Rational Handling of Multiple Goals for Mobile Robots

Richard Goodwin **Reid Simmons**

November 1, 1993

CMU-CS-93-204

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Extended version of a paper that appeared in the Proceedings of the First International Conference on Artificial Intelligence Planning Systems, College Park Maryland 1992.

Abstract

The mobile robot planning domain is dynamic, with goals becoming active asynchronously. In order to successfully operate in this environment, a robot must be able to interrupt and reformulate its plan of action on-the-fly. This report investigates a method for incorporating the accomplishment of a new goal into a partially executed plan. A decision theoretic approach using net present value as the decision criterion serves as the basis for determining goal ordering dynamically. The appropriateness of net present value over other criteria is argued. The approach has been implemented on a robot operating in an office setting. Examples from this domain and a planetary exploration domain are used to show the advantages of the approach with respect to fixed priority and heuristic approaches.

This research is supported by NASA under contract NAGW-1175 and in part by a Natural Sciences and Engineering Research Council of Canada 1967 Science and Engineering Scholarship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NASA or the Natural Sciences and Engineering Research Council of Canada.

Keywords: mobile robot, planning, decision theory, utility theory, net present value, benefit-cost ratio

1 Introduction

This report examines a method for handling multiple active goals for mobile robots. Specifically, the focus is on asynchronous goal activation and on how to incorporate the accomplishment of a newly active goal into a partially executed plan. A utility based decision theoretic approach is adopted for investigating the tradeoffs that must be made.

The applicability of decision theory to problems in artificial intelligence and planning in particular has long been recognized. In his review of the use of decision theory, Horvitz argues for its use as a basis for making choices in artificial intelligence [Horvitz *et al.*, 1988]. In an early example, Feldman and Sproull, in their analysis of planning for the hungry monkey problem, used utility based decision theory to evaluate plans taking into account uncertainty and risk [Feldman and Sproull, 1977]. More recent work has focused on time dependent planning [Boddy and Dean, 1989] and on applying decision theory to search [Russell and Wefald, 1991]. The work presented here differs from these in that it focuses on plan evaluation when all goals are not initially known and the plan must be reformulated as goals become active. In such cases, the time dependent utility of goal satisfaction, as well as the time distribution of utilities and resource use, must be taken into account.

Given a utility based framework, one must choose an appropriate decision criterion. In this report a number of such criteria are analyzed: net value, benefit-cost ratio, net present value and cutoff period. Net present value is shown to have some advantages over other criteria when dealing with non-independent goals with discrete resource requirements and time dependent utilities. Plan evaluation based on net present value has been incorporated into a planning system that can interrupt an executing plan and dynamically order goals. The planning system has been applied to two mobile robot domains. The Ambler [Simmons and Krotkov, 1991], a prototype planetary exploration robot designed to carry out scientific missions, has been used as a model for a number of simulations. A Hero 2000 robot, used to perform a number of tasks in our lab, has been used as a vehicle for implementing the ideas. This report examines a number of examples from both domains to show the advantage of using a decision theoretic approach over heuristic based methods and fixed priority schemes. In particular, decision theoretic approaches can lead to more effective usage of the robot's resources including computational resources.

2 Utility Based Rationality

Modern decision theory is concerned with making rational choices among alternatives [Raiffa, 1968]. Rational is taken to mean choosing the course of action that maximizes the expected value of some desired quantity, such as utility. Decision theory provides mechanisms for dealing with uncertainty and the cost of acquiring information. For this reason, it is being increasingly used for planning in real-world domains. Recent examples of the approach can be found in [Wellman, 1988] and [Chrisman and Simmons, 1991].

There are two requirements for formulating a planning problem in terms of decision theory. A method is needed to assign benefits or utilities to the accomplishment of each goal and costs or negative utilities to the consumption of each resource. Secondly, a decision criterion is needed to assess the relative merit of alternative plans.

The assignment of utilities is highly dependent on the set of tasks being considered and the desired behaviour. The exact magnitude of the utility values assigned is not as significant as the relative magnitudes which should reflect the relative priority of the goals.

