STANFORD LOGIC GROUP
REPORT NO. LOGIC-87-11

January 1988

A Decision-Theoretic Approach to the Control of Planning Search

David E. Smith

Department of Computer Science
Stanford University
Stanford, CA 94305

Abstract

One impediment to the realization of effective planning systems has been the problem of controlling search. In this paper we lay the foundations for a decision-theoretic approach to the control of planning search. We assume that the planner has models available of the cost of achieving atomic goals and their negations. We also assume that it has models of the likelihood of being able to achieve such goals. Using this information, we show how a planner can choose between alternative actions for achieving goals and subgoals, can choose the order in which to plan for conjuncts in a conjunctive goal or subgoal, can decide when to make assumptions or insert conditionals into a plan, and can decide when to interleave planning with execution.

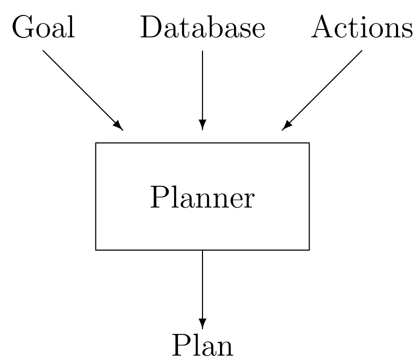

Figure 1: Planning

1 Introduction

One of the most difficult areas for A.I. has been that of building systems that can do synthesis tasks such as planning and design. Such systems are clearly important if we wish to have robots that can function autonomously in complex or remote environments, or systems that can design non-trivial electrical or mechanical devices.

By a planning system, we mean a program like that illustrated in Figure 1 that takes a statement of a goal or task to be performed, some description of the state of the world, and a description of the actions available. It then outputs some program of actions that will likely result in achieving the goal. In general, this program might include sensory actions, conditionals and loops. It might also be incomplete, in the sense that certain portions may require additional planning before they can be achieved.

There have been many attempts to build general purpose planning systems in AI. Some of the earliest and most well known systems are those of Green [11], Fikes and Nilsson [2, 3], Sacerdoti [17, 16, 18], Tate [27, 26], and Warren [31]. Unfortunately, none of these early planners proved very satisfactory for nontrivial problem tasks. One reason is that they suffered from fundamental epistemological deficiencies. For example, in most of these systems the notion of time was limited to the simple situation calculus, uncertainty or incompleteness could not be modelled, and there was no model of sensory actions, conditionals, or actions with conditional outcomes. These epistemological deficiencies are further reflected in basic architectural deficiencies of these early planners. For example, with no knowledge of uncertainty or incompleteness, these planners were designed to produce only complete plans that were guaranteed to succeed. As a result, these planners did not have to deal with more complex plan structures like conditionals and loops. In the last 10 years there has been a good deal of work in AI on these epistemological problems, and on developing planning procedures that deal with such things as uncertainty, sensory and conditional actions, and actions having derivable consequences.

There is however, a second, and somewhat more insidious reason that general purpose planning and synthesis systems have not been realized; that is the problem of *efficiency*. For any nontrivial planning problem, the search space of possible plans is absolutely enormous. First of all, for any given subgoal that a planner is considering, there can be many different operations that might achieve it. The majority of these will probably be unfruitful, because their preconditions are difficult or even impossible to satisfy. A second complication is that many of the subgoals that a planner encounters are likely to be conjunctions. If the planner chooses to work on the subgoals in the wrong order, a good deal of backtracking may be required in order to find a plan, and the plan that results may be very inefficient.

The addition of uncertainty or incompleteness to the system's world model adds additional complexity to the search space and the search process. For example, when should the planner postpone further work on a plan and allow execution to begin? When faced with uncertainty should the planner make a plausible assumption about the state of the world, should it introduce sensory actions and conditionals into the plan, or should it try to force the world into a known state?

1.1 Approaches to the Efficiency Problem

One approach to dealing with the efficiency problem in planning is to provide a library of skeletal or compiled plans for the goals expected by the system [3, 6]. To construct a plan a skeletal planner selects the appropriate skeletal plan, and then fills in all of the details.

If the number of goals and situations a planner is expected to face are relatively small it may be possible to provide a sufficient library of plans so that the planner will never have to resort to first-principles planning. However, if the planner must face a tremendous range of possible goals and situations (as in the commonsense world) the planner will eventually run into a situation not covered by a skeletal plan. Thus, while a large library of skeletal plans seems to be crucial for good performance, skeletal planning alone is not sufficient. For those problems where no skeletal or compiled plan exists we are left with the same nasty search control problem that confronted early general purpose planning systems

Several methods have been suggested for helping to control search in general purpose planning systems. Two of the most notable deal with controlling search for conjunctive goal expressions. One approach, taken by Sussman [25], Sacerdoti [17, 18], Wilkins [32], and others, is to plan for each conjunct separately, and then try to combine the partial plans in ways that ensure that they do not conflict with one another. A second important technique is hierarchical planning [16, 26, 23, 24], where different predicates are given priorities and those with higher priority receive planning attention before those with lower priority. The intent of this scheme is to ensure that those conjuncts critical to accomplishing a goal are processed before those that are relatively minor.

The trouble with both of these techniques is that they are brittle; in some cases they result in very inefficient plans, in other cases they do not help to control the search. The method of planning each conjunct separately and recombining the resulting plans is particularly susceptible to this problem; if the goals interact heavily, the planner may not be able to put the independent plans together, or may do so only in a very inefficient way. An example of this is travel planning problems having more than one destination. Assembling independent plans for getting to each destination could result in a good deal of wasted travel.

Hierarchical planning can also lead to inefficient plans, or may waste planning effort. The trouble is that it is not always possible to provide a single static hierarchy that will result in processing conjunctive goals in an efficient order. Thus, whether a particular conjunct is critical to accomplishing a conjunctive goal, or merely a minor detail, depends heavily on the state of the world. Both Sacerdoti [16] and Sproull [23] have remarked on this problem.

1.2 Approach

In this paper, we take a somewhat different approach to the problem of controlling search in planning. The basic idea is to include models of the cost of achieving different atomic goal expressions, and models of the chance these goal expressions can be achieved. Using this information, simple decision theory can be used to evaluate the potential utility of considering each of the different subgoals in the search space. This information can then be used to guide a best-first search procedure in looking for plans to achieve the overall goal of the planner.

As an example of the kind of information we require, consider the simple goal expression $At(x, l)$ of having a particular object at a particular location. In general, the cost of achieving such a goal depends on many factors: the weight and fragility of the object to be moved, the distance and terrain between the object and the intended location, how much stuff is piled on the object and the intended location, and, of course, the capabilities of the agent. Thus, we would not expect such a cost function to be just a simple distance or weight metric. Instead it might consist of a number of rules (i.e. implications) perhaps involving other complex concepts, such as the movability of an object, or the accessibility of a location.

One potential objection to this approach is that there is a vast amount of this kind of information that will be necessary, and, as a result, it will be tedious, or perhaps impractical to provide all of the necessary information. There are two answers to this objection. First of all, in our view, knowledge of the difficulty and chance of achieving different goals is a crucial part of commonsense knowledge about the world and about the actions available to an agent. For example, we know that it is much easier to get a pencil across a room than it is to get a truck to Japan, which is in turn much easier than getting something to Mars. Such knowledge seems to play a crucial role in the way people focus their planning activity.

A second answer to the above objection is that we do not believe that such knowledge must be entered by hand. Initially we intend to supply the cost and probability models by hand, however, it appears that techniques developed in [21] can be extended to allow a system to automatically derive cost and probability models for goal expressions from more basic cost information about the primitive actions available. Ultimately, machine learning of these cost and probability models may also be feasible, but this is outside the scope of this paper.

In some respects, the basic idea behind this approach is not novel; many people have used models of the value of different subgoals in a search space, and have used best-first search to make a choice among alternatives. In particular, Sproull [23] used this sort of decision theoretic model in the construction of a travel planning program. What is novel about this approach is

1. the way we estimate what the world will look like when certain goals must be achieved,

2. the way we determine the order in which to plan the conjuncts in a conjunctive goal,

3. the application of decision theoretic techniques to the problem of planning under uncertainty.

In the sections that follow we will describe the approach in greater detail and show how we believe these techniques can be applied to controlling search in planning. We start with the simple case, by supposing that the planner's model of the world is perfect. We also suppose that its model of the available actions is perfect. Under these assumptions, we first consider the case of choosing between alternative actions for achieving a top level atomic goal. We then expand the treatment to cover arbitrary conjunctive subgoals. Finally, we show how the same information can be used to automatically decide the order in which to plan for conjuncts in a conjunctive goal or subgoal.

