An Empirical Study of Sensing and Defaulting in Planning

Kurt Krebsbach
Computer Science Dept.
University of Minnesota
4-192 EE/CSci Building
200 Union Street SE
Minneapolis, MN 55455
krebsbac@cs.umn.edu

Duane Olawsky
Math/CS Dept.
Macalester College
228A Olin Hall
1600 Grand Avenue
St. Paul, MN 55105
olawsky@macalstr.edu

Maria Gini
Computer Science Dept.
University of Minnesota
4-192 EE/CSci Building
200 Union Street SE
Minneapolis, MN 55455
gini@cs.umn.edu

Abstract

Traditional approaches to task planning assume that the planner has access to all of the world information needed to develop a complete, correct plan which can then be executed in its entirety by an agent. Since this assumption does not typically hold in realistic domains, we have implemented a planner which can plan to perform sensor operations to allow an agent to gather the information necessary to complete planning and achieve its goals in the face of missing or uncertain environmental information. Naturally this approach requires some execution to be interleaved with the planning process. In this paper we present the results of a systematic experimental study of this planner's performance under various conditions. The chief difficulty arises when the agent performs actions which interfere with or, in the worst case, preclude the possibility of the achievement of its later goals. We have found that by making intelligent decisions about goal ordering, what to sense, and when to sense it, the planner can significantly reduce the risk of committing to premature action. We have studied the problem both from the perspective of reversible and irreversible actions.

1 INTRODUCTION

The study described in this report constitutes part of an ongoing research project in the area of task planning under uncertainty. Traditional approaches to task planning assume that the planner has access to all of the world information needed to develop a complete, correct plan which can then be executed in its entirety by an agent. Of course, for most complex domains, having all of the necessary world information at plan time cannot be assumed. We have implemented a planner, BUMP, which is capable of interleaving planning and execution. BUMP is able to defer portions of the planning process which depend on unknown or uncertain information until the information in question can be obtained through sensors. In this case, BUMP inserts sensor operations directly into the plan which the agent executes to enable further planning.

Alternately, BUMP may choose to assume a default value for the uncertain information rather than plan to sense it. We call this distinction the defer/default question, and it has played a central role in guiding our recent research efforts.

Deferral and defaulting each have strengths and weaknesses. Deferral can be attractive with good sensors because it reduces planner uncertainty, however, sensing can become prohibitively expensive. In addition, satisfying preconditions for sensor operations can in itself be time-consuming, and as we will see, increases the probability of performing premature actions.

Defaulting can be risky, but it allows the planner to complete more of the plan before execution begins. This allows the planner to see further into the plan and detect problems which may lie beyond the horizon of the deferral point. As domain uncertainty increases however, further planning becomes increasingly arbitrary.

In general, it is difficult to know whether to defer and sense a given uncertain value or simply choose a default value and face the risks. Deciding on the best strategy for a given planning problem consists of computing the tradeoffs of various strategies, but as we will see, such a computation quickly becomes intractable for even a modest degree of uncertainty, suggesting the need for heuristic techniques.

2 PURPOSE OF STUDY

The experiments described here were designed to answer general questions about factors that influence the plan quality, and how to use those factors when deciding on planning strategies. In this section we discuss three types of planning strategies and three measures of plan quality. It is useful to think of the quality measures as functions to minimize or maximize, and the strategies as means to that end.

2.1 PLANNING FACTORS

We have identified the following as important factors of task planning with sensors. Our studies have shown that by intelligently controlling these factors, a planner can improve its performance, often dramatically. Thus, we define an *overall planning strategy* as a set of algorithms to determine each of the following parameters for a given problem instance.

Goal ordering: The initial goal ordering describes the order in which BUMP will attempt to construct portions of the plan to satisfy each goal. This ordering is fixed when planning commences and does not change. It is important to note however, that this order is not necessarily the order of execution. BUMP is fairly good at ordering and reordering actions to exploit helpful goal interactions and avoid harmful ones. Choosing an initial goal ordering to facilitate intelligent action reordering is one way to improve BUMP's performance. We found it advantageous to carefully order the planner's initial goals based on the amount and type of unknown information at the start of planning. Thus, we varied and examined goal orders to determine heuristics for a *goal ordering strategy*.

