Deferring Task Planning in the Tool Box World: Empirical Results

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

September 25, 1995

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 report 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 experiments described in this report constitute 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. For this reason, 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 have come to 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, this 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 Experiments

The experiments we performed were aimed at answering general questions about factors which influence the quality of a plan, and how we can use those factors when deciding on planning strategies. In this section we will discuss three types of planning strategies and three measures of plan quality. We have found it useful to think of the quality measures as functions to minimize or maximize, and the strategies as means to that end.

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: 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, [OG90] identified two general strategies to manage the transfer of control between the planning and execution modules. We examined these two *control strategies.*

For both strategies if the planner discovers that it needs some 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 depend 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, Continue Elsewhere 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 our *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.

Plan Quality Criteria

Before it makes sense to further discuss the strategies above, we must be clear as to the objectives they are intended to serve. We have come to call such objectives *plan quality criteria*, and 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 Rates: 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 difficulties 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 actions.

Execution Cost: For this criterion we measured plan length, that is, the cost of all actions to be performed in the final plan when the planner is allowed to recover from premature actions (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.

3 The Experiments

In this section we will outline the experiments we conducted. Each experiment consisted of running the planner, BUMP, on a number of carefully controlled planning problems. In most cases we attempted to decide on the most challenging subset of problems in the domain, generate them in turn, and measure BUMP's performance in solving each of them.

All of the experiments consisted of problems in the *tool box world*. In the tool box world, the robot is in a room with n tool boxes $t_1, t_2, \ldots t_n$, 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 bolts 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] Finally, we assume the robot's starting location to be distinct from the location of any of the tool boxes. Since the planner's goals are strongly associated with particular tool boxes, this assumption was meant to avoid any bias in our results.

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. We describe these commonalities here and then specify modifications with the individual studies. 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). Several of the experiments described here use a four-box world adding an extra box v and an extra bolt b_v. In other respects they are similar to the three-box experiments.

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:

[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.

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

Table 1: Summary of experiments conducted.

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

Goal Ordering: In several of the experiments we were interested in the effect that reordering the initial goals might have on the performance of the planner.

Control Strategy: Stop and Execute versus Continue Elsewhere.

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.

Results of the experiments will often be broken down to show the effects of goal ordering, control strategy and defer/default decisions. A summary of the experiments we conducted can be found in Table 1.

3.1 3-Box Success-Based

In our first experiment, we measured plan *success* by computing 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. In each problem there are three top level goals. Each goal is of the form (bolted t b_t), where t denotes a tool box instance, and b_t a bolt instance. For simplicity, we will identify goals by the tool box they refer to (e.g., the *T-goal* for the above goal).

Table 2 provides the results of this experiment. The standard test set described earlier was used. 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,

Stop and Execute

	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

Continue Elsewhere

	456	45-	4-6	-56	4--	-5-	--6	---
STU	100	63	74	100	30	44	67	22
TSU	100	63	100	74	44	30	67	22
TUS	100	74	100	63	67	30	44	22
SUT	100	74	63	100	30	67	44	22
UST	100	100	63	74	44	67	30	22
UTS	100	100	74	63	67	44	30	22
Avg	100	79	79	79	47	47	47	22

Table 2: Success Rates, 3 box, All Goal Orderings

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 headings above each table indicate the control strategy used, either Stop and Execute or Continue Elsewhere.

In the following sections we informally describe observations and principles based on the results of the success-based experiments. We conclude this section with a more formal analysis and propose an overall planning strategy for maximizing success.

More information means increased success

The results of experiments 1 and 2 are broken down by goal ordering and averaged in Table 2. They suggest a number of things. Let us look at the summary line of the SE portion of this table. As we move to the right in the table, the success rates get worse, as expected. Certainly the planner will be more likely to commit to premature action with less a priori information.

Get unknown information early

Again consider the SE table. 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, consider the three columns with one unknown. In each column, 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. In fact, each goal ordering produces 100% success in *exactly* one place in those three columns. The same applies to Continue Elsewhere.

It appears as though we can achieve the highest success rates in the one unknown case by considering the goal involving that unknown *first*. For example, we can achieve 100% success in column 45- by ordering the U-goal first (either as UST or UTS) because b_u is the bolt of unknown size. To explain the unknown b_u phenomenon we hypothesized that in the 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 in the planning process, increasing the chances that satisfying that goal would involve undoing some actions which had already been executed. Keep in mind that planning and execution are interleaved, and that 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] This experiment confirmed our suspicions that it is possible to improve average performance by controlling the goal ordering based on which information is missing for a particular problem.