In Section 3 we reintroduce uncertainty into the descriptions and show how this affects search control. Finally, we consider how these techniques can be used to decide whether or not to interleave planning with action, and to decide whether or not to insert tests and conditionals into plans.

2 The Certain World

To keep matters simple, we will first consider the case where the planner's model of the world is perfect, i.e. there is no uncertainty about the state of the world, or about the effects of actions on the world. In the next section, we will relax this requirement and consider the more realistic, but more complicated case where there is uncertainty about the state of the world, and about the effects of actions.

In order to talk about the control of planning, we first need to have a precise means of referring to actions and plans, and their associated costs.

For convenience, we will use the first order predicate calculus with the usual syntax and logical operators for describing the world. Any other language with sufficient expressive power would do as well. We will use q_s to refer to the proposition that q holds in the situation s. A situation s can be characterized as the set of sentences that hold at some particular instant in time. Note that we are not making a commitment to any particular temporal logic, since the set of sentences that hold at a given time can be defined in any reasonable temporal logic.

Primitive actions are taken to be individual operations that can be performed by an agent without further reduction. For example, moving an arm joint to a particular angle, or until a particular force is achieved might be primitive actions for a typical robot.

We will assume that the effects of actions are described by rules of the form:

$$q_1 \wedge q_2 \wedge \ldots \wedge q_j \wedge Performed(a)_s \Rightarrow c_1 \wedge c_2 \wedge \ldots \wedge c_k$$

where the premises q_i are normally dependent on s and the conclusions c_i are normally dependent on some situation s' corresponding to the time when the action is completed. We will refer to the q_i as a precondition set for the action a and c_i as a conclusion set for a. If there is more than one such axiom for a given action, there would be more than one such precondition and conclusion set. Normally when referring to the preconditions or consequences of an action it will be clear which set we are referring to from context. We will therefore use $Pre(a)$ and $Cons(a)$ to refer to the preconditions and consequences of an action a in the given context.

We will use the expression $C(a, s)$ to refer to the cost of performing a primitive action a in the situation s. This cost could be any convenient measure of time or other resources that might be required to perform the action. As noted in the previous section, we assume that this information is available for all primitive actions.

By a *plan*, we will mean a sequence of primitive actions. The expression $s|p$ will be used to refer to the situation that results when the plan p is performed in situation s. For a plan, p we define $C(p, s)$ recursively as follows:

$$C(ab, s) \equiv C(a, s) + C(b, s|a)$$

$B(q, s)$ will be used to denote the *best* plan for achieving a goal q in situation s. For the certain world, the best plan is simply the cheapest one. That is:

$$B(q, s) \equiv p \in Plans(q) : \min C(p, s).$$

By a *partial plan* we will mean an arbitrary set of constraints on the world. Thus a goal statement is a partial plan. Likewise, the specification of a partially ordered set of actions to be performed is also a partial plan.

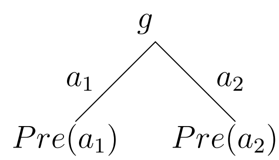

Figure 2: Simple choice between actions

Using the notion of a best plan, we can extend the definition of C to partial plans. For a partial plan q we let $C(q, s)$ refer to the expected cost of performing $B(q, s)$ (the best possible completion for q) in s. Formally,

$$C(q, s) \equiv C(B(q, s), s).$$

As noted in Section 1.2 we assume that $C(q, s)$ is available for all q that are atomic expressions or negations of atomic expressions.

In similar fashion, we extend the $|$ notation to partial plans. For a partial plan q we let $s|q$ refer to the situation that would result if $B(q, s)$ (the best possible completion of q) were performed in s. Formally,

$$s|q \equiv s|B(q, s).$$

2.1 Choosing Between Alternatives

Suppose that we have a planner working on a top-level goal g, and there are two different actions a_1 and a_2 that would allow the goal to be achieved, as illustrated in Figure 2. Which possibility should the planner choose? If both a_1 and a_2 are guaranteed to succeed, and all of their preconditions are satisfied in the world, the decision is a simple one; the planner should choose the action that is cheapest to perform. Thus, if $C(a_1, s) < C(a_2, s)$, the planner should prefer a_1, whereas, if $C(a_1, s) > C(a_2, s)$, the planner should prefer a_2. Since we are assuming that the planner can compute $C(a, s)$ for any primitive action a and situation s, it is straightforward to mechanize this comparison.

If the preconditions for an action a are not satisfied, the selection task is somewhat more difficult; we need to know both the cost of performing the action a and the cost of achieving the preconditions for the action. The cost of establishing the preconditions is given by $C(Pre(a), s)$. The situation that would result from establishing the preconditions is $s|Pre(a)$, so the cost of performing a in this new situation is given by $C(a, s|Pre(a))$. Putting these two costs together, the cost of achieving g using a is given by

$$C(Pre(a), s) + C(a, s|Pre(a)). \tag{1}$$

To compare two actions, a_1 and a_2, we would need to calculate this expression for both actions. However, unlike the simple case where the preconditions are already established, these expressions are not so easy to automatically compute. First consider the $C(a, s|pre(a))$ term. As before, we are assuming that the planner can compute $C(a, s)$ for any primitive action a and situation s. The trouble is, we don't yet know what the situation $s|Pre(a)$ looks like. In order to discover this, we would need to find the best plan for establishing $Pre(a)$ in s and then use this plan to predict the resulting state. Doing all of this planning would defeat the purpose of our cost analysis – to direct planning effort to only the most promising portions of the search space. We therefore need a computationally tractable way of estimating what $s|q$ looks like. If q is consistent with s it makes sense to just assume that $s|q \approx s \cup q$. But if q is not consistent with s we need to overturn those facts in s that are in conflict with q. We can use the possible world construction described in [8, 9, 10] for this purpose; we approximate $s|q$ by the nearest possible world to s in which q holds. Using this construction, it is possible to estimate $s|Pre(a)$, which allows us to compute $C(a, s|Pre(a))$.

The other term, $C(Pre(a), s)$ is equally troublesome. Recall that we are assuming the planner can compute $C(q, s)$ for any atomic proposition q. If $Pre(a)$ consists of only a single clause we are in business. Otherwise, we need some way of computing $C(Pre(a), s)$ from information about each of the atomic clauses in $Pre(a)$. This is the subject of the next section.

2.2 Conjunctions

Consider a conjunction e consisting of the conjuncts $e_1, \ldots, e_n$ and suppose that we are interested in knowing the expected cost of achieving the conjunction using the best possible plan. If none of the clauses in e interact with each other in any way, we could simply find the best plan for achieving each one in isolation, and then string those plans together. The cost of the resulting plan would be

$$C(e, s) = \sum_{c \in e} C(c, s).$$

Unfortunately, this kind of independence rarely holds; achieving one conjunct c often changes the world in such a way that some of the other conjuncts become either easier or more difficult. If we knew the order in which the conjuncts would be achieved we could take this effect into account in computing the cost of achieving the conjunction. Let the function $\tau(c)$ denote the set of conjuncts in e that will be achieved before c. Given such an ordering function τ, the cost of achieving the conjunction e would be given by

$$C(e, s) = \sum_{c \in e} C(c, s|\tau(c)). \tag{2}$$

Using the techniques introduced earlier for estimating $s|q$ we could use this equation to automatically compute the cost of achieving the conjunction e. There are, however, two important bugs with this approach. First, we have not shown how to find an appropriate temporal ordering τ for a set of conjuncts. In fact, in the case of non-serializable conjunctions there may be no such temporal ordering. We will postpone this issue until Section 2.4. For now, assume that we can find one. The second problem is that we have assumed that none of the conjuncts in e share any free variables. While this is sometimes true of conjunctive goals, it is by no means universally true. For example, consider a simple goal like "get B on a green object", which might be formalized as

$$On(B, x) \wedge Green(x)$$

In this case, the best possible plan would involve putting B on a nearby green object, or if there aren't any green objects, on a nearby object that is easy to paint. However, equation (2) would estimate the cost of achieving the conjunction as the cost of moving B to the nearest object that would support B, plus the cost of painting the easiest object to paint. This estimate might be very low, because the closest object that supports B might not be easy to paint, and the easiest object to paint might not support B, or might be far away.