A number of possible decision criteria have been suggested in the literature [Sassone and Schaffer, 1979]. Much of the body of work done on the development and analysis of the different criteria has focused on its applicability to economic domains [Simon, 1982]. The insights resulting from this work can be adapted to the mobile robot domain. Four commonly used criteria are examined below with comments on their appropriateness.

2.1 Net Value

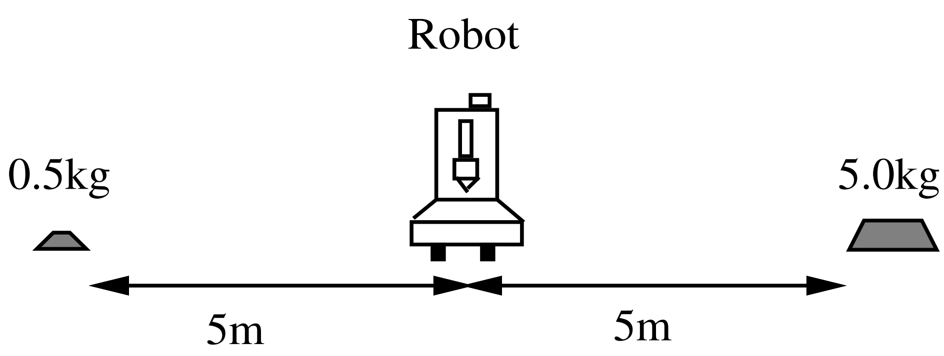
The simplest decision criteria that can be used is net value. The net value of a plan is the sum of the expected benefits minus the sum of the expected costs. The alternative with the highest net value is preferred.

$$NetValue = \sum_{t=0}^{n} B_t - \sum_{t=0}^{n} C_t \ or \ \int_0^n B_t dt - \int_0^n C_t dt = \int_0^n U_t dt \tag{1}$$

B_t : the benefit incurred at time t.
C_t : the cost incurred at time t.
U_t : $B_t - C_t$
n : the life of the project, or time of the last cost or benefit.

The net value criteria suffers from two crucial short comings. It does not take into account the resources needed to generate a given net return. It also does not distinguish between alternatives that have different time distributions for incurring costs and receiving benefits.

The net value method does not exhibit a preference for options that conserve resources. No distinction is made between two options that have the same net return but incur different costs. Conserving resources is desirable to the extent that it allows the unused portion to be used for other purposes. Consuming a resource involves an opportunity cost corresponding to the gains that could be had by investing the resource in other ways. This opportunity cost must be taken into account when evaluating alternatives. One method that has been used is to select the option with the lowest cost when more than one option has the highest net value. This solution only partially accounts for the opportunity cost and is rarely applicable since it is unlikely that the net values will match exactly.



- **Plan 1 :** Go right, pick up gold, go left, pick up gold.
- **Plan 2 :** Go left, pick up gold, go right, pick up gold.

Figure 1: **The greedy plan, plan 1, gets more gold sooner.**

The time distribution of costs and benefits is crucial when selecting among alternatives. It is desirable to have a greedy bias for acquiring benefits sooner and delaying consumption of resources. Consider the example shown in figure 1 where a robot is collecting gold, where gold has a fixed utility per pound collected. A 5.0 kg block of gold is 5 meters to the left of the robot and a 0.5 kg block is 5 meters to the right of the robot. There are two alternative plans for collecting both blocks depending on which block is collected first. Both plans have the same net utility, 5.5 kg of gold, but one gets more gold sooner. The greedy plan should be preferred. The extra benefit of collecting the larger block first can be used for a longer period of time. The robot gets the use of 4.5 more kg of gold for some period of time. Risk is also reduced by accomplishing the higher utility goal sooner since it reduces the probability that the world will change before the higher utility goal is achieved. Similarly, delaying resource expenditures allows the resource to be held longer and possibly used for some option that was not originally available. The net value criteria makes no distinction between the two plans for collecting the gold.