Quantum Levels

In the case of Stop and Execute, it is clear that for a given problem pair of goal ordering and bolt size scenario, BUMP can perform at one of only 3 levels of success, 100, 63, or 37 percent. We will refer to these as *quantum levels* or *quanta*. Similar quanta can be seen in the Continue Elsewhere case, although there are 7 instead of 3 quanta. It seems from this experiment that we can

[3] It is useful to note here that 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 any closing operations are performed. This solves the problem, but carries with it often 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 basically trades bolting/unbolting operation costs with transportation costs. Whether this is a good trade of course depends on the domain.

move up a quantum level by ordering initial goals properly. The following two examples serve as evidence of this.

Consider the SE table again, 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% quantum level for both. Why? In the first case, the goal involving the unknown is considered last. In the second case, the two goals involving unknowns are considered first.

Consider a second example: STU 4 performs at the 37% quantum level and TSU 4 at 63%, although the bolt sizes for b_t and b_u are the unknowns in both cases. This difference makes sense when we consider the following. In STU the *S-goal* is tackled first. Since the size of bolt b_s is known the entire goal is finished, and then the T-goal is considered. It requires a sensor operation which will be ordered after the *S-goal* steps. (Since the bolt b_t is in box t, the robot must leave s and go to t.) That causes problems. With TSU, the *T-goal* is considered first and the sensor operation done immediately. This raises the success rate to the same as TSU 45-. In effect, by reordering intelligently it appears we can reduce the number of unknowns by one, but as we will see, the improvement is not quite that dramatic.

Continue Elsewhere

We see the same pattern with the CE control strategy.[4] Here however, it appears the CE strategy is more susceptible to small changes in degrees of uncertainty, performing better than SE with one unknown, usually worse but sometimes better with two unknowns, and markedly worse with three unknowns. Such a sensitivity to unknown information makes sense when one considers that BUMP commits to far more of the plan in general when using the CE strategy, which works to its advantage when there is little uncertainty, but causes many problems as the degree of uncertainty increases.

A useful heuristic

A very useful heuristic seems evident here: Try to consider the goals that you know the least about first. Sort the goals in increasing order of knowledge.[5] Using this heuristic should yield the predicted success rates in Table 3. It seems

[4] The CE 456 experiments were not actually run since they are, by definition, identical to SE 456. In the full information case, both strategies exhibit the same behavior.

[5] It is important to note that this 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 such issues are not addressed.

	456	45-	4-6	-56	4--	-5-	--6	---
SE	100	100	100	100	63	63	63	37
CE	100	100	100	100	67	67	67	22

Table 3: Potential Success Rates, 3 Boxes, Correct Ordering

from this experiment that SE and CE perform at close to the same level. CE is slightly better on average in the 2 unknown case (by 4%).[6] SE is 12 percentage points better in the zero information case. CE performs quite badly in the zero information case because it constructs nearly an entire plan before any information is ascertained through sensors.

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 that the top-level goals are reordered to obtain the highest success rate for the given number of unknowns.

As can be seen from Table 3 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. 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. So, the increased success with extra "known" information must be adjusted by the reliability of that information. Of course, a similar point can be made regarding sensor reliability. The data in Table 3 assumes that all sensor readings are correct, and this is clearly fictional.

3.2 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

[6] CE does better than SE in 4-6 and -56 when it can avoid locking wrench 6 in box S by planning past the b_t sensing to look at the *U-goal*. The same advantage can be seen in 4 and -5-.

the reliabilities of the associated sensor readings. Also, assume $r_1 \geq r_2 \geq r_3$. (d_1, d_2, and 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).

With one unknown we have two possible defaulting scenarios: default on that unknown or sense it. In the first case the expected success is $r_1 q_0$. In the second case it is $s_1 q_1$. Since $q_0 = q_1 = 1.0$, we should take the default if $r_1 \geq s_1$. We would normally not expect this to be the case, so with one unknown it is usually better to use a sensor. (This ignores the additional execution cost of sensing. This will be taken into account in our later cost based analyses.)

More generally, 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:

$$\{r_1 r_2 r_3 q_0, r_1 r_2 s_3 q_1, r_1 s_2 r_3 q_1, s_1 r_2 r_3 q_1,$$
$$r_1 s_2 s_3 q_2, s_1 r_2 s_3 q_2, s_1 s_2 r_3 q_2, s_1 s_2 s_3 q_3\}$$

Once the maximum is found, the associated deferral strategy consists of the default/defer decisions indicated. For example, $s_1 r_2 r_3 q_1$ corresponds to using defaults d_2 and d_3 and getting a sensor reading for the other unknown. (Either control strategy would be acceptable here.)