To obtain the best plan for a conjunctive goal like this, a planner might need to try many different variable bindings for the shared variables. In practice this is often impractical because of he size of the search space. As a result, most planners plan for one conjunct, binding any free variables in the process, then plan for another, and so forth, only backtracking to consider alternative bindings when failing to find a plan for some later conjunct. If we knew this planning order for the clauses in a conjunction we could use this information to get a better estimate of the cost of plans that are likely to result from such planning procedures. Let the function $\varphi(c)$ denote the set of conjuncts in e that precede c in the planning order. Furthermore, let $c \backslash q$ refer to a new clause c' in which all of the variables shared with q have been bound to skolem constants. We can then write down an approximate equation for the cost of achieving a conjunction containing shared variables:

$$C(e, s) = \sum_{c \in e} C(c \backslash \varphi(c), s|\tau(c)). \tag{3}$$

Given φ and τ we could easily automate this computation. As with τ, we have not shown how to find a planning order φ for a conjunction. We will postpone consideration of this matter until Section 2.5. For now, assume that we have this information available.

It is interesting to note that while equation (2) provides a lower bound on the cost of achieving a conjunction, equation (3) provides an upper bound. If the planner is somehow smart, and chooses good bindings for the shared variables, it might be able to produce a plan that is better than the plan that would be achieved by binding the variables in a strict planning order. Normally, with a best first search procedure, a lower bound cost estimate would be preferred, because it retains admissibility of the search procedure. However, we speculate that the upper bound will generally prove to be more accurate, and hence will do

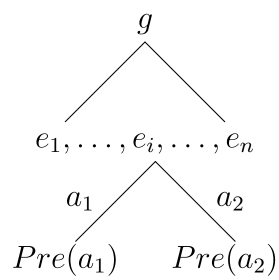

Figure 3: Choice between actions

a better job of controlling the search. This is an interesting empirical question that remains to be resolved.

One other interesting side issue raised by this analysis is the possibility of using cost information to help a planner choose good bindings for shared variables in a conjunction *prior* to the actual planning process for the individual conjuncts. Consider the $On(B, x) \wedge Green(x)$ example: Given rules for estimating the cost of achieving On, and rules for estimating the cost of achieving $Green$, the planner could form the cost expression that is the sum of the two. The planner could then search for an object x in its model of the world that would minimize this cost expression. After finding such an x it could then substitute the x into the conjunctive goal expression, and then plan for the individual conjuncts. The advantage of this approach is that the search for good bindings for the shared variables is done independently of the search for a good plan for achieving each conjunct. For conjunctive goals that share variables this may result in either a combinatorial savings, or more efficient plans.

2.3 Choosing Between Alternatives (Revisited)

In Section 2.1 we considered the problem of choosing between two alternative actions a_1 and a_2 for achieving a top level goal g. Using the analysis of the previous section, we can now extend the analysis from Section 2.1 to cover the choice between alternatives for goals that are not top level atomic expressions. Consider the situation shown in Figure 3 where we have a top level goal g that has been reduced to the conjunction $e = e_1 \wedge \ldots \wedge e_n$. Suppose that the planner is working on the conjunct $c \in e$, and there are two actions a_1 and a_2 that could be used for achieving c. For convenience, let p refer to the set of clauses in e that precede c in the temporal order, and let f represent the set of clauses in e that follow c in the temporal order. Formally,

$$p \equiv \tau(c),$$
$$f \equiv e - c - \tau(c).$$

The total cost of achieving the conjunction consists of three components. First, we have the cost of achieving those conjuncts, p, that occur before c in the temporal order. The cost of achieving these conjuncts is unaffected by the choice between a_1 and a_2, and therefore consists of a subset of the terms from equation (3):

$$C(p, s) = \sum_{d \in p} C(d \backslash \varphi(d), s|\tau(d)).$$

The second component is simply the cost of achieving c in the situation $s|p$ using either a_1 or a_2. This cost is given by equation (1)

$$C(Pre(a_i)a_i, s|p) = C(Pre(a_i), s|p) + C(a_i, s|p|Pre(a_i))$$

The third component is the cost of achieving those conjuncts, q, that follow c in the temporal ordering τ. After achieving c the situation will be $s' \equiv s|p|Pre(a_i)|a_i$. Each clause, d, that follows c will therefore be achieved in the situation $s'|\tau(d) - p - c$. The cost for achieving these conjuncts is therefore

$$C(q, s') = \sum_{d \in q} C(d \backslash \varphi(d), s'|\tau(d) - p - c).$$

Putting these three components together we get

$$\sum_{d \in p} C(d \backslash \varphi(d), s|\tau(d)) + C(Pre(a), s|p) + C(a, s|p|Pre(a)) + \sum_{d \in q} C(d \backslash \varphi(d), s'|\tau(d) - p - c).$$

In comparing two actions a_1 and a_2 for achieving c, the first sum in this expression will be the same for both actions and will therefore cancel out. However, the final sum cannot be dropped. This is because the terms depend on the situation s' which is influenced by the actions used to achieve c. As a result, in general we cannot choose the action for achieving a conjunct c independent of the other clauses in the conjunction.

2.4 Computing Temporal Order for Conjunctions

In the previous two sections we assumed that we knew the appropriate temporal ordering, τ, for each conjunction. Normally, the temporal ordering for the conjuncts in a conjunction is decided during the actual planning process. The planner works on one conjunct, and in doing so, finds temporal constraints among the conjuncts. In some cases conjunctions may not be *serializable* [12]; that is, it may be necessary to partially achieve one conjunct, then partially achieve another, before completely achieving either one. The classic example of a non-serializable conjunctive goal is Sussman's anomaly [25]. In Sussman's anomaly the goal is the simple blocks world statement $On(A, B) \wedge On(B, C)$ and the database contains $On(A, Table)$, $On(B, Table)$, and $On(C, A)$. If we attempt to satisfy either conjunct in its entirety we will end up undoing that conjunct in order to achieve the other conjunct. In such cases it doesn't really make sense to talk about a temporal ordering for a conjunction.

However, our purpose in finding a temporal ordering for a conjunction is 1) to estimate the cost of achieving the conjunction, and 2) to allow us to choose a reasonable order in which to plan for the clauses in the conjunction. Our assumption of a temporal order in no way restricts the actual temporal order to be determined during the planning process. Thus, even for non-serializable conjuncts it makes sense to use an approximate temporal ordering.

Let e refer to the set of conjuncts of interest. First suppose that none of the elements in e share any variables. If τ is the temporal ordering in which the conjuncts will be achieved, the cost of achieving the conjunction is, as given previously in equation (2):

$$C(c,s) = \sum_{c \in e} C(c, s|\tau(c))$$

A best possible temporal ordering would then be one that minimizes this expression, i.e.:

$$\tau : \min \sum_{c \in e} C(c, s|\tau(c)).$$

If there are variables shared between clauses in e the choice becomes more complicated. In this case we are interested in finding the best possible ordering given that those variables are already chosen to give the optimal plan. Formally, let v refer to the set of all variables that are shared by two or more conjuncts in e. Then we are interested in choosing the ordering:

$$\tau : \min_{\substack{v \in D \\ \tau}} \sum_{c \in e} C(c \backslash v, s|\tau(c)),$$

where D is the domain over which the variables v can range.

This has rather abysmal computational properties. In the first place, the domain D may be large, making the minimization over v impractical. One possible approach is to choose v arbitrarily from the domain by binding those variables to skolem constants and hope that the resulting temporal ordering provides an accurate cost assessment for the actual plan that would be found for this conjunction.

Even if we do this, there is a second problem: given n conjuncts we still face the prospect of having to consider $n!$ possible temporal orderings. Doing this is reasonable if $n \leq 4$. For large n we need a more efficient means of searching the space of possible orderings.

One possibility is to do pairwise comparisons between the different conjuncts. For two conjuncts c and d, if

$$C(c,s) + C(d,s|c) \ll C(d,s) + C(c,s|d)$$

then it is reasonable to suppose that c should precede d. Thus we could determine for conjunct 1, which, if any of the remaining conjuncts are strongly affected if conjunct 1 precedes or follows them. Then for each such conjunct, c, we could determine which other conjuncts are strongly affected by their ordering with respect to c, and so on. This approach has n^2 behavior, which is reasonable considering that the number of conjuncts in a typical conjunction is generally less than ten. However, note that this approach is not guaranteed to give the best ordering, because in doing these pairwise comparisons we are ignoring the effect of other conjuncts that we may later discover need to precede one or both of the conjuncts. Another way of saying this is that the effect of one conjunct on the cost of achieving a later one is not always independent of intervening conjuncts, or of other conjuncts preceding both conjuncts. However, this heuristic approach does seem to match our intuitions about how rough temporal ordering can be done quickly.

2.5 Computing Planning Order for Conjunctions

Given a temporal ordering τ for a set of conjuncts e, we now consider the problem of finding the best planning order φ for the conjuncts. Our purpose in doing this is 1) so that the planner can use equation (3) to evaluate the cost of achieving a conjunction, and 2) so that the planner can work on the conjuncts in a sensible order.