When to sense: A critical decision when interleaving planning and execution is *when* to switch from one to the other. In related research, Olawsky and Gini [1990] identified two general strategies to manage the transfer of control between the planning and execution modules (i.e., *control strategies*). In each strategy, if the planner discovers that it requires unknown information it inserts a sensor operation into the plan to obtain the information. It then plans to satisfy any preconditions of this sensor operation.

In the first strategy, known as Stop and Execute (SE), when the planner encounters a goal whose achievement depends on information it has planned to sense, control is transferred to the execution module. The sensor process and all processes ordered before it in the current partial plan are executed. Control then returns to the planner.

In the second strategy, Continue Elsewhere (CE), goals whose achievement depends on information to be sensed are deferred. Planning continues elsewhere. Only when all goals are either planned to completion or deferred does execution initially commence. Execution halts after each sensor operation to allow completion of a deferred goal. In general, CE allows much more planning, albeit less informed planning, to occur ahead of the first execution phase.

We believe these two strategies to be of particular interest because they seem to be the only truly *domain independent* control strategies which we have found useful. Any other strategies we have considered are not general-purpose, and are only useful under rather specific circumstances.

What to sense: Finally, there is the question of *which* uncertain quantities to sense and which to default. We will refer to this as the *deferral strategy*. Choosing a reasonable deferral strategy requires careful consideration of domain-specific factors such as default reliabilities, sensor reliabilities, planning costs, execution costs, and the cost of human intervention.

2.2 PLAN QUALITY CRITERIA

Before discussing the strategies above, we must be clear as to the objectives they are intended to serve. We call these objectives *plan quality criteria*, and we have identified and gathered data on a number of them, each of which could stand alone, or be used in conjunction with other criteria as a measure of plan quality.

Success Rate: For this measure of planner performance, we computed the *percentage of problems* in which BUMP was able to construct plans in which no processes needed to be undone as a result of being executed prematurely. One of the major challenges in interleaving planning and execution is to keep the robot from performing actions which may interfere with goals not yet considered. The most common example of this in our experiments occurred when the robot bolted closed a tool box only to discover that it contained a wrench (or bolt) needed to accomplish a later goal. Under this criterion we considered such plans failures, in effect assuming the agent was unable to recover from such premature action.

Execution Cost: For this criterion we measured the cost of all actions to be performed in the final plan when the planner is allowed to recover from premature action (i.e. undo and redo these actions). This provided us with some indication of *how* inefficient the inferior solutions were to previously unsuccessful problems. For these experiments, we simply counted each instantiated process (action) in the plan as having unit cost, although it would be trivial to assign varying costs to various types of actions.

Planning Cost: Finally, in some experiments we tracked the amount of planning work done by the planner. Since BUMP is an agenda-based planner, a reasonably accurate indication of planning work is the number of items it placed on its agenda.

When the quality criterion is the success rate, we assume there are no deadlines which must be met by the planner in order to succeed. In domains in which such deadlines are important, execution and planning cost should be used as quality criteria.

3 EXPERIMENTS IN THE TOOL BOX WORLD

Each experiment consisted of running BUMP on a carefully controlled set of problems. We attempted to select subsets of problems which were especially prone to premature action, and study BUMP's performance in solving each of them. In the near future we plan to conduct a study in which subsets are randomly constructed to examine BUMP's average case performance as well.

The experiments consisted of problems in the *tool box world*. In this world, the robot is in a room with n tool boxes, each containing wrenches and bolts of various sizes. The robot knows the initial locations of the wrenches and bolts. Bolts are identified by a unique name, and wrenches are identified by size. The robot has been instructed to close and bolt one or more tool boxes with particular bolts. To perform each bolting operation, the robot must use a wrench of a size that matches the bolt. A sensor is available that can classify bolts by their size (e.g., a number from 1 to 10). For simplicity, the bolt sizes are indicated along the same scale as the wrench sizes. We also assume the robot has a tool belt into which it can put an unlimited number of bolts and wrenches.[1]

The test set for our experiments varied slightly from one experiment to the next, but there are a number of characteristics shared by most of them. More detailed descriptions of the experiments can be found in [Krebsbach *et al.*, 1991]. The majority of the studies deal with a three-box world. These boxes are called S, T, and U, and they are to be bolted with bolts b_s, b_t and b_u, respectively. Each of these three bolts has a different size—b_s has size 4, b_t size 5, and b_u size 6. The bolts are initially in their respective boxes (e.g., b_s is in box S).[2] All of the tool boxes are initially open. In all of the experiments described in this paper the robot begins at a neutral site (one unrelated to any work that it must do). Since the planner's goals are strongly associated with particular tool boxes, this assumption was meant to avoid any bias in our results.

