Computing optimal policies for partially observable decision processes using compact representations. *In* Proceedings of the Thirteenth National Conference on Artificial Intelligence. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge.

BOUTILIER, C., and M. L. PUTERMAN. 1995. Process-oriented planning and average-reward optimality. *In* Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal. Morgan Kaufmann, San Francisco, pp. 1096–1103.

DEAN, T., J. FIRBY, and D. MILLER. 1988. Hierarchical planning involving deadlines, travel times, and resources. Computational Intelligence, **4**(4):381–398.

DEAN, T., L. KAELBLING, J. KIRMAN, and A. NICHOLSON. 1993. Planning with deadlines in stochastic domains. *In* Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI, Washington, DC, pp. 574–579.

DEAN, T., and K. KANAZAWA. 1989. A model for projection and action. *In* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco, pp. 985–990.

DEAN, T., and S. H. LIN. 1995. Decomposition techniques for planning in stochastic domains. *In* Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco, pp. 1121–1127.

DOAN, A., and P. HADDAWY. 1996. Sound abstraction of probabilistic actions in the constraint mass assignment framework. *In* Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, pp. 228–235.

DRUMMOND, M. 1989. Situated control rules. *In* Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, San Francisco, pp. 103–113.

DUBOIS, D., and H. PRADE. 1984. Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory. *In* Fuzzy Sets and Decision Analysis. *Edited by* H. J. Zimmerman, L. A. Zadeh, and B. R. Gaines. North Holland, Amsterdam, pp. 209–240.

ETZIONI, O. 1991. Embedding decision-analytic control in a learning architecture. Artificial Intelligence, **1-3**(49):129–160.

FELDMAN, J. R., and R. F. SPROULL. 1975. Decision theory and artificial intelligence II: The hungry monkey. Cognitive Science, **1**:158–192.

HADDAWY, P. 1991. Representing plans under uncertainty: A logic of time, chance, and action. PhD thesis, University of Illinois (Report no. UIUCDCS-R-91-1719).

HADDAWY, P., A. DOAN, and R. GOODWIN. 1995. Efficient decision-theoretic planning: Techniques and empirical analysis. *In* Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, pp. 229–236.

HADDAWY, P., A. DOAN, and C. E. KAHN Jr. 1996. Decision-theoretic refinement planning in medical decision making: Management of acute deep venous thrombosis. Medical Decision Making, **16**(4):315–325.

HADDAWY, P., and S. HANKS. 1990. Issues in decision-theoretic planning: symbolic goals and numeric utilities. *In* DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control. Morgan Kaufmann, San Francisco.

HADDAWY, P., and S. HANKS. 1993. Utility models for goal-directed decision-theoretic planners. Technical Report 93-06-04, University of Washington, Department of Computer Science & Engineering, September. Available via anonymous FTP from `~ftp/pub/ai/cs.washington.edu`.

HADDAWY, P., and M. SUWANDI. 1994. Decision-theoretic refinement planning using inheritance abstraction. *In* Proceedings of the Second International Conference on AI Planning Systems. Morgan Kaufmann, San Francisco, pp. 266–271.

HANKS, S. 1990a. Practical temporal projection. *In* Proceedings of the Eighth National Conference on Artificial Intelligence. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge.

HANKS, S. 1990b. Projecting plans for uncertain worlds. PhD thesis, Yale University, Department of Computer Science, January.

HORVITZ, E. J., G. F. COOPER, and D. E. HECKERMAN. 1989. Reflection and action under scarce resources: Theoretical principles and empirical study. *In* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco, pp. 1121–1127.

KEENEY, R. L., and H. RAIFFA. 1976. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & Sons, New York.

KOENIG, S. 1992. Optimal probabilistic and decision-theoretic planning using Markovian decision theory. PhD thesis, Computer Science Division (EECS), UC Berkeley, May.

LITTMAN, M. L., A. R. CASSANDRA, and L. P. KAELBLING. 1995. Learning policies for partially observable environments: Scaling up. *In* Proceedings of the Twelfth International Conference on Machine Learning. Morgan Kaufmann, San Francisco.

MCDERMOTT, D. 1982. A temporal logic for reasoning about processes and plans. Cognitive Science, **6**:101–155.

PARR, R., and S. J. RUSSELL. 1995. Approximating optimal policies for partially observable stochastic domains. *In* Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco.

PEMBERTHY, J. S., and D. WELD. 1992. UCPOP: A sound, complete, partial order planner for ADL. *In* Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, pp. 103–114. Available via anonymous FTP from `~ftp/pub/ai/cs.washington.edu`.

PEREZ, A. 1995. Learning search control knowledge to improve plan quality. Technical Report CMU-CS-95-175, Carnegie-Mellon University, July.

PUTERMAN, M. L. 1994. Markov Decision Processes. John Wiley, New York.

RAIFFA, H. 1968. Decision Analysis. Addison-Wesley, Reading, MA.

RUSSELL, S. J., and E. H. WEFALD. 1991. Do the Right Thing: Studies in Limited Rationality. MIT Press, Cambridge, MA.

TASH, J., and S. RUSSELL. 1994. Control strategies for a stochastic planner. *In* Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge, pp. 1079–1085.

VERE, S. 1983. Planning in time: Windows and durations for activities and goals. IEEE Transactions on Pattern Analysis and Machine Intelligence, **5**(3):246–267.

VON NEUMANN, J., and O. MORGENSTERN. 1947. Theory of Games and Economic Behavior, 2nd ed. Princeton University Press, Princeton, N.J.

WELLMAN, M. P. 1990. Formulation of Tradeoffs in Planning Under Uncertainty. Pitman, London.

WELLMAN, M. P., and J. Doyle. 1991. Preferential semantics for goals. *In* Proceedings of the Ninth National Conference on Artificial Intelligence. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge.

WELLMAN, M. P., and J. Doyle. 1992. Modular utility representation for decision-theoretic planning. *In* Proceedings of the First International Conference on AI Planning Systems. Morgan Kaufmann, San Francisco.

WHITMORE, G. A., and M. C. FINDLAY. 1978. Stochastic Dominance. Lexington Books, Lexington, MA.

WILLIAMSON, M. 1996. A value-directed approach to planning. PhD thesis, University of Washington, Department of Computer Science and Engineering, June.

WILLIAMSON, M., and S. HANKS. 1994. Optimal planning with a goal-directed utility model. *In* Proceedings of the Second International Conference on AI Planniing Systems. Morgan Kaufmann, San Francisco.

WILLIAMSON, M., and S. HANKS. 1996. Flaw-selection strategies for value-directed planning. *In* Proceedings of the Third International Conference on AI Planning Systems. Morgan Kaufmann, San Francisco.

ZIMMERMAN, H., L. A. ZADEH, and B. R. GAINES. 1984. Fuzzy Sets and Decision Analysis. TIMS Studies in the Management Sciences, vol. 20. North Holland, Amsterdam.