It is interesting to examine the default reliabilities required in the three box domain. For example, consider a case with two initial unknowns (i.e., $r_1 = 1.0$). Let us also assume 100% reliable sensors. The expected success rates are now

$$\{r_2 r_3, r_2, r_3, r_2 r_3, 0.67, 0.67 r_2, 0.67 r_3, 0.37\}$$

Assuming $1 > r_2$ (recall $r_2 \geq r_3$), the optimal strategy is clearly either the second (take d_2 but not d_3) or the fifth (take neither default). The former is preferred if $r_2 \geq 0.67$. That means a default that is correct two thirds of the time is better than a 100% reliable sensor reading. If the sensors are unreliable the default information looks even better.

A similar analysis for three unknowns (again assuming perfect sensors) gives the following strategy:

If $r_2 \geq 0.67$	take d_1 and d_2
elseif $r_1 \geq 0.55$	take d_1
else	sense everything

Here again, even mediocre defaults look attractive. Considering more realistic sensors with an 80% accuracy rate, we obtain the strategy

If $r_3 \geq 0.8$	take d_1, d_2 and d_3
elseif $r_2 \geq 0.54$	take d_1 and d_2
elseif $r_1 \geq 0.44$	take d_1
else	sense everything

3.3 3-Box Cost-Based

Execution Cost

After our initial experiments concerning control strategy and goal ordering, we ran the same experiments using a different plan quality criterion, namely *execution cost*. With this new measure of plan quality however, we must now allow the planner to produce inefficient plans if necessary, by adding redundant actions, or planning to undo and redo actions to achieve its goals.

With this new ability, BUMP was able to find a valid solution to each of the 2592 experiments we conducted.[7] Table 4 shows execution cost data broken down into individual goal orderings and summarized.

Both the SE and CE execution cost tables mirror the results obtained when performing the same experiments from a success standpoint. The average costs in each column within a given unknown group are identical. However, with this new quality criterion, more quanta are obtained. For SE, 8 quanta can be identified rather than 3. For CE, 8 quanta are also present, rather than 7. This suggests that a cost-based approach is more useful to us in revealing the true quality of plans that BUMP produces. For instance, groups of experiments which appeared equivalent from a success standpoint are distinguished from each other here. Plans which committed to premature action were all considered failures when measuring success, but can now be distinguished by *how* much inefficiency is introduced by premature action.

It is somewhat surprising just how much additional execution is introduced by premature actions caused by missing information. For SE, exactly 50% more processes are necessitated in the 3 unknown case than in the full information case. This number is nearly 61% for CE.

From a cost perspective, it is not possible however, to achieve the same quality rating as a case with one fewer unknown, as was the case with success rates. In other words, reordering doesn't quite eliminate the effect of one of the unknowns, as suggested by our success rate experiments. For instance, in the 3 box problem, optimal reordering shortened plans by an average of about 10%.

[7] 2592 = 2 control strategies x 8 bolt scenarios x 6 goal orderings x 27 wrench placement scenarios.

Stop and Execute

	456	45-	4-6	-56	4--	-5-	--6	---
SE STU	20.52	24.48	24.78	22.07	29.11	26.15	26.44	30.78
SE TSU	20.52	24.48	22.07	24.78	26.15	29.11	26.44	30.78
SE TUS	20.52	24.78	22.07	24.48	26.44	29.11	26.15	30.78
SE SUT	20.52	24.78	24.48	22.07	29.11	26.44	26.15	30.78
SE UST	20.52	22.07	24.48	24.78	26.15	26.44	29.11	30.78
SE UTS	20.52	22.07	24.78	24.48	26.44	26.15	29.11	30.78
Avg	20.52	23.78	23.78	23.78	27.23	27.23	27.23	30.78

Continue Elsewhere

	456	45-	4-6	-56	4--	-5-	--6	---
CE STU	20.52	24.78	24.11	22.44	29.33	28.00	26.67	33.00
CE TSU	20.52	24.78	22.44	24.11	28.00	29.33	26.67	33.00
CE TUS	20.52	24.11	22.44	24.78	26.67	29.33	28.00	33.00
CE SUT	20.52	24.11	24.78	22.44	29.33	26.67	28.00	33.00
CE UST	20.52	22.44	24.78	24.11	28.00	26.67	29.33	33.00
CE UTS	20.52	22.44	24.11	24.78	26.67	28.00	29.33	33.00
Avg	20.52	23.78	23.78	23.78	28.00	28.00	28.00	33.00

Table 4: Execution Costs, 3 Box, All Goal Orderings

Planning Cost

At the same time, we gathered data on our third quality criterion, *planning cost*. Here we simply kept track of the number of items placed on the agenda throughout the entire process of plan formulation and execution as our measure of planning cost. The average planning costs are shown in Table 5.