Table 1: Summary of Experiments for Success Rates, Execution Cost, and Planning Cost Criteria.

Exp	Boxes	Quality	Control Strategy	Goal Orders	Num of Unknowns
1	3	success	SE	all	all
2	3	success	CE	all	all
3	3	e-cost	SE	all	all
4	3	e-cost	CE	all	all
5	3	p-cost	SE	all	all
6	3	p-cost	CE	all	all
7	4	e-cost	SE	4	1
8	4	p-cost	SE	4	1
9	4	e-cost	SE	6	2
10	4	p-cost	SE	6	2

Each experiment consists of hundreds or thousands of planner runs using systematically defined sets of initial conditions, goal orderings and planning strategies. The variables defining these test sets are the following:

Wrench Location: Each of the wrenches may initially be in any tool box. For three box experiments this implies 27 possible wrench placement scenarios.

Goal Ordering: We studied the effect of reordering the initial goals on the performance of the planner. For three box experiments, this involves 6 possible orderings.

Control Strategy: SE versus CE.

Defer/Default Decisions: The size of each bolt is either known or unknown at the start of the first planning phase. For three bolts there are 8 combinations.

Table 1 provides a short summary of the ten experiments conducted. Complete experimental data can be found in [Krebsbach *et al.*, 1991].

4 MAJOR RESULTS

In this section we will outline the major results of the experiments, and principles and heuristics we developed based on the results. The results of experiments

[1] We are not concerned here with the arm-empty conditions as used in typical definitions of the blocks world. Our main goal in defining this domain is to study how sensor use can be interleaved with planning.

[2] This causes the robot to see less of the world while solving its early goals since it need not go anywhere to get a bolt. While this may at first appear to simplify the problem, in effect it tests the planner on a more difficult set of problems than it would by chance. The more places BUMP travels to to get bolts, the more of a chance it has to gather other information, quite possibly information it could use to make more informed action ordering decisions. This in turn would decrease BUMP's vulnerability to failures due to premature action.

Table 2: Experiment 1 (3 Box, Stop and Execute, Success Based).

	456	45-	4-6	-56	4--	-5-	--6	---
STU	100	63	63	100	37	63	63	37
TSU	100	63	100	63	63	37	63	37
TUS	100	63	100	63	63	37	63	37
SUT	100	63	63	100	37	63	63	37
UST	100	100	63	63	63	63	37	37
UTS	100	100	63	63	63	63	37	37
Avg	100	75	75	75	54	54	54	37

Table 3: Experiment 3 (3 Box, Stop and Execute, Execution Cost Based).

	456	45-	4-6	-56	4--	-5-	--6	---
STU	20.5	24.5	24.8	22.1	29.1	26.1	26.4	30.8
TSU	20.5	24.5	22.1	24.8	26.1	29.1	26.4	30.8
TUS	20.5	24.8	22.1	24.5	26.4	29.1	26.1	30.8
SUT	20.5	24.8	24.5	22.1	29.1	26.4	26.1	30.8
UST	20.5	22.1	24.5	24.8	26.1	26.4	29.1	30.8
UTS	20.5	22.1	24.8	24.5	26.4	26.1	29.1	30.8
Avg	20.5	23.8	23.8	23.8	27.2	27.2	27.2	30.8

1 and 3 are provided in Tables 2 and 3 respectively. In each table, the headings along the horizontal axis indicate which of the three bolt sizes are known in the order b_s, b_t, and b_u respectively. So, for instance, 4-6 indicates that bolt b_s is of size 4, bolt b_t is of unknown size, and bolt b_u is of size 6. Vertical lines separate columns into groups with the same number of unknowns. The labels along the vertical axis denote goal orderings. For example, an ordering of TSU means the initial goal involving box T was attempted first, followed in turn by the S and U goals.

One immediate observation from Tables 2 and 3 is that more unknown information means decreased success and increased cost. Certainly the planner will be more likely to perform premature actions with less a priori information. This general trend continued throughout all of the experiments.