It is important to note that the planning order for a set of conjuncts may be very different from the temporal ordering. To see this, consider a simple (partial) travel plan consisting of three legs: travel from Stanford to San Francisco airport (SFO), travel from San Francisco airport to Boston's Logan airport, and travel from Logan airport to MIT. The temporal ordering for these three steps is obvious – first we get to the airport, then fly to Boston, then get to MIT. Suppose that we tried to plan for these steps in the same order. In this case we may arrange to end up at the airport at a time when there are no convenient flights, or when the only flight available is costly. Worse yet, it might be that Logan is snowed in, or that the air traffic controllers are on strike. This could lead to a very long wait at the airport.

For this conjunction, the planner should plan the flight leg before planning for ground transportation at either end. Intuitively, the reason is that the number of flights is limited, and the difference in cost between these flights may be substantial. In contrast, the other two legs can be accomplished at almost any hour, and the cost of achieving them does not depend significantly on the time of day.

It is worth noting that the planning order may not be fixed for a given conjunction. In our trip example, the best planning order might very well depend upon the actual circumstances. If the trip were from Stanford to UCLA, we might wish to plan one of the ground transportation legs first, in order to take advantage of available rides, or desirable traffic conditions. In flying to LA this is a reasonable strategy because there are so many flights available that the time of departure probably has little effect on the cost or likelihood of achieving that leg of the trip.

This example illustrates why the technique of hierarchical planning has often proven problematic; there may not be a single, fixed ordering for a given conjunction. The best ordering depends, in general, on the state of the world and on the other possible actions available to the agent.

In equation (3) we gave the following expression for computing the cost of achieving a conjunction e using a temporal ordering τ and a planning order φ:

$$C(e, s) = \sum_{c \in e} C(c \backslash \varphi(c), s | \tau(c))$$

Given a temporal ordering, τ, we could therefore find the best planning order φ by simply trying every possible permutation of the clauses from e in the above equation, and choosing the one that gives the cheapest total cost. Thus:

$$\varphi : \min \sum_{c \in e} C(c \backslash \varphi(c), s | \tau(c))$$

Using this equation to determine planning order suffers from the familiar problem of having to consider all $n!$ possible orderings of the conjuncts. Fortunately we can do much better than this by considering how the terms in the summation are affected by the planning order. We first note that the lowest possible expected cost for a clause c is given by $C(c, s|\tau(c))$, i.e. the expected cost with none of its variables bound. If variables are bound in the planning of other clauses, this cost could go up. The highest possible cost is when all variables shared with other clauses are already bound. In this case the expected cost of achieving the conjunct would be given by $C(c \backslash v, s|\tau(c))$, where v is the set of shared variables in c. The basic insight is that if these two costs are not much different (i.e. binding the variables in a conjunct doesn't have much effect on the cost of achieving the conjunct) we can postpone working on the conjunct until later. Formally:

Theorem 1 *Let v be the set of all variables shared by two or more conjuncts in a conjunction e. Planning for a conjunct $c \in e$ can be postponed until after planning for all other conjuncts in e if*

$$\Delta C(c, v) \equiv C(c \backslash v, s|\tau(c)) - C(c, s|\tau(c))$$

is small compared to $C^*(e, s) \equiv \sum_{c \in e} C(c, s|\tau(c))$.

Using this theorem we could examine the conjuncts initially, and postpone those with small ΔC. We could then repeat the whole process, using the reduced set of conjuncts and the corresponding reduced set of variables. For many problems this is enough to reduce the number of conjuncts sufficiently that the $n!$ problems go away. In particular, this approach disposes of the trip planning problem quickly, because ΔC is small for both ground transportation legs of the trip in comparison to C^*, which includes the cost of the flight. As a result, planning for these two legs would be postponed until after planning for the flight.

The above theorem also suggests a powerful heuristic algorithm for determining planning order. The basic idea is to repeatedly strip off the conjunct that suffers the least by being postponed.

Algorithm 1

1. Let v be the set of all variables common to two or more conjuncts in e.
2. For each conjunct $c \in e$ compute $\Delta C(c, v)$.
3. Remove the conjunct with the smallest ΔC value from the set e. Planning for this conjunct should be postponed until after planning for all other conjuncts in e.
4. Remove all other conjuncts, d, from e for which $\Delta C(d, v)$ is negligible. Planning for these conjuncts should also be postponed until after planning for all remaining conjuncts in e.
5. Repeat until e is empty.

We can further improve on this algorithm by noticing that ΔC does not need to be recomputed for every conjunct remaining in e each time we remove a conjunct from e. In fact, the only ones we need to recompute are those containing a variable that is shared only with the conjunct that was removed. Thus, if we were to keep a list of the conjuncts containing each variable, when a conjunct is removed, we could immediately determine those conjuncts remaining that share variables uniquely with the conjunct that was removed.

In the worst case, this algorithm has n^2 behavior. This occurs when every conjunct that is removed causes recomputation of ΔC for every remaining conjunct in the set e. For example, the conjunction

$$\begin{array}{l} r_1(x_{12}, x_{13}, \ldots, x_{1n}) \\ r_2(x_{12}, x_{23}, \ldots, x_{2n}) \\ \qquad\qquad \vdots \\ r_i(x_{1i}, x_{2i}, \ldots, x_{i-1,i}, x_{i,i+1}, \ldots, x_{i,n}) \\ \qquad\qquad \vdots \\ r_n(x_{1n}, x_{2n}, \ldots, x_{n-1,n}) \end{array}$$

takes this algorithm $\Theta(n^2)$ time. However, this is a particularly bizarre case because every conjunct uniquely shares one variable with every other conjunct. Suppose that a conjunction has k variables that are shared by two or more conjuncts. Note that each one of these variables can cause the recomputation of ΔC for only a single conjunct (the last one remaining in the ordering process that contains that variable). Thus, the above algorithm actually has a worst case behavior of $n + k$ (for $k < n^2$). For the typical conjunction, $k \leq n$, which means that the behavior of the algorithm is usually linear in n.

Another interesting property of this algorithm is that recomputed values of ΔC always get smaller. To be more precise:

$$v' \subset v \Rightarrow \Delta C(c, v') \leq \Delta C(c, v)$$

This is why we can remove all other conjuncts having negligible $\Delta C(c, v)$ at each stage in the algorithm; their values can only get smaller as we remove more conjuncts.

While the above algorithm is heuristic in nature, we can actually prove some rather good bounds on its performance. Let $c_1, \ldots, c_n$ be the ordering determined by the above algorithm. Let v_i be the set of shared variables remaining in e just before the clause c_i is postponed. The cost of the plan produced using this planning order will be

$$\sum_{c_i \in e} C(c_i \backslash v_i, s | \tau(c_i))$$

The best possible plan can be no better than

$$C^*(e, s) \equiv \sum_{c_i \in e} C(c_i, s | \tau(c_i))$$

If we subtract the two, we get

$$\sum_{c_i \in e} C(c_i \backslash v_i, s | \tau(c_i)) - C(c_i, s | \tau(c_i))$$

$$= \sum_{c_i \in e} \Delta C(c_i, v_i)$$

which is just the sum of the final ΔC values computed by the algorithm for each conjunct. It is a simple matter to modify the algorithm to keep track of this running sum. As long as the sum remains small in comparison to the overall expected cost of the plan C^*, the ordering provided by the algorithm is guaranteed to be a good one. However, if the sum becomes large, it may be worthwhile to consider other possible orderings. This could be done by modifying the algorithm to consider postponement of other conjuncts than the one with minimal ΔC if that ΔC is large. This changes the algorithm into a best-first search procedure based on the accumulated sum of the ΔC values of postponed conjuncts.

3 The Uncertain World

In the analysis of the previous section we assumed that the planner had perfect knowledge of the world, and perfect knowledge of the affects of its actions. When uncertainty is present, either in the planner's model of the world, or in the planner's description of the available actions, the analysis becomes considerably more complicated. We can extend much of our analysis to deal with the uncertain world.

Consider again the simple case shown in Figure 4, where we have a top level goal g, and there are two different actions, a_1 and a_2, that have some chance of achieving g. Suppose that the preconditions for both of these actions are already true of the situation s. Further suppose that a_1 has only a 50% chance of achieving g while a_2 is certain to achieve g. If a_1

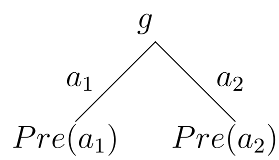

Figure 4: Simple choice between actions

and a_2 have the same cost then we should clearly prefer a_2. But if a_2 is much more expensive than a_1 it is no longer obvious what we should do. If after attempting a_1, the action a_2 would still be possible, we might want to try a_1 first, and if that fails, then try a_2. However, if a_1 changes the world so that a_2 would no longer be possible then we may wish to prefer a_2.