While plan length increased by 50% for SE as less information was available, planning cost increased more slowly. For SE, about 39% more agenda items are necessitated in the 3 unknown case than in the full information case. However, this number was 65% for CE. As stated previously, CE does very poorly when it has very little information to begin with, because it still constructs a large percentage of the plan prior to any sensing. This translates into large numbers of premature actions, which implies more planning and execution to recover from them.

Finally, when examining the two cost tables together a peculiar fact appears. In SE, cases that appeared symmetrical with respect to the success criterion (e.g.,

Stop and Execute

	456	45-	4-6	-56	4- -	-5-	- -6	- - -
SE STU	104.00	122.85	122.59	105.04	142.89	124.19	123.93	144.22
SE TSU	104.00	122.85	105.04	122.59	124.19	142.89	123.93	144.22
SE TUS	104.00	122.59	105.04	122.85	123.93	142.89	124.19	144.22
SE SUT	104.00	122.59	122.85	105.04	142.89	123.93	124.19	144.22
SE UST	104.00	105.04	122.85	122.59	124.19	123.93	142.89	144.22
SE UTS	104.00	105.04	122.59	122.85	123.93	124.19	142.89	144.22
Average	104.00	116.83	116.83	116.83	130.33	130.33	130.33	144.22

Continue Elsewhere

	456	45-	4-6	-56	4- -	-5-	- -6	- - -
CE STU	104.00	127.63	123.85	112.89	152.89	144.33	136.44	172.00
CE TSU	104.00	127.63	112.89	123.85	144.33	152.89	136.44	172.00
CE TUS	104.00	123.85	112.89	127.63	136.44	152.89	144.33	172.00
CE SUT	104.00	123.85	127.63	112.89	152.89	136.44	144.33	172.00
CE UST	104.00	112.89	127.63	123.85	144.33	136.44	152.89	172.00
CE UTS	104.00	112.89	123.85	127.63	136.44	144.33	152.89	172.00
Average	104.00	121.46	121.46	121.46	144.56	144.56	144.56	172.00

Table 5: Planning Costs, 3 Box, All Goal Orderings

SE TSU 4 and SE TUS 4) demonstrate a small difference in *execution cost*. Furthermore, it is not the TUS ordering that is shortest, as we might expect since the T and U goals involve unknown information, it is TSU. Moreover, when considering *planning cost* TUS takes less work! This may at first seem surprising, as one would expect more agenda items would generally imply more process nodes.

After investigating this phenomenon on a more detailed level, we found that the slight increase in execution cost for TUS was largely due to the fact that 7 plans contained one extra *goto process*. We saw examples of extra gotos in both recovery and non-recovery cases.[8] Thus it seems that ordering the only goal with known information last is slightly worse than ordering it second to last, since the planner does not make use of the known information in time to avoid a redundant action.

[8]It is possible for the plan to contain more goto processes than is strictly necessary without recovering. In this particular case, it occurs because, by considering the goal involving the known information last, it loses an opportunity to reorder actions in such a way as to avoid returning to a box it had been at previously.

	Plan Cost	Execution Cost
SE STUV	205.44	41.19
SE TSUV	190.75	38.76
SE TUSV	186.24	38.18
SE TUVS	188.80	38.84

Table 6: Planning and Execution Costs, 4-box, 3 unknowns, All Goal Orderings, Stop and Execute

3.4 4-box Cost-Based

The main purpose of extending the study to *four* tool box experiments was to determine the proper position for the one goal involving known information. Based on prior experiments, we suspected it wouldn't be first or last. This suspicion was confirmed.

Table 6 shows average planning and execution costs for planning problems involving four tool boxes. Because of the computational explosion of possible problems involving four tool boxes, we limited this experiment to problems in which only one bolt size is known ($b_s = 4$), and only four goal orderings are used.