4.1 CHOOSING A GOAL ORDERING

One of the major results of this study was that most sensing should come as early as possible in the plan. The disadvantage of potentially premature action caused by early sensing was, in most cases, outweighed by the advantage of constructing most of the plan with more information.

4.1.1 Ordering To Maximize Success

Consider Table 2, in particular, experiments STU 45- and STU --6. In the former, there is only one unknown, b_u. In the latter, there are two, b_s and b_t, however BUMP performs at the same 63% level for both. Also, average success rates in columns with the same number of unknowns are identical, however, the percentages in each column are distributed differently by goal ordering. For instance, in each column with one unknown there are two goal orderings which produce 100% success, and four which produce only 63% success, but the goal orderings are different in each column. Both of these behaviors are the result of a single underlying principle.

To understand this behavior, we consider an example more closely. Note that 100% success can be achieved in column 45- (of Table 2) by ordering the U-goal first (either as UST or UTS). We hypothesized that in cases where the size of b_u was unknown, it was crucial to BUMP's success to know the size of b_u early in the planning process. This could be accomplished by reordering goals so that the *U-goal* was attacked first. If this was not done, the goal involving b_u would be one of the last two BUMP would try to accomplish. Therefore, it would not sense the size of b_u until later, increasing the chances that achieving it would involve undoing some actions which had already been executed. Since planning and execution are interleaved, some execution is very likely to have been performed by the time BUMP encounters its later goals. If any of the executed actions involve bolting closed a box containing a needed wrench for b_u, the plan will no longer be successful.[3] Experiment 1 confirmed our suspicions that it is possible to improve average performance by controlling the goal ordering based on which information is missing for a given problem. The following heuristic describes the optimal ordering:

Success-Based Ordering Heuristic: When there are goals whose achievement depends only on known information, and other goals which depend on unknowns, order all goals involving unknowns before those involving only knowns.[4]

The same general principle applies to Continue Elsewhere.

[3] In general, more specialized strategies are probably necessary to avoid such problems. We have performed some experiments using a strategy called Sense Before Closing, in which all sensor processes are ordered before all closing operations. This solves the problem, but often introduces severe costs of its own. In the worst case, each tool box would have to be visited twice instead of once, so Sense Before Closing trades bolting/unbolting operation costs with transportation costs. Whether this is a good trade of course depends on the domain.

[4] This goal ordering heuristic depends critically on the assumption that the planner can identify connections between its top-level goals and the unknown domain propositions in the problem. In these experiments there is a one-to-one correspondence between goals and potential unknowns, so the issue is not addressed. Another related issue not addressed here is what should be done when goals rely on differing numbers of unknowns.

4.1.2 Ordering to Minimize Cost

Similar observations on goal ordering can be made when cost is the quality criterion. From Table 3 we can see that the highest cost occurs when a known goal is considered first. However, looking at the 4-- column, cost is minimized when there is an unknown first and an unknown last.[5] The goal orderings TSU and UST are both examples of this. We compared the plans generated with the TSU and TUS goal orderings to determine the cause of this behavior. As shown in Table 2 there is no difference in the number of plans involving premature actions for these two goal orderings. The slight difference in cost results from the way in which SE breaks up the planning work into phases. With the TSU goal ordering, BUMP plans the entire S-goal as soon as it obtains the sensor reading for bolt b_t. In several problems this allows BUMP to do two things while it is at S: to get wrench 5 and to close box S. This allows it to complete its task with only one trip to box S. When the TUS ordering is used, BUMP does not plan the S-goal until it has already closed box T and sensed bolt b_u. If wrench 5 (the one needed to close box T) is in box S, the robot must make one trip to S to obtain wrench 5 and a second trip after sensing b_u to close box S. This extra goto operation accounts for the increased cost.

To better understand this behavior we conducted experiments 7 through 10 using 4 boxes (SE control strategy only). Complete results of these experiments are described in [Krebsbach *et al.*, 1991]. These results are summarized by the following heuristic:

Cost-Based Ordering Heuristic: When there are goals whose achievement depends only on known information, and other goals which depend on unknowns:

1. place one unknown in the first position,
2. place one in the final position (if possible),
3. place any other unknowns following the first one,
4. place all knowns in the remaining positions.