2.2 Benefit-Cost Ratio

The benefit-cost ratio is the sum of the benefits divided by the sum of the costs.

$$\frac{B}{C} = \frac{\sum_{t=0}^{n} B_t}{\sum_{t=0}^{n} C_t} \; or \; \frac{\int_0^n B_t dt}{\int_0^n C_t dt} \tag{2}$$

Taking the ratio of benefits and costs gives a measure of the rate of return. Alternatives that incur less cost to produce the same net benefit will be preferred. In economics, investments are selected by ranking the investment opportunities in order of decreasing benefit-cost ratio and accepting investment opportunities until the available resources are exhausted or the rate of return falls below the cost of capital. This greedy optimization algorithm allows the opportunities to be considered independently and leads to a very efficient decision procedure that is linear in the number of opportunities. Etzioni, in the design of an autonomous agent, uses the algorithm as a basis for the agent's decision control loop[Etzioni, 1989].

As Etzioni points out, there are problems when the opportunities require a discrete amount of each resource and resources are limited [Etzioni, 1989]. In such cases, the problem can be shown to be intractable by a reduction from the knapsack problem [Garey and Johnson, 1979]. In practice, use of the greedy algorithm does lead to problems. Imagine a situation in which an exploration robot has located two adjacent items of interest. One item has a higher value than the other, but also consumes proportionately more resources to extract. Further suppose that there were not enough resources to take both samples — exactly one must be chosen. In this case, the greedy algorithm would choose the suboptimal plan that gives the higher rate of return, but a lower net return.

There is a modified version of the greedy algorithm in which the greedy solution is compared to a solution consisting solely of the item with the maximum net return, and the better of the two solutions selected. This modified algorithm can be shown to be within a factor of two of optimal [Garey and Johnson, 1979]. In the sample selection example above, the modified greedy algorithm correctly chooses the option with the highest net return. Suppose, however, that the situation was changed so that there were enough resources to sample both but that the item with the lower rate of return was degrading over time. The modified greedy algorithm would not be able to generate the optimal plan to first sample the low rate of return item and then to sample the high rate of return item. The algorithm fails to find the optimal solution because the opportunities are not independent. In general, considering opportunities in isolation is insufficient and combinations must be evaluated when selecting a plan.

Another difficultly with using the benefit-cost ratio is that it is dependent on the exact definition of costs and benefits. Suppose there are two methods a robot can use to traverse a room: one that is fast, uses little energy, but is noisy, and a second method that is quiet, but takes longer and uses more energy. Should the negative utility of disturbing others in the room with the noisy traversal be counted as a cost or as a negative benefit? Clearly, the way in which such external effects are treated will affect the ratio, and hence the robot's decisions.

2.3 Net Present Value

One method for taking the time distribution of utilities into account is to use present values. The present value of a cost or benefit is the actual value to be received in the future discounted by a fixed discount rate (d). Using the present value of the costs and benefits takes into account their time distribution. Discounting future utilities creates a preference for benefits that accrue sooner and costs that occur further in the future. For example, given the choice between two plans that achieve the same benefit for the same initial cost, the one that returns the benefit sooner is preferred.

The net present value of a sequence of costs and benefits is the net of the present value of each negative utility/cost or positive utility/benefit. This is the most widely used metric in cost-benefit analysis and is generally considered superior to other metrics [Sassone and Schaffer, 1979].

$$NPV = \sum_{t=0}^{n} \frac{U_t}{(1+d)^t} \; or \; \int_0^n \frac{U_t}{(1+d)^t} dt \tag{3}$$

Resource investments are chosen by generating all feasible combinations of opportunities and selecting the one with the highest net present value. Even in the case where the investment options are not independent (as in the second version of the sampling example above) this method prefers the option that maximizes the net present value of the return.

Net present value treats negative benefits and costs equivalently. There is no need to make arbitrary distinctions. Summing the costs and benefits does mean, however, that they must be normalized to the same scale. In economics this is done by expressing quantities in equivalent dollar values. For the robot domain, quantities can be normalized to their equivalent value in terms of a specific resource or benefit such as time, battery charge or samples taken. In order to create a preference for conserving resources. the opportunity costs associated with consuming the resource must be taken into account.

Adopting the use of net present value results in making the correct choices. It does however lead directly to the intractable problem of having to generate and evaluate a combinatorial number of alternatives. Some method must be used to reduce the number of alternatives that have to be considered. The approach taken in this work has been to generate only a subset of the possible combinations and to do this incrementally as new opportunities become available. Details of the method used are given in the following section on the planning framework.