From the above example we can see that when the world is uncertain, the decision between two actions depends on several factors:

1. the cost of each action,
2. the chance that each action will achieve the desired goal,
3. the extent to which each action would damage the world in the event of failure, and
4. the importance of achieving the overall goal g.

The notion of the cost of performing an action is already familiar, but the other three factors are new. In order to formalize the criteria for choosing between actions we need additional notation allowing us to refer to probability and importance.

3.1 Preliminaries

We will use the expression $P(q, s)$ to refer to the probability that a proposition q holds in the situation s. For convenience we let

$$\begin{aligned} \overline{P}(q, s) &\equiv 1 - P(q, s) \\ &= P(\neg q, s). \end{aligned}$$

We let $I(g)$ refer to the importance of a goal g. This quantity is to be regarded as the price of failure to achieve the goal.

Recall, from the previous section, we defined $C(q, s)$ for a partial plan to be:

$$C(q, s) \equiv C(B(q, s), s).$$

Likewise, we defined $s|q$ for a partial plan to be:

$$s|q \equiv s|B(q, s).$$

Unfortunately, our simple definition of the best possible completion for a partial plan q only took cost into account. In an uncertain world, this is no longer sufficient. We must therefore redefine B to take probability of success and price of failure into account:

$$B(q, s) \equiv p : \min[C(p, s) + P(\neg g, s|p)I(g)]$$

As before, this is a minimization over all possible plans. The first term in the minimization is the expected cost of performing the plan p in situation s. The second term is the expected damage that will result from failure of the plan. The sum therefore represents the total expected cost of using the plan p, exclusively, to try to accomplish g. Note that, unlike our previous definition for B, we do not require that the best plan be guaranteed to succeed. For our new definition, if $I(g)$ is relatively small, the best plan could be the empty plan, having no chance of success, but no cost. The total damage would then be just the importance of the goal. Intuitively, this means that the goal is more trouble than it is worth. If $I(g)$ is larger, the best plan under this definition will typically consist of many smaller fallible plans strung together with conditionals, so that if the first fails, the second is tried, and so on. In this case the probability of failure usually becomes small, and the cost term in the minimization dominates.

Using this new definition of B, our previous definitions of C and $|$ remain unchanged. We define one other quantity using B; $A(q, s)$ will be used to refer to the *achievability* of q in s, that is, the probability that the best plan for achieving the proposition q from the state s will succeed. Formally,

$$A(q, s) \equiv P(q, s|q).$$

As mentioned in the Section 1.2, we are assuming that information is available that will allow us to compute $A(q, s)$ for atomic propositions and their negations. For convenience, we define $\overline{A}(q, s)$ as follows

$$\begin{aligned}\overline{A}(q, s) &\equiv 1 - A(q, s) \\ &= P(\neg q, s|q).\end{aligned}$$

3.2 Choosing Between Alternatives

Using the notation introduced above, we can write down formal expressions for the expected cost of using an action a to achieve a goal g from a situation s. First, there is the term $C(a, s)$ for the cost of performing a. There is probability $\overline{P}(Cons(a), s|a)$ that a will not achieve its intended objective, $Cons(a)$. In this case we need to consider the difficulty of achieving g from the resulting state. This is given by the expression $C(g, s|a)$. There is some chance that it will not be possible to achieve g from the state $s|a$. This is given by the probability $\overline{A}(g, s) = 1 - A(g, s|a)$. In this case, complete failure will occur, and the additional penalty $I(g)$ is incurred. Putting this all together, we have:[1]

$$C(a, s) + \overline{P}(Cons(a), s|a)[C(g, s|a) + \overline{A}(g, s|a)I(g)].$$

In the material to follow, we will often need to include many expressions like the one above. For convenience, we define

$$E(p, g, s) \equiv C(p, s) + \overline{A}(p, s)[C(g, s|p) + \overline{A}(g, s|p)I(g)]. \tag{4}$$

Intuitively, we can think of $E(p, g, s)$ as referring to the net cost or damage that will be incurred if the partial plan p is performed in attempting to achieve the goal g.

To see how we can use this equation, we return to the example illustrated in Figure 4, where we have a top level goal g and two possible actions a_1 and a_2 that have some chance of achieving g.

Example 1 First suppose that the preconditions for both actions hold in the initial state. Further suppose that a_1 has a 50% chance of achieving g while a_2 is certain to achieve g. In this case we have

$$\begin{aligned} E(a_1, g, s) &= C(a_1, s) + .5[C(g, s|a_1) + \overline{A}(g, s|a_1)I(g)] \\ E(a_2, g, s) &= C(a_2, s). \end{aligned}$$

Clearly, if $C(a_2, s) \leq C(a_1, s)$ the second action, a_2, will be better.

Example 2 Suppose instead, that $C(a_2, s) \gg C(a_1, s)$. If the importance of the goal is low enough that $.5I(g) \leq C(a_2, s)$ then we should prefer the first action, a_1, alone.

Example 3 Again suppose that $C(a_2, s) \gg C(a_1, s)$ but that the importance of g is very high. If a_1 messes up the world so that g cannot be accomplished easily from $s|a_1$, then either the term $C(g, s|a_1)$ will be large, or the term $\overline{A}(g, s|a_1)I(g)$ will be large. If these are larger than $C(a_2, s)$ then a_2 will be preferred.

Example 4 Again suppose that $C(a_2, s) \gg C(a_1, s)$ and the importance of g is very high. However, suppose that a_2 is still possible in $s|a_1$. This means that $C(g, s|a_1) \leq C(a_2, s)$. As a result a_1 should be preferred, because if it fails, a_2 can still be used.

[1]In fact, this expression is not quite right: The term $C(g, s|a)$ is already an average over both situations where a succeeds and situations where a fails. What we really want here is the situation resulting from performing a in s where the intended consequences of a do not hold. As yet, we do not have any sensible notation for representing this subtlety.

In all of these examples we assumed that the preconditions for achieving the actions a_1 and a_2 already held in s. If this is not the case, more terms must be added to the cost equation. For an action a, with preconditions $Pre(a)$, we first need the term $C(Pre(a), s)$ for the expected cost of achieving the preconditions. There is chance $\overline{A}(Pre(a), s)$ that the preconditions will not be achieved by the best plan for those preconditions. In this case we will have the additional cost, $C(g, s|Pre(a))$ of achieving the goal g from this resulting state. If g cannot be achieved (by the best plan) from this state the additional cost $I(g)$ will be incurred. Putting these terms together we get

$$C(Pre(a), s) + \overline{A}(Pre(a), s)[C(g, s|Pre(a)) + \overline{A}(g, s|Pre(a))I(g)].$$

Note that by our definition of E, this is just $E(Pre(a), g, s)$. In addition to these terms we have the terms for the case where the preconditions are accomplished using the best plan. The probability of this is just $A(Pre(a), s)$. In this case we will incur the cost $E(a, g, s|Pre(a))$. Putting these terms together, the entire expected cost for achieving the preconditions for a, followed by a is given by

$$E(Pre(a), g, s) + A(Pre(a), s)E(a, g, s|Pre(a)). \tag{5}$$

As with the previous examples, we could use this equation to decide between any two possible actions for achieving a top-level goal g.

3.3 Conjunctions

Given the above definitions it is fairly straightforward to extend the expected cost analysis to conjunctions. As we did in Section 2.2, we assume a temporal ordering denoted by the function τ and a planning order denoted by the function φ. The total expected cost associated with each conjunct c will be $E(c\backslash\varphi(c), g, s|\tau(c))$. By our definition of E, this includes the cost of performing the best plan for $c\backslash\varphi(c)$, as well as the expected cost of achieving g in the resulting world if the best plan for $c\backslash\varphi(c)$ fails to achieve its objective. Likewise, it includes the cost associated with failure to achieve g if g is not possible in the resulting world.

Note, however, that the best plan for $c\backslash\varphi(c)$ will only be attempted if the best plans for all of the other conjuncts earlier in the temporal ordering are successful. Thus, the E term for c only contributes to the cost a certain percentage of the time, given by $A(\tau(c), s)$. The expected cost contribution for each conjunct c will therefore be $A(\tau(c), s)E(c\backslash\varphi(c), g, s|\tau(c))$. Summing over all conjuncts, we get

$$E(e, g, s) = \sum_{c \in e} A(\tau(c), s)E(c\backslash\varphi(c), g, s|\tau(c)) \tag{6}$$

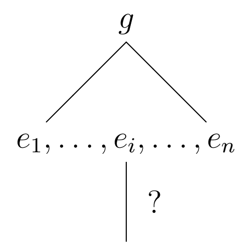

Figure 5: Coping with uncertainty

Note that equation (5) is just a special case of this equation, where the temporal order is $Pre(a)$ followed by a.