4.2 CHOOSING A CONTROL STRATEGY

We found the CE strategy to be more susceptible to small increases in uncertainty, performing better than SE with one unknown, usually worse with two, and markedly worse with three. CE's sensitivity to unknown information makes sense when one considers CE's main advantage and disadvantage. Its advantage is that it performs more planning prior to the first execution cycle. This reduces the risk of performing premature actions if there are few unknowns, because BUMP can see further into the plan and perform action reordering to avoid conflicts it wouldn't detect with SE until it's too late. However, as uncertainty increases this further planning becomes less informed, and ordering decisions become more arbitrary, increasing the probability of performing premature actions which lead to failure or severe cost penalties. For instance, in the case of 3 unknowns, BUMP using CE was able to find successful plans in only 22% of the 3 box problems, as compared with 37% for SE.

[5] This ordering tied for best in the success-based case.

5 A SUCCESS-BASED OVERALL STRATEGY

Let us now make a first attempt at our goal of finding a good overall planning strategy. In addition to the ordering heuristic we must have a method for selecting a control strategy and a deferral strategy. We will try to maximize success through our selection of a strategy. We will assume here that once a control strategy and a deferral strategy have been selected, the top-level goals are reordered to obtain the highest success rate for the given number of unknowns.

As we have shown, we can always improve success rates by having additional known information. Thus, if our default information were 100% reliable, it would always make sense to use it and obtain a 100% success rating (with either control strategy). Of course, default information is rarely, if ever, 100% reliable. (We define reliability of a value to be the probability that it is correct. No notion of amount of error or distance from the correct value is considered.) If incorrect default information is used, the robot will most likely encounter an execution time error. This will necessitate some sort of execution time error recovery, and the resulting execution will certainly be inefficient. We consider this a failure. So, the increased success with extra "known" information must be adjusted by the reliability of that information. A similar point can be made regarding sensor reliability.[6] The data in all of our experiments assume that all sensor readings are correct, and this is clearly fictional.

5.1 ANALYSIS

To make this discussion more concrete, let us analyze the expected success rates given the reliability of our default values and our sensors. Let r_1, r_2 and r_3 be the reliabilities of our three defaults, d_1, d_2 and d_3 and let s_1, s_2 and s_3 be the reliabilities of the associated sensor readings. Also, assume $r_1 \geq r_2 \geq r_3$. (d_1, d_2, and

[6] Note that the relative reliability of defaulted information does not affect the optimal goal ordering from a success-based perspective, since BUMP will fail if any of the defaults are incorrect, regardless of when they are used. The same applies to the relative reliabilities of sensed information. To obtain an optimal goal ordering, it is only important that the success-based ordering heuristic be followed.

Table 4: Success-Based Strategies.

Take	$\sigma = 1.0$	$\sigma = 0.8$	$\sigma = 0.6$
d_1	$r_1 \geq 0.55$	$r_1 \geq 0.44$	$r_1 \geq .33$
d_1, d_2	$r_2 \geq 0.67$	$r_2 \geq 0.54$	$r_2 \geq .40$
d_1, d_2, d_3	—	$r_3 \geq 0.80$	$r_3 \geq .60$

d_3 are in no particular order relative to the planning process.) When a bolt size is known at the start of planning, this corresponds to a default reliability of 100%. Let q_0, q_1, q_2 and q_3 be the maximum potential success rates for cases with 0, 1, 2 and 3 unknowns respectively. From our experiments, these values are 1.0, 1.0, 0.67 (with CE), and 0.37 (with SE).

We can now calculate the expected success rate taking into account the default and sensor reliabilities. For example, the success rate when taking default d_1 and sensing the other unknowns is $r_1 s_2 s_3 q_2$. The best overall strategy in any particular instance of the three box problem can be found by computing the maximum of the following set of values:

$$\{r_1r_2r_3q_0, r_1r_2s_3q_1, r_1s_2r_3q_1, s_1r_2r_3q_1,\\ r_1s_2s_3q_2, s_1r_2s_3q_2, s_1s_2r_3q_2, s_1s_2s_3q_3\}$$

Once the maximum is found, the associated deferral strategy consists of the default/defer decisions indicated.

It is interesting to examine the default reliabilities required in the three box domain. For simplicity, assume that all sensor readings have the same reliability σ. Table 4 shows for various values of σ how reliable the defaults must be to make them worth taking. Looking at the column labeled $\sigma = 1.0$, we note that if the best default has reliability ≥ 0.55 it is better to take that default than to use a 100% reliable sensor. If $r_1 \geq r_2 \geq 0.67$, it is better to take two defaults. For more realistic values of σ, we see that the defaults need not be very reliable at all. This is due to the reduction in premature actions that can be avoided by having more knowledge early in the planning process.