2.4 Cutoff Period

Another decision criteria that is appropriate in some situations is the cutoff period method. With this method, a specific length of time is chosen and the alternative with the best net return up until the cutoff time is selected. In economics, this method is generally only used to evaluate risky ventures, such as start up companies. In the agent domain, it would be applicable if the domain imposes a limited window of opportunity in which the agent can act. For example, if a robot has only two hours before its battery will run out, it would be appropriate to choose the plan that would produce the highest net utility in two hours. This method can be used when there are multiple windows of opportunity. For example, if a robot must recharge for an hour every two hours, each period of activity could be treated as a cutoff period. However, using the cutoff period method in such a situation would preclude the consideration of plans that require multiple time windows to complete. We mention the cutoff period method only for completeness.

2.5 Discount Rate

The net present value method requires a discount rate. While discounting future values accounts for the time preferences of costs and benefits, the choice of a discount rate is highly problematic. The discount rate reflects a willingness to trade present benefits for future costs. A low discount rate results in decisions that focus on long term impacts; a high discount rate results in greedy decisions. The discount rate incorporates assumptions about risk aversion and the predictability of the environment. For example, using a higher discount rate decreases risk by reducing dependence on the accuracy of predictions about the future since plans with more current benefits are preferred.

The discount rate we have chosen for planning is based on an estimate of the effective planning window for the robot. The effective planning window is the duration of time for which it is useful to make plans. Making plans for events beyond this window of time is of little utility since there is a high likelihood that the situation will change and the plans will no longer be applicable. The discount rate is set so that utilities at the end of the effective planning window are discounted by one half. This rate is currently fixed. If the robot was learning a model of the environment, it would be desirable to adjust the discount rate as the model was refined and confidence in its accuracy increased.

3 Planning Framework

We have developed a planning framework that is geared toward handling asynchronous activation of goals involving robot motion and manipulation. A set of abstract actions is used to construct linear, conditional plans which are refined for execution by means of hierarchical decomposition of the abstract actions. Associated with each abstract action is the information needed to determine if and how the action can be interrupted. When a new goal becomes active, the plan generator creates a set of plans by merging the plan of achieving the new goal with the existing plan. The plan with the highest expected net present value is selected for continued execution.

3.1 Plan Representation

A conditional plan to achieve a set of goals is represented as a tree of abstract actions. Figure 2 shows a simplified version of the plan for putting a cup in the bin. The plan consists of two abstract actions: one to determine if the object is in fact a cup and the second to put it in the bin if it is. There is a branch in the plan for each possible outcome of the abstract actions and associated with each branch is the a priori probability of the corresponding outcome. These probabilities are used to weight the value of each branch when calculating the expected net present value of a plan.

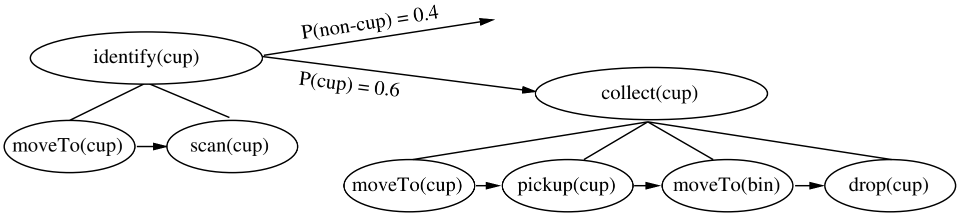


Figure 2: **Cup Collection Plan.**

Executing actions requires use of the robot's resources, such as wheels and grippers. As in the O-PLAN plan representation [Currie and Tate, 1985], each abstract operation specifies the resources that it requires. Resource information allows the planner to efficiently interrupt an action. For example, if the *collect(cup)* action in Figure 2 were interrupted to handle a recharge goal, the robot would not need to put the cup down since recharging does not require the gripper resource. Thus, after recharging, the robot can continue to collect cup plan without having to pick up the cup again which would be quite time consuming. Abstract actions may need to include "phantom" sub-actions [Sacerdoti, 1977] to establish any required state since conditions may be clobbered between abstract actions. In the cup collection plan of Figure 2, both abstract actions include the *moveTo(cup)* sub-action. The second *moveTo* is a "phantom" action since it will not actually be executed if the plan is not interrupted.