It appears that the analysis given in Sections 2.4 and 2.5 for determining temporal order and planning order for a conjunction can also be extended to the uncertain world. The primary difference seems to be that instead of using cost, C, and change in cost, ΔC, for making ordering decisions, we must use a utility measure, the ratio of probability of achieving a proposition, to expected cost of achieving that proposition. Formally:

$$U(q, s) \equiv \frac{A(q, s)}{E(q, g, s)}$$

and

$$\Delta U(q, v) \equiv U(q, s) - U(q \backslash v, s)$$

The details of this analysis have not yet been completed.

3.4 Options under Uncertainty

There are a number of interesting control issues that arise in an uncertain world that do not arise in a certain world. Suppose that the planner is currently investigating a subgoal $e = e_1 \wedge \ldots \wedge e_n$ for achieving its overall goal g. Let c be the conjunct in e that the planner has chosen to work on next, and suppose that it is uncertain whether or not c will hold at the appropriate time. The question is, what should the planner do about planning for c. There are a number of different possibilities:

1. **Assumption:** it could assume that the condition c holds, and proceed to work on the remaining conjuncts,

2. **Coercion:** it could coerce the world into a known state, by planning to achieve c whether it holds or not,

3. **Conditional Planning:** it could insert a conditional into the plan and plan separately for the two different possibilities.

4. **Predetermination:** it could attempt to determine the needed information while planning,

5. **Deferral:** it could defer planning for c until later when the state of the world is known more completely,

To illustrate these different possibilities consider a mobile robot operating in a machine shop. Suppose that the robot's goal is to fabricate a widget, and to do so it needs to bore a hole in a piece of stock using the drill press. The complication is that there might be some debris on the drill press table.

The first option, assumption, is that the robot could simply assume that the table is clear, and continue planning for the remaining conjuncts in c. Intuitively, this would be a reasonable strategy if 1) the probability is high that the table will be clean, and 2) failure to take account of a dirty table would not be a disaster. Assumption is not so reasonable if there is a high chance of failure, or if the widget is difficult to repair or replace.

The second option, coercion, is that of planning to achieve the precondition that the table be clear, independent of whether or not it already is. For example, the robot could guarantee that the table is clear by always vacuuming the table, whether there is debris there or not. This would be a good alternative if the probability that the table is clear is very low. It would also be a good alternative if it is easier to clean the table than to determine whether or not it is clean. This might be the case if the robot has poor visual abilities.

The third option is for the planner to insert a conditional into the plan, so that the robot will follow different predetermined courses of action depending on whether there is debris on the table or not. In this case, the planner must also realize a sensory goal of determining the condition of the table at the appropriate time. This option is a reasonable one if 1) there is a fair chance that the desired condition will not hold, 2) the cost of achieving the condition is significant, and 3) it is possible to determine the condition accurately using available sensory actions.

The fourth option, predetermination, is for the robot to look across the room to see if the drill press table is clear, and then proceed with the remainder of the planning. This alternative can be seen as an extreme case of conditional planning, where the sensory actions and the conditional are placed at the very beginning of the plan, and all further planning is delayed until after the sensory actions and conditional are performed. In our example, predetermination would be a viable option if the robot could just look across the room to determine if the drill press table were clean, and if it knew there was very little chance that the table would get messed up in the interim.

The fifth option, delay, is for the robot to plan the actions for getting to the drill press, inserting the appropriate bit, setting the speed, and aligning, clamping, and drilling the widget, but postpone planning for the subgoal of having a clean table. In this case, the robot would start executing the partial plan by moving to the drill press, changing the bit, and setting the speed. Then it would run into the unplanned subgoal that the table must be clear. This would require that the robot stop to check this subgoal, and plan to accomplish the subgoal if it is not already true. Deferral would be a good option if 1) the planner were fairly confident of its ability to generate a plan for cleaning the table, and 2) if there were no critical time constraints imposed on the robot during execution.

To choose between the different alternatives for dealing with uncertainty we need to be able to evaluate the expected cost of the plans that result from each of these alternatives. In equation (6) we already have the basic tools necessary to do this; we just need to substitute the appropriate action for dealing with the uncertain clause, c, into the equation. We start with the simplest case, coercion, and then consider assumption, and finally, conditional plans and predetermination.

3.4.1 Coercion

Given the conjunction e, let c refer to the uncertain clause in e that the planner is concerned with. As before, let p refer to the set of conjuncts in e that precede c in the temporal order, and let q represent the set of conjuncts in e that follow c in the temporal order.

Expanding equation (6) we get:

$$E(e,g,s) = E(p,g,s) + A(p,s)\Big[E(c,g,s|p) + A(c,s|p)E(q,g,s|p|c)\Big] \tag{7}$$

Since we are assuming that c will be coerced, no further reduction of this equation is necessary, provided that the best plan for achieving c in $s|p$ will be used.

3.4.2 Assumption

For the case of assumption, the analysis is a bit trickier. First of all, we assume that, if c doesn't hold, other actions will get performed before the error is discovered. If the planner has already completed planning for the conjuncts in q it may be possible to predict the point at which an error in the assumption will be discovered.

Let q_1 represent the subset of q that will be attempted before the problem with c is discovered, and let q_2 be the remainder of the conjuncts in q. As before, we first have the expected cost term, $E(p,g,s)$ of achieving p. Since c is being assumed, there is no cost associated with that part of the plan, and it will always be successful. As a result, the cost of achieving q_1 will always be incurred. Assuming that q_1 is achieved we then have two terms, depending on whether or not the assumption about c was actually correct. If it is, the cost of achieving q_2 is incurred. Otherwise, the cost of achieving g in this new "mangled" world will be incurred. Putting all of these terms together, we get:

$$E(p,g,s) + A(p,s)\Big[E(q_1,g,s|p) + A(q_1,s|p)\Big[P(c,s|p)E(q_2,g,s|p|q_1) + P(\neg c,s|p)E(\mathcal{F},g,s|p|q_1)\Big]\Big] \tag{8}$$

where $\mathcal{F}$ refers to the empty plan that always fails. Thus

$$E(\mathcal{F},g,s) = C(g,s) + \overline{A}(g,s)I(g).$$

We can simplify the above equation slightly by noting that

$$E(q_1,g,s|p) + A(q_1,s|p)P(c,s|p)E(q_2,g,s|p|q_1) = E(q,g,s|p|q_1) - A(q_1,s|p)P(\neg c,s|p)E(q_2,g,s|p|q_1)$$

If we substitute this into equation (8) we get

$$E(p,g,s) + A(p,s)\Big[E(q,g,s|p) + A(q_1,s|p)P(\neg c,s|p)\Big[E(\mathcal{F},g,s|p|q_1) - E(q_2,g,s|p|q_1)\Big]\Big] \tag{9}$$

Since we are assuming that pcq is a good plan for achieving g, we know that it will be at least as difficult to achieve g in $s|p|q_1$, as it will be to achieve q_2. This means that the $E(\mathcal{F},g,s|p|q_1) - E(q_2,g,s|p|q_1)$ term is always greater than or equal to zero.

Suppose we compare equation (9) with equation (7), the expected cost for coercion. The initial E term, and the $A(p,s)$ multiplier are the same for both equations, so we can cancel these terms. Comparing the remaining terms we see that assumption is preferable to coercion if and only if

$$E(q,g,s|p) + A(q_1,s|p)P(\neg c,s|p)\Big[E(\mathcal{F},g,s|p|q_1) - E(q_2,g,s|p|q_1)\Big] \le E(c,g,s|p) + A(c,s|p)E(q,g,s|p|c)$$

As it stands, this equation is not particularly illuminating. Suppose we make the simplifying assumption that the achievement of c in $s|p$ does not significantly affect the cost of achieving q. We can then collect the E terms for q on the left hand side giving:

Theorem 2 *Assumption is preferable to coercion if and only if*

$$\overline{A}(c,s|p)E(q,g,s|p) + A(q_1,s|p)P(\neg c,s|p)\Big[E(\mathcal{F},g,s|p|q_1) - E(q_2,g,s|p|q_1)\Big] \le E(c,g|s|p)$$

From this, we see that as the cost of achieving the condition c goes up, assumption becomes a better alternative. Likewise, as the probability that c will hold in $s|p$ increases, assumption becomes a better alternative. Conversely, as the difficulty of achieving g in the mangled world $s|p|q_1$ increases, coercion becomes the better option. All of these match our intuitions about what should happen. A somewhat less intuitive observation is that if there is significant chance that c may not be achievable in $s|p$ then coercion may be the better alternative. This is because it is less costly to fail early than late.