6 A COST-BASED OVERALL STRATEGY

When the robot is able to detect at some point that a default value or a sensor reading was erroneous and then take corrective actions, it makes more sense to use cost as the quality criterion. As described earlier, cost can be measured either in terms of execution cost or planning effort. We will focus on execution cost since we believe this is generally the more significant aspect. A similar analysis could be developed for planning effort.[7]

6.1 ANALYSIS

The analysis in this case is a good deal more complicated since many more options come into play. For example, if a decision is made to try a default which later turns out incorrect, the robot could then try to recover by using a sensor. If the sensor reading also turns out to be incorrect, it might still be possible to recover with human intervention (presumably at a very high cost).[8]

As before, let r_i be the reliability of a default value and s_i the reliability of the sensor reading. In place of the success rates q_i that we used in our previous analysis, we need the average execution costs under various scenarios. We define the function C_i to return these costs when there are i unknowns. C_i takes i arguments where each argument is a sequence of one, two or three of the letters D, S and I. This sequence reflects which of the resources — default, sense and intervene — were used for the given unknown as well as the order in which they were tried. It is assumed that the last resource is always successful and that intervention is always successful. For example, DS means an incorrect default followed by a correct sensor operation. SDI means an incorrect sensor reading followed by an incorrect default value followed by successful human intervention. $C_1(SDI)$ would be the expected execution cost under this scenario when there is one unknown.

Given this information we can develop formulas for the expected costs of various attempted solutions. For example, with one unknown the expected cost of defaulting with sensing and intervention as backup actions is expressed by the following weighted sum:

$$r_1C_1(D) + (1-r_1)s_1C_1(DS) + (1-r_1)(1-s_1)C_1(DSI)$$

[7] The cost of recovering from incorrect information (sensed or defaulted) depends on the state of the environment when the false information is discovered. Thus, the optimal goal ordering from a cost-based perspective may be influenced by the relative reliabilities of the information in question. This is in contrast to our observations regarding optimal success-based goal ordering. We are currently investigating this question.

[8] Some other options that we do not consider in this analysis are

1. to try a different sensor, or
2. to continue trying the same sensor.

If the sensor is working at all (i.e., there is a non-zero probability of a correct reading), then with persistence the second option should eventually produce a correct reading. The probability of n readings all being incorrect goes to 0 as $n \to \infty$. This might also have a very high cost. The same analysis technique could be used to characterize the cost of both these options.

Table 5: Sample C_1 Values.

D	20	S	22
DS	30	SD	30
DI	40	SI	42
DSI	50	SDI	50
I	35		

Table 6: Expected Costs.

Strategy	$r_1 = 0.7$ $s_1 = 0.8$	$r_1 = 0.2$ $s_1 = 0.8$
default, sense, intervene	**24.2**	31.2
sense, default, intervene	24.8	26.8
sense, intervene	26.0	**26.0**
default, intervene	26.0	36.0
intervene	35.0	35.0

Assuming the C_1 values shown in Table 5 and the reliabilities $r_1 = 0.7$ and $s_1 = 0.8$ the expected cost is 24.2.

An alternative strategy would try sensing first followed by defaulting and then intervention. The weighted sum cost formula for this strategy is

$$s_1C_1(S)+(1-s_1)r_1C_1(SD)+(1-s_1)(1-r_1)C_1(SDI)$$

There are three other strategies in which one or more of the resources is not tried. The expected costs are

no default: $s_1C_1(S) + (1 - s_1)C_1(SI)$

no sensing: $r_1C_1(D) + (1 - r_1)C_1(DI)$

neither default nor sense: $C_1(I)$

One of these strategies might be appropriate if sensing or defaulting is particularly unreliable and the cost of intervention is light.