The execution of each abstract action consists of the sequential execution of a number of sub-actions. For the purpose of interrupting actions, each sub-action is characterized as either uninterruptible, restartable or resumable. An uninterruptible sub-action cannot be stopped once it has begun execution. For the hero robot, paper delivery is uninterruptible because if the robot ever put the paper it was carrying down, it would never be able to pick it up again. A restartable sub-action can be interrupted, but the entire sub-action must be repeated when execution is resumed. Any initial effort expended is lost. Scanning the cup can be interrupted, but a partial scan provides no information. The entire scan must be repeated when the action is resumed. A resumable sub-action is one that can be interrupted and only the undone portion of the sub-action needs be completed when execution on it is resumed. For an exploration robot, mapping the geology in an area is a resumable task. If the task is interrupted and later resumed, the robot only has to complete the undone portion of the mapping.

Sub-actions that can be interrupted require additional information to be able free and re-acquire resources. For example, if the robot interrupts the *moveTo(bin)* sub-action of the *collect(cup)* action in figure 2 in order to deliver printer output, its gripper must first be freed by putting down the cup before it can pick up the printer output. The cup must be then re-acquired before the *moveTo(bin)* action can be resumed.

3.2 Plan Generation

Generating a plan for a set of goals is, in general, an intractable problem [Chapman, 1987]. The use of the linearity assumption that goals can be satisfied one at a time enables the planner to decompose the problem and generate a plan efficiently. Even with this linearity assumption, there is still a combinatorial number of possible goal orderings that could be considered when trying to optimize the plan. In order to avoid considering all possible orderings, our plan generator creates only a subset of the alternatives that is linear in the size of the original plan. The ordering of actions is the original plan is maintained. New plans are created by inserting the actions for the new goal into the existing plan. If the current action can be interrupted, one of the new plans will interrupt the action and attempt the new goal immediately. Other plans are generated by inserting the actions for the new goal after each of the actions in the existing plan.

The decision not to consider goal reordering or interleaving is based on the assumption of a benign world and a near optimal original ordering for the actions. It is similar to the strategy used in intention-based planning where the planner makes a commitment to its existing plan and filters out options that are inconsistent with this commitment [Bratman *et al.*, 1988]. Unlike Bratman et al.'s IRMA architecture, our current planner does not have a mechanism to override its commitment to its current plan. Whether limiting the planner to examining only a subset of possible goal orderings is rational depends on whether the opportunity cost of not considering other possible orderings is offset by the savings in computation time [Doyle, 1988].

The plan generator can also include domain-specific methods for generating plans. For the Hero domain, a method was added for inserting a new goal when the currently executing action involves carrying an object from one location to another. An *on-the-way* plan is created in which the robot immediately starts achieving the new goal, but drops any objects it is carrying at the point on the new path that is closest to its intended destination.

4 Hero Robot Domain

The Hero 2000 robot operates in an office setting performing a number of tasks [Simmons *et al.*, 1990]. These tasks include delivering printer output, taking objects from one workstation to another, and finding cups on the floor and putting them in a bin. The robot must also maintain its battery charge in order to be able to perform these tasks. The robot has a single manipulator and can carry only one object at a time.

Plan generation and selection using a net present value decision criterion has been incorporated into the software used to control the Hero 2000 robot. The Task Control Architecture (TCA) [Simmons *et al.*, 1990], an operating system for robots, is used as a basis for the implementation. TCA provides mechanisms to schedule and control multiple goals, execute plans and monitor the environment.

Direct experimentation with the Hero robot is time consuming. In order to investigate a larger variety of examples and a larger range of parameter values, a system for simulating the robot, using the planning framework described above, was created with the Maple symbolic math system [Char, 1987]. The simulation software is domain independent. It is targeted to a particular domain by specifying action models, expected time and outcome probabilities, as well as the utility of accomplishing each goal.