From a planning cost and execution cost standpoint, the TUSV goal ordering seems to be the best, slightly better than TUVS and TSUV. This means attacking the known goal *third* gives the best average results. Goal ordering STUV is clearly the worst, as was expected, since the fact that the planner gets its first sensor operation for free is not taken advantage of by attacking the *S-goal* goal first. In the four box problems we conducted, attacking the known goal first results in an extra 19.2 agenda items (10.3% more planning) and 3.01 more processes (7.9% more execution) than the optimal ordering on average.

3.5 4-Box, 2 Unknowns, Cost-Based

To flesh out our understanding of the effects of goal ordering in the four box world we performed a comparative study of six goal orderings (see Table 7) using all 256 possible placements of wrenches in the four box world. We used the Stop and Execute strategy and took the sizes of bolts b_s and b_t to be known. We selected six goal orderings (from a possible 24) by noting that the S and T goals were symmetric (the bolt size was known for both of them) and so were the U and V goals. Thus, we selected the 6 orderings that have S before T and U before V.

As we hypothesized it was clearly better to have a goal with an unknown bolt

	Plan Cost	Execution Cost
SE STUV	189.25	37.01
SE SUTV	184.74	36.43
SE SUVT	187.30	37.09
SE USTV	163.68	32.98
SE USVT	169.78	34.25
SE UVST	162.96	33.39

Table 7: Planning and Execution Costs, 4 Box, 2 Unknowns, All Goal Orderings, Stop and Execute

size first. Given that we start with an unknown what is the best order for the remaining three goals? As noted in our earlier ordering experiments there are competing principles at work here. It is good to encounter unknowns early in the planning process, but it is also good to save one unknown for the end. These competing principles explain the data from this experiment too. Goal orderings USTV and UVST are roughly equivalent in performance with USVT performing slightly worse. USVT obeys neither principle: It finishes with a known and it has another known before an unknown. This suggests the following heuristic:

Ordering Heuristic: When there are goals with known information and goals with unknown information,

1. place one unknown in the first position,
2. if there are more unknowns, place one in the final position,
3. place any other unknowns in sequence following the first one.
4. all knowns go in the remaining positions.

4 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 of 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.

4.1 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).[9]

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:

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

Assuming the C_1 values shown in Table 8 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

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

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

[9] 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.

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

Table 8: Sample C_1 values.

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

Table 9: Expected costs.

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

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

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 8 and the reliabilities $r_1 = 0.7$ and $s_1 = 0.8$, are shown in the second column of Table 9. 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

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

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.

5 Discussion

Interleaving of planning and execution has been used and discussed extensively in robotics [McD78, DL86, Cha91] but few researchers have addressed the more specific problem of deciding what to sense and when. Our work has been inspired, among others, by the work of [TS89], who alternate between improvising and planning. Since sensing is assumed to be expensive, the system prefers actions with the fewest sensor requests first. The results they obtained show the importance of good heuristics over sophisticated planning strategies.

Brooks [Bro82] 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 [DAD86] uses 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 [HM91] have more recently proposed methods for sensor planning based on probabilistic models of uncertainty.

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. [HCH89] propose a general model for reasoning under scarce resources that is based on decision theory. Boddy [Bod91] has studied time-dependent problems and proposed a framework based on decision models for constructing solutions to time-dependent problems. Chrisman and Simmons [CS91] produce near optimal cost plans by using Markov Decision Processes to decide what to sense. Hsu [Hsu90] 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. An anytime algorithm [DB88] is used to choose the appropriate action on the current partial plan when the system has to act.

6 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

[Bod91] Mark Boddy. Solving time-dependent problems: A decision-theoretic approach to planning in dynamic environments. Tech Report CS-91-06, Brown University Dept. of Computer Science, 1991.

[Bro82] Rodney A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1:29–69, 1982.

[Cha91] David Chapman. Combinatorics and action. In *Working Notes of the AAAI Fall Symposium on Sensory Aspects of Robotics Intelligence*, pages 26–30, 1991.

[CS91] Lonnie Chrisman and Reid Simmons. Sensible planning: Focusing perceptual attention. In *Proc. of AAAI-91*, Los Angeles, CA, 1991.

[DAD86] 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.

[DB88] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proc. of AAAI-88*, St. Paul, MN, 1988.

[DL86] Edmund H. Durfee and Victor R. Lesser. Incremental planning to control a blackboard-based problem solver. In *Proc. of AAAI-86*, Philadelphia, PA, 1986.

[HCH89] 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.

[HM91] 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.

[Hsu90] 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.

[McD78] Drew McDermott. Planning and acting. *Cognitive Science*, 2:71–109, 1978.

[OG90] 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.

[TS89] 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.