Given the C_1 cost estimates and the reliabilities, we can calculate the optimal strategy for one unknown by evaluating the five above formulas and finding the minimum. The expected costs under the five strategies, assuming the costs in Table 5 and the reliabilities $r_1 = 0.7$ and $s_1 = 0.8$, are shown in the second column of Table 6. In this case, the best strategy is default, sense then intervene. If on the other hand the reliability of the default is 0.2 we get the costs shown in column 3. Here, the best strategy is to sense then intervene. Note that it is better in this case to ask immediately for intervention than to try a default and then request help if there is a problem. The default is not reliable enough to risk the extra cost associated with an incorrect guess and the cost of intervention is small.

Let us next consider the formulas for expected cost with two unknowns. One scenario would try both defaults first, backed up by sensing and intervention. The resulting formula is

$$\begin{aligned}
&r_1r_2C_2(D, D) + \\
&\quad (1 - r_1)r_2s_1C_2(DS, D) + \\
&\quad r_1(1 - r_2)s_2C_2(D, DS) + \\
&\quad (1 - r_1)(1 - r_2)s_1s_2C_2(DS, DS) + \\
&\quad (1 - r_1)r_2(1 - s_1)C_2(DSI, D) + \\
&\quad r_1(1 - r_2)(1 - s_2)C_2(D, DSI) + \\
&\quad (1 - r_1)(1 - r_2)(1 - s_1)s_2C_2(DSI, DS) + \\
&\quad (1 - r_1)(1 - r_2)s_1(1 - s_2)C_2(DS, DSI) + \\
&\quad (1 - r_1)(1 - r_2)(1 - s_1)(1 - s_2)C_2(DSI, DSI)
\end{aligned}$$

This formula is certainly much more complicated than the formulas for one unknown. In fact the number of terms to be summed in a formula that considers all three resources — default, sense and request intervention — grows exponentially (3^n for n unknowns). The number of factors in the longest term is $2n + 1$. Thus, calculating the expected cost of just one scenario is $O(n3^n)$. Even the amount of cost data that must be collected grows exponentially in the number of unknowns. There are many other scenarios that must be evaluated and compared to this one to find the optimal strategy of sensing and defaulting.

Clearly, we cannot effectively calculate this optimal strategy unless the number of unknowns is quite small. Rather, we need heuristic techniques that will help us find an approximately optimal strategy. Finding such techniques will be a subject of our future research.

7 DISCUSSION

A well recognized problem with planning is the inability of most planners to deal with the inexactness and noise of the real world.

Several solutions have been proposed that range from eliminating planning altogether in favor of reactive planning [Brooks, 1986] or situated systems [Agre and Chapman, 1987, Kaelbling, 1988], to combining reactivity and planning [Georgeff and Lansky, 1987, Sanborn and Hendler, 1988], to interleaving planning with execution [McDermott, 1978, Durfee and Lesser, 1986, Chapman, 1991], to preplanning for every contingency [Schoppers, 1987], to verifying the executability of plans and adding sensing whenever needed to reduce the uncertainty [Brooks, 1982, Doyle *et al.*, 1986].

Brooks [1982] verifies the feasibility of a plan in light of uncertainties and errors and decides when sensors are needed to reduce the amount of error. Doyle et al. [1986] use sensors to verify the execution of a plan. The sensor requests are generated after the plan has been produced by examining the preconditions and postconditions of each action in the plan. Domain dependent verification operators map assertions to perception requests and expectations. The entire process is done before executing the plan. Hager and Mintz

[1991] have more recently proposed methods for sensor planning based on probabilistic models of uncertainty.

Few have addressed the more specific problem we address. Our work has been inspired, among others, by the work of Turney and Segre [1989], who alternate between improvising and planning. Since sensing is assumed to be expensive, their system prefers actions with the fewest sensor requests first. The results obtained show the importance of good heuristics over sophisticated planning strategies. The quality of the heuristic improvisation strategy has the largest effect on the quality of the solution. This seems to suggest that it is more important to develop good heuristics than to develop a sophisticated planner.

The need to plan with incomplete information raises important theoretical issues. A number of authors have proposed decision theoretic approaches to planning and control. Horvitz et al. [1989] propose a general model for reasoning under scarce resources that is based on decision theory [Dean, 1990]. Chrisman and Simmons [1991] produce near optimal cost plans by using Markov Decision Processes to decide what to sense.

Drummond and Bresina [1990] propose an algorithm that maximizes the probability of satisfying a goal. More recent work of Minton et al. [1991] analyzes in a rigorous way a linear and a non-linear planner in terms of their overall efficiency, examining both search space complexity and time cost. Hsu [1990] proposes to plan with incomplete information by generating a "most general partial plan" without committing to any choice of actions not logically imposed by the information available at that point. She uses an anytime algorithm [Dean and Boddy, 1988] to choose the appropriate action on the current partial plan when the system has to act.