Primitive Action Times		
Action	Time (sec)	Description
identifyCup	20	Scan and classify a potential cup.
grabCup	10	Grab a cup with the gripper.
putCupInBin	10	Drop the cup in the bin.
grabPaper	15	Grab paper from the printer.
deliverPaper	20	Give paper to the person.
getObject	10	Get an object from a person.
deliverObject	20	Deliver object to a person.
ungrabObject	5	Drop an object on the floor.

Figure 3: **Hero Expected Action Times**, (seconds) for primitive actions.

The characteristics of the Hero domain were determined empirically (Figure 3 and 4). Euclidean distance and average speed are used to estimate travel times. A discount rate of 0.2% per second was chosen which results in discounting utilities six minutes in the future by 1/2. The six minute time frame is sufficient for the robot to complete one or two tasks, reflecting the robots effective planning horizon.

Locomotion Time

$$MoveTime(a,b) = \frac{distance(a,b)}{HeroSpeed} + stanceTime(stance(b))$$

$$stanceTime(standing) = 0$$
$$stanceTime(centered) = 20$$

Figure 4: **Hero Expected Travel Time** (seconds and feet). The robot must be centered on an object in order to scan it or pick it up. Other actions can be performed in the standing stance.

The utility of having the robot accomplish one of its goals depends on its value to the people in the office in terms of the amount of time it saves them. Time saved, or not saved, was used as a basis for normalizing costs and benefits. The utility values used are the normalized sums of the costs and benefits for satisfying each goal. The time dependent nature of the goals was also taken into account. Delivering printer output and carrying objects from one workstation to another must be done in a timely fashion since people are waiting. The utility of both these activities is represented by a function that is initially almost flat, but decrease to near zero after a delay. The height of the function represents the intrinsic value of accomplishing the goal and the cut off represents the acceptable delay (Eqs 4 and 5). Cup collection is of general benefit, but since no one is waiting for it, it is time insensitive and of relatively low importance (Eq 6). Charging after a low battery indication is not directly beneficial to anyone, but is necessary for the robot to operate. If the robot delays recharging too long and runs out of charge, someone will be required to intervene. For this reason, recharging is characterized by a function containing a negative exponential component, making the utility prohibitively negative after an initial delay. This delay represents the time before the robot would start to lose power (Eq 7).

$$Utility(printer, delay) = \frac{100}{1 + e^{\frac{delay-200}{10}}} \tag{4}$$

$$Utility(delivery, delay) = \frac{200}{1 + e^{\frac{delay-200}{10}}} \tag{5}$$

$$Utility(collectCup, delay) = 10 \tag{6}$$

$$Utility(recharge, delay) = -e^{delay-200} \tag{7}$$

Figure 5: **Hero utility functions**, (delay in seconds)

The following three examples from the Hero domain illustrate the usefulness of the methods described. In particular, we demonstrate the method's superiority over fixed priority and heuristic approaches.

4.1 Cup Collection Example

Suppose the robot is attending to a low utility goal when a new high utility request is received. The robot must decide whether to continue with the current plan or to suspend it until the new high utility goal is accomplished. In the example, illustrated in figure 6, the robot is executing a plan to collect a cup when it receives a request to deliver printer output. Objects are placed in the room in such a way that the cup and the bin are only short detours on the way to the printer from the initial robot location.

Figure 7 shows how the preferred plan varies as a function of the new goal's activation time. This graph was produced by running the simulation with different activation times for the new printer delivery goal. The graph shows that before the robot has picked up the cup, it will suspend cup collection in favour of the printer request. Once the cup is picked up, it will be dropped off on the way to the printer, unless the robot is sufficiently close to the bin to make putting the cup in the bin worthwhile. The distance at which it becomes worthwhile to complete the cup collection first is affected by the relative utilities of cup collection and printer output delivery, by the discount rate, by the cost of re-acquiring the cup, and by the relative positions of the robot, the bin and the printer. Suspending the cup collection task incurs an extra cost to retrieve the cup since the robot must put it down in order to deliver the output. The relative positions are significant since moving toward the bin may move the robot towards or away from the printer.