One final note is in order; the planner may not always know q_1 and q_2. In this case it seems better to be conservative and assume $q_1 = q$ and $q_2 = \emptyset$; that is, assume that if the condition c fails to hold, the entire rest of the plan will be performed before the error is detected. In this case, the $E(q_2, g, s|p|q_1)$ term drops out in the above comparison.

3.4.3 Conditional

Until now, we have not talked about conditional statements or sensory actions in plans. In order to talk about these, we need to introduce some new notation. We will use the expression $?c$ to indicate the goal of determining whether or not c is true in the world. The expression $\langle ?c, a, b\rangle$ will refer to the conditional action of first determining c, then achieving a if c is true, and b otherwise.

The cost of performing a conditional statement $\langle ?c, a, b\rangle$ can be expressed in terms of the costs of the individual steps $?c$, a, and b. First, there is the cost of determining whether or not the condition c is true. This is given by $E(?c, g, s)$. Suppose that the status of c is successfully determined (many effectory and sensory actions may be required). Then if c is true, the cost $E(a, g, s|?c)$ will be incurred, otherwise, the cost $E(b, g, s|?c)$ will be incurred. Thus:

$$E(\langle ?c, a, b\rangle, g, s) = E(?c, g, s) + A(?c, s)\Big[P(c, s)E(a, g, s|?c) + P(\neg c, s)E(b, g, s|?c)\Big]$$

Given our conjunction $e = pcq$, we can now use this to express the cost of the conditional plan that involves checking c, achieving q if c holds, and using some alternate plan to achieve g otherwise. (For simplicity, we are only considering the case where the sensory action occurs immediately before the condition c must be accomplished.) We get:

$$\begin{aligned}
&E(p\langle ?c, q, \mathcal{F}\rangle, g, s) \\
&\quad = E(p, g, s) + A(p, s)E(\langle ?c, q, \mathcal{F}\rangle, g, s|p) \\
&\quad = E(p, g, s) + A(p, s)\Big[E(?c, g, s|p) + A(?c, s|p)\Big[P(c, s|p)E(q, g, s|p|?c) \\
&\qquad\qquad\qquad\qquad + P(\neg c, s|p)E(\mathcal{F}, g, s|p|?c)\Big]\Big]
\end{aligned} \tag{10}$$

While it is true in general that the planner may want to consider an entirely different plan for achieving g if c does not hold in the world, often the appropriate course is simply to achieve c if it doesn't already hold. We will refer to this as *conditional coercion*. For this alternative, the expected cost equation will be:

$$\begin{aligned}
&E(p\langle ?c, q, cq\rangle, g, s) \\
&\quad = E(p, g, s) + A(p, s)E(\langle ?c, q, cq\rangle, g, s|p) \\
&\quad = E(p, g, s) + A(p, s)\Big[E(?c, g, s|p) + A(?c, s|p)\Big[P(c, s|p)E(q, g, s|p|?c) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad + P(\neg c, s|p)E(cq, g, s|p|?c)\Big]\Big] \\
&\quad = E(p, g, s) + A(p, s)\Big[E(?c, g, s|p) + A(?c, s|p)\Big[P(\neg c, s|p)E(c, g, s|p|?c) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad + A(c, s|p|c?)E(q, g, s|p|?c|c)\Big]\Big]
\end{aligned} \tag{11}$$

If we assume that $?c$ can always be achieved, and that achieving $?c$ doesn't mess up the world, this equation can be simplified to be:

$$E(p, g, s) + A(p, s)\Big[C(?c, s|p) + P(\neg c, s|p)E(c, g, s|p) + A(c, s|p)E(q, g, s|p|c)\Big] \tag{12}$$

Suppose we compare this with equation (7), the equation for coercion. As before, the initial E and A terms drop out. The final terms in both equations also drop out. As a result, we get

Theorem 3 *Conditional coercion should be preferred to coercion if and only if*

$$C(?c, s|p) \leq P(c, s|p)E(c, g, s|p)$$

Proof: From the equations we know that conditional coercion will be preferred to coercion if and only if

$$C(?c, s|p) + P(\neg c, s|p)E(c, g, s|p) \leq E(c, g, s|p).$$

Collecting terms gives the desired result. □

3.4.4 Predetermination

As we mentioned earlier, predetermination is a special form of conditional planning, where the sensory action and conditional are pushed all the way to the beginning of the plan. Predetermination usually also implies that planning is deferred for all subsequent actions in the plan, but this is not really the important issue in comparing predetermination with conditional planning, since deferral could equally well be applied to one or both branches of an ordinary conditional plan. The important difference between predetermination and conditional planning is the difference in sensing cost between the beginning of the plan, and the time at which the condition must be accomplished.

If we ignore the issue of deferral, the expected cost for a plan produced by predetermination will be:

$$
\begin{aligned}
&E(\langle ?c, pq, pcq\rangle, g, s) \\
&\quad = E(?c, g, s) + A(?c, s)\Big[P(c, s|p)E(pq, g, s|?c) + \quad P(\neg c, s|p)E(pcq, g, s|?c)\Big] \\
&\quad = E(?c, g, s) + A(?c, s)\Big[E(p, g, s|?c) + A(p, s|?c)\Big[P(c, s|p)E(q, g, s|?c|p) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad + P(\neg c, s|p)E(cq, g, s|?c|p)\Big]\Big]
\end{aligned}
\tag{13}
$$

Suppose we assume that the performance of $?c$, and the achievement of c do not significantly change the cost or likelihood of achieving either p or q. Then we can simplify the above expression to be:

$$
\begin{aligned}
E(?c, g, s) + A(?c, s)\Big[E(p, g, s) + A(p, s)\Big[&P(c, s|p)E(q, g, s|p) \\
&+ P(\neg c, s|p)E(cq, g, s|p)\Big]\Big]
\end{aligned}
\tag{14}
$$

If we further assume that $?c$ can always be achieved (although perhaps at considerable cost) the above expression can be further simplified to

$$
C(?c, s) + E(p, g, s) + A(p, s)\Big[P(c, s|p)E(q, g, s|p) + P(\neg c, s|p)E(cq, g, s|p)\Big] \tag{15}
$$

In comparing this with equation (12) we note that the only difference is in the cost term for $?c$. Whenever

$$
C(?c, s) \leq A(p, s)C(?c, s|p)
$$

predetermination should be preferred over a conditional plan; that is, whenever the cost of determining c is lower initially than at the time when c needs to hold, predetermination should be preferred. Note that this only holds when $A(?c, s) \approx 1$. When $A(?c, s) \ll 1$ the cost term, $E(?c, g, s)$, will become large, and a conditional plan will be preferred.

Note that these costs do not take planning time into consideration. To make the comparison fair in this regard, we are assuming that planning will be delayed for both branches of the conditional plan.

3.5 Deferral

Deferral is quite different from the other options for dealing with uncertainty. Even though planning may be deferred for a condition c, it will still be necessary to decide what to do about the condition later on. The decision is therefore delayed, but not eliminated. In our example, after the robot gets to the drill press, and performs various setup operations, it will still need to decide what to do about the precondition of having a clean table. It could decide to assume the condition, to force the condition to be true (coercion), or to sense the condition and insert a conditional. As a result, it seems more sensible to regard deferral not as a means of coping with uncertainty, but rather as a means of avoiding planning for conditions that are relatively unlikely to occur. In fact, we might want to apply deferral to clauses other than those suffering from uncertainty. For example, after inserting a conditional into a plan, it might be appropriate to defer planning on either one or both branches of the conditional, even though the status of the conditions in those branches may be known with certainty.

As we just implied, the advantage to deferring planning for a condition is to avoid doing planning for a situation that is unlikely to actually occur. If it is unlikely that the drill press table is dirty, then constructing a plan for cleaning the table would usually be wasted effort. If planning cost were the only consideration, we would want our planner to defer work on every subgoal until the moment that it needs to be achieved. However, there are two other factors that need to be considered before deferring planning for any given clause:

1. temporal constraints on the planning process,

2. potential inaccuracy of the cost and probability models.

Time constraints on the planning process can either favor or oppose deferral. If there are time critical sequences of operations in a plan, there may not be enough time in between steps to allow for planning of deferred conditions. For example, our shop robot might need to perform a series of fabrication operations quickly while a material remains at a given temperature and consistency. Conversely, if there is little time before a plan must be initiated, the planner may not have enough time to fill in all the details before beginning execution of the plan. As an example of this, our shop robot might need to proceed to the drill press immediately, because access to the drill press is about to be blocked by other shop operations.