Acknowledgement

We would like to thank Mark Boddy for useful and influential discussions about topics in this paper, most notably issues of cost analysis and decision theory.

References

[Agre and Chapman, 1987] Philip Agre and David Chapman. Pengi: A theory of activity. In *Proc. of AAAI-87*, pages 268–272, Washington, July 1987.

[Brooks, 1982] Rodney A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1:29–69, 1982.

[Brooks, 1986] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[Chapman, 1991] David Chapman. Combinatorics and action. In *Working Notes of the AAAI Fall Symposium on Sensory Aspects of Robotics Intelligence*, pages 26–30, 1991.

[Chrisman and Simmons, 1991] Lonnie Chrisman and Reid Simmons. Sensible planning: Focusing perceptual attention. In *Proc. of AAAI-91*, Los Angeles, CA, 1991.

[Dean and Boddy, 1988] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proc. of AAAI-88*, St. Paul, MN, 1988.

[Dean, 1990] Thomas Dean. Planning under uncertainty and time pressure. In *Proceedings of the DARPA Workshop on Planning*, pages 390–395, San Diego, CA, November 1990.

[Doyle *et al.*, 1986] Richard Doyle, David Atkinson, and Rajkumar Doshi. Generating perception requests and expectations to verify the execution of plans. In *Proc. of AAAI-86*, pages 202–206, Philadelphia, 1986.

[Drummond and Bresina, 1990] Mark Drummond and John Bresina. Anytime synthetic projection: maximizing the probability of goal satisfaction. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, Mass, August 1990.

[Durfee and Lesser, 1986] Edmund H. Durfee and Victor R. Lesser. Incremental planning to control a blackboard-based problem solver. In *Proc. of AAAI-86*, Philadelphia, PA, 1986.

[Georgeff and Lansky, 1987] Michael P. Georgeff and Amy L. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, WA, 1987.

[Hager and Mintz, 1991] G. Hager and M. Mintz. Computational methods for task-directed sensor data fusion and sensor planning. *International Journal of Robotics Research*, 10:285–313, 1991.

[Horvitz *et al.*, 1989] Eric J. Horvitz, Gregory F. Cooper, and David E. Heckerman. Reflection and action under scarce resources: theoretical principles and empirical study. In *Proceedings of IJCAI-89*, pages 1121–1127, Detroit, MI, August 1989.

[Hsu, 1990] Jane Yung-jen Hsu. Partial planning with incomplete information. In *Proceedings of the AAAI Spring Symposium on Planning in Uncertain, Unpredictable, or Changing Environments*, 1990.

[Kaelbling, 1988] Leslie Pack Kaelbling. Goals as parallel program specifications. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 60–65, St. Paul, MN, August 1988.

[Krebsbach *et al.*, 1991] Kurt Krebsbach, Duane Olawsky, and Maria Gini. Deferring task planning in the tool box world: Empirical results. Technical Report TR 91-60, University of Minnesota Department of Computer Science, Minneapolis, MN, 1991.

[McDermott, 1978] Drew McDermott. Planning and acting. *Cognitive Science*, 2:71–109, 1978.

[Minton *et al.*, 1991] Steven Minton, John Bresina, and Mark Drummond. Commitment strategies in planning: a comparative analysis. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 259–265, Sydney, Australia, August 1991.

[Olawsky and Gini, 1990] Duane Olawsky and Maria Gini. Deferred planning and sensor use. In *Innovative Approaches to Planning, Scheduling and Control: Proceedings of the DARPA Workshop on Planning*, pages 166–174, San Diego, CA, November 1990.

[Sanborn and Hendler, 1988] James Sanborn and James Hendler. A model of reaction for planning in dynamic environments. *Artificial Intelligence in Engineering*, 3(2):95–102, 1988.

[Schoppers, 1987] Marcel Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1039–1046, Milan, Italy, August 1987.

[Turney and Segre, 1989] Jennifer Turney and Alberto Segre. A framework for learning in planning domains with uncertainty. Technical Report 89-1009, Department of Computer Science, Cornell University, Ithaca, NY, May 1989.