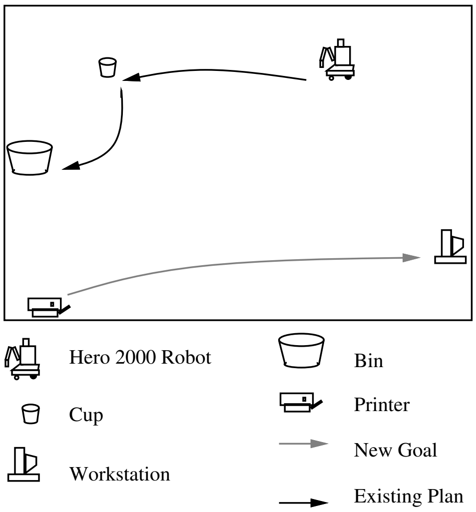


Figure 6: **Cup Collection Example.**

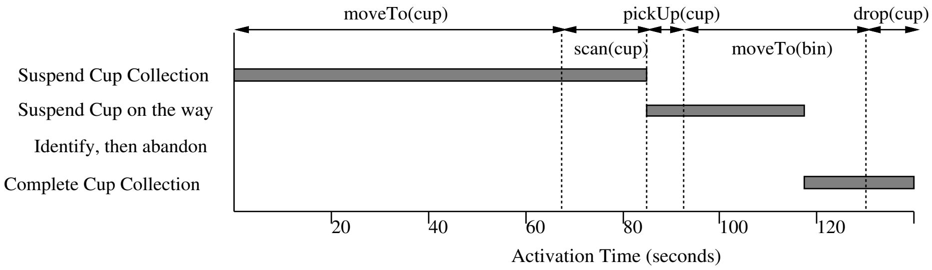


Figure 7: **Best Goal Ordering versus Activation Time. (Cup Collection)**

The time line in Figure 8 describes the "Suspend Cup Collection" plan. The time delay for each goal is the expected time interval between the activation time and the expected time of accomplishment. The utility of accomplishing each goal is calculated by substituting the goal delay time into the utility equation (Eqs 4 – 7). The discount interval is the time interval from the current time to the expected time of accomplishment.

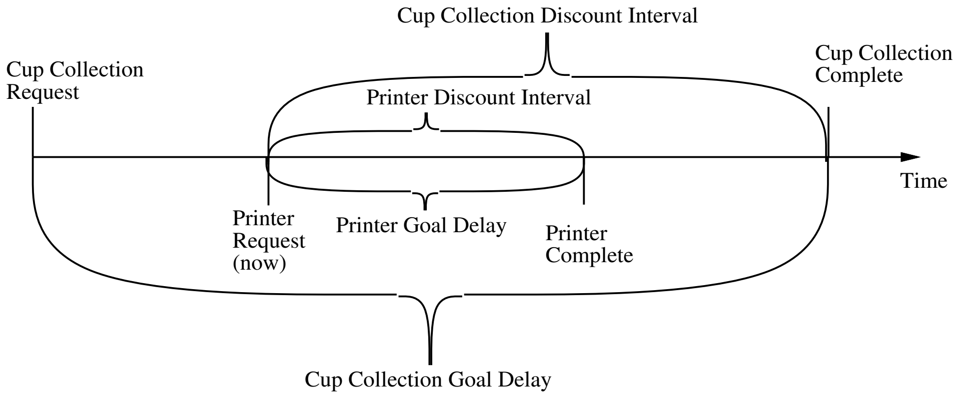


Figure 8: **Time Line for Suspending Cup Collection.**

The utility values are discounted by the discount interval using formula (3) to determine the net present value of each plan. The discount interval and the time delay will be different if some time has elapsed since the activation time, as in Figure 8.