Initially, we do not intend to consider this issue, because, for the most part, our intended application does not have these kinds of critical time constraints. In other words, we are assuming that the cost of planning before execution is the same as the cost of planning during execution.

A second consideration that bears on deferral decisions is the accuracy of cost and probability of success models. Suppose that achieving a particular subgoal clause is expected to be easy, but turns out to be difficult or impossible when detailed planning is attempted. If planning for the clause is deferred, the planner would not discover the inaccuracy until after potentially irreparable changes have been made to the world. The result may be that the robot can no longer achieve its goal. As an example of this, suppose that the robot believes it will be easy to clean the drill press table and therefore defers planning for this contingency. After arriving at the drill press and setting everything up, suppose that the robot finds that the table is dirty. But now, access to the vacuum cleaner may be blocked by some other operation in the shop. As a result, cleaning the table could be hard or even impossible. However, if the robot planned for the cleaning subgoal ahead of time it might discover the potential difficulty in fetching the vacuum cleaner, and modify its plan accordingly.

As in the previous sections we would like to be able to write down an expected cost expression that will allow us to decide whether planning for a given clause should be deferred or not. Unfortunately, we do not have the necessary information to be able to do this. In order to describe the benefits of deferring planning for a clause we would need to have a model of the expected cost of generating a plan for achieving the clause. Unfortunately we don't have models of planning cost. The other factor that we need is the potential damage that might result from deferral if the condition turns out to be much harder to achieve than expected. Unfortunately, we have no information about the accuracy of our cost models. Ultimately, it may be necessary or desirable to provide such models to a planner. But for now we will use a simple intuitive approximation to decide when planning should be deferred for a condition.

Given a conjunction $e = pcq$, where c is the clause the planner is currently concerned with, we will assume that deferral of planning for c is advantageous whenever

$$A(p, s)E(c, g, s) \ll I(g).$$

According to this equation, when c is unlikely to be needed, or when c is easy to achieve, deferral is a sensible option. This matches our intuition that when the difficulty of achieving c increases, the planner needs to plan for c more carefully, to assure that c can actually be achieved.

For clauses in a conditional plan, the above expression needs to be changed slightly; in this case, we need to include the probability that the conditional branch will be taken. Thus, if the plan we are considering is $p\langle ?c, q, r\rangle$, we will assume that the planner should delay planning for q whenever

$$A(p, s)P(c, s)E(q, g, s) \ll I(g).$$

4 Final Remarks

In this paper we have laid the theoretical foundations for a decision theoretic approach to the control of search for a general purpose planner. We showed how this approach could be used to

1. choose between alternative actions for achieving a subgoal,

2. choose the order in which to plan for conjuncts in a conjunctive goal or subgoal,

3. decide when assumption, or the insertion of conditionals was desirable,

4. decide when to defer planning for subgoals.

As we have noted throughout the paper, there are a number of loose ends that still remain to be tidied. First of all, as mentioned in Section 3.2, our notation for talking about the state of the world after failure is not yet adequate. Second, some work remains to be done to expand the treatment of temporal and planning orders for conjunctions to the uncertain world. Third, our treatment of deferral is considerably less rigorous than we would like. A fourth problem is with our use of A, the achievability of a proposition. In writing all of our cost equations we have assumed that if an action did not achieve the desired result it would be noticed immediately. In reality this is not always true, especially for robots with limited sensory abilities. As a result, we need to change the meaning of A to refer to the probability that an action will be perceived as successful. It is not yet clear how this will effect the various equations.

But perhaps more important than these problems, are the many empirical questions raised by this research. Will it be practical to provide the appropriate cost and probability models needed in order to make use of these techniques? We speculate that it will, but this remains to be demonstrated. If so, how much overhead will be incurred in order to do the necessary expected cost computations? The overhead should add only a constant factor to each planning step, while the savings of this kind of control should be exponential. But, this also remains to be demonstrated.

Because these empirical questions are so crucial to the utility of this approach, we intend to direct our energy towards implementation of these techniques in a working planning system. As an application area we will be working with a system for disassembly and reassembly of simple electro-mechanical objects given models of their components and relationships.

Acknowledgements

Thanks to Jeff Finger for discussions (over many years) that lead to my appreciation of the difficulties involved in planning in an uncertain world. Discussions and work with Matt Ginsberg lead to the possible worlds approximation for determining what the world will look like after subgoals have been achieved. Thanks also to Mike Genesereth, Nils Nilsson, and Yoav Shoham, whose critical comments have forced me to be more careful about my terms and definitions.

This research was supported under ONR contract N00014-81-K-0004.

.

References

[1] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32(3):333–377, 1987.

[2] R. Fikes, P. E. Hart, and N. J. Nilsson. Learning and executing generalized robots plans. *Artificial Intelligence*, 3:251–288, July 1972. Also *SRI International* Technical Report.

[3] R. F. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[4] J. J. Finger. RESIDUE: A deductive approach to design synthesis. Technical Report STAN-CS-85-1035, Stanford University, 1985.

[5] J. J. Finger. *Exploiting Constraints in Design Synthesis*. PhD thesis, Stanford University, 1986.

[6] P. E. Friedland and Y. Iwasaki. The concept and implementation of skeletal plans. *Journal of Automated Reasoning*, 1(2):161–208, 1985.

[7] M. R. Genesereth and N. J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufman, Los Altos, 1987.

[8] M. L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, 1986.

[9] M. L. Ginsberg and D. E. Smith. Reasoning about action I: A possible worlds approach. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*. Morgan Kauffman, Los Altos, CA, 1987. Also in *Artificial Intelligence* 35(2), 1988.

[10] M. L. Ginsberg and D. E. Smith. Reasoning about action II: The qualification problem. *Artificial Intelligence*, 35(3):165–196, 1988.

[11] C. C. Green. Theorem proving by resolution as a basis for question-answering systems. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 183–205. American Elsevier, New York, 1969.

[12] R. Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33(1):65–88, 1987.

[13] D. V. McDermott. Flexibility and efficiency in a computer program for designing circuits. Artificial Intelligence Laboratory Memo AI-TR-402, Massachusetts Institute of Technology, 1977.

[14] D. V. McDermott. Planning and acting. *Cognitive Science*, 2(2):79–109, 1978.

[15] D. V. McDermott. Reasoning about plans. In J. Hobbs, editor, *Formal Theories of the Commonsense World*, pages 269–317. Ablex, Hillsdale, N.J., 1984.

[16] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5(2):115–135, 1974.

[17] E. D. Sacerdoti. The non-linear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pages 206–214, Tbilisi,USSR, Aug. 1975.

[18] E. D. Sacerdoti. *A Structure for Plans and Behavior*. American Elsevier, New York, 1977.

[19] Y. Shoham. Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence*, 33(1):89–104, 1987.

[20] D. E. Smith. Controlling inference. Technical Report STAN-CS-86-1107, Stanford University, Apr. 1986. PhD thesis.

[21] D. E. Smith. Controlling backward inference. *Artificial Intelligence*, 39(2):145–208, 1989.

[22] D. E. Smith and M. R. Genesereth. Ordering conjunctive queries. *Artificial Intelligence*, 26(2):171–215, 1985.

[23] R. Sproull. *Strategy Construction Using a Synthesis of Heuristic and Decision-theoretic Methods*. PhD thesis, Stanford University, 1977.

[24] M. J. Stefik. *Planning with Constraints*. PhD thesis, Computer Science Department, Stanford University, Jan. 1980.

[25] G. J. Sussman. *A Computer Model of Skill Aquisition*. American Elsevier, New York, 1975.

[26] A. Tate. Project planning using a hierarchical non-linear planner. D.A.I. Research Report 25, University of Edinburgh, Aug. 1976.

[27] A. Tate. Generating project networks. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 888–893, Cambridge, MA, Aug. 1977.

[28] R. Treitel. Sequentialization of logic programs. Technical Report STAN-CS-86-1135, Stanford University, Nov. 1986. PhD thesis.

[29] R. J. Treitel and M. R. Genesereth. Choosing directions for rules. *Journal of Automated Reasoning*, 3(4):395–431, 1987.

[30] R. J. Treitel and D. E. Smith. Effects of changing inference direction on the optimal ordering of a rule's antecedents. Technical Report LOGIC-87-12, Stanford University, Sept. 1987.

[31] D. H. D. Warren. WARPLAN: A system for generating plans. D.A.I. Research Report 76, University of Edinburgh, June 1974.

[32] D. E. Wilkins. Domain-independent planning: Representation and plan generation. *Artificial Intelligence*, 22(3):269–301, 1984.