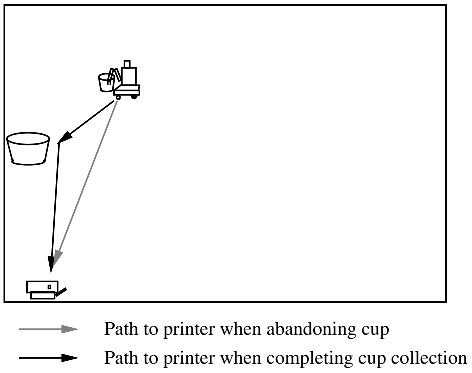


Figure 9: **Printer Path Difference.**

The tradeoff that is being made can be seen more clearly by considering the differences in the paths the robot would take to get to the printer. Figure 9 shows the two paths: one that goes directly to the printer and one that goes by way of the bin. The direct path will always be shorter, but as the robot approaches the bin, the difference becomes arbitrarily small. The corresponding delay incurred by deferring the printer goal approaches zero, as does the corresponding cost. At some point, completing the cup collection first becomes the preferred plan. In Figure 7, this point occurs when the robot is about 4.5 feet from the bin.

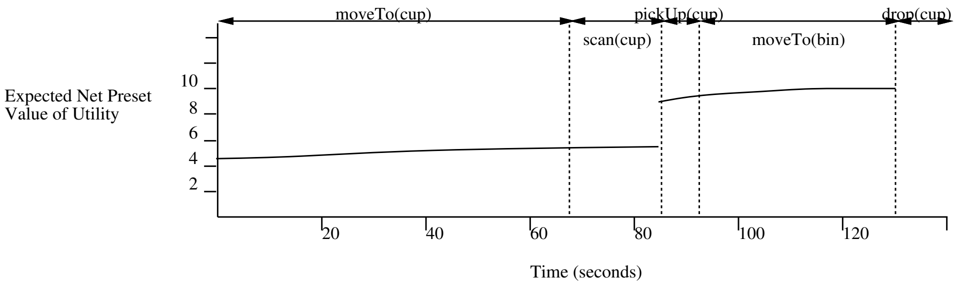


Figure 10: **Cup Collection Utility Versus Time.**

It is interesting to see how sensing operations affect the expected utility of various plans by changing the expected outcome probabilities. Figure 10 shows how the expected net present value of completing the cup collection varies with time. The smoothly rising curve is due to discounting of future values. The step is due to the result of the *scan(cup)* action. Once the object is determined to be a cup, the expected utility no longer has to be reduced by the probability that the object is not a cup. Such steps in the utility function are characteristic of the point in time when a particular branch in a conditional plan is taken.

The example illustrates the advantage of this method over fixed priority schemes. A fixed priority scheme, as was used in the original Hero system, could create situations where the robot would drop the cup beside the bin rather than expend the extra few seconds needed to drop it in the bin. This would get the printer output delivered a few seconds earlier, but it requires the robot to expend a significantly greater amount of time to return, re-acquire the cup, and finish the task.

This example also serves to show some of the limitations of heuristic-based approaches. As stated above, the distance at which the cup collection should be completed depends on a number of factors. For a heuristic method to take these factors into account would require a large number of very specialized heuristics [Feldman and Sproull, 1977]. The utility based method is more general. It depends only on having access to a utility function and on a method of predicting how long a sequence of action will take.

4.2 Delivery Example

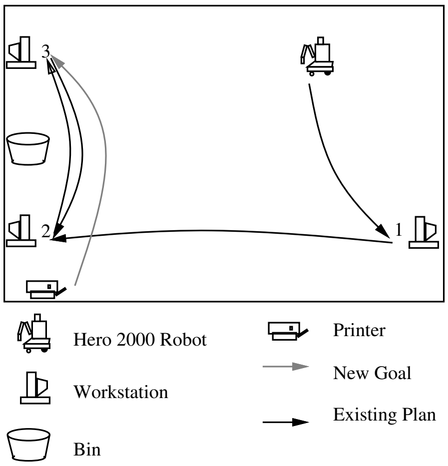


Figure 11: **Delivery Example.**

By properly ordering the achievement of goals, the robot can take advantage of synergistic opportunities derived from the spatial relationships between goals. This result follows naturally from the utility-based approach. Consider the situation shown in Figure 11: The robot is making a sequence of deliveries, one from workstation 1 to workstation 2 and a second from workstation 3 to workstation 2. If a printer request arrives for workstation 3 then the robot can reduce its amount of travel by picking up the output on the way to workstation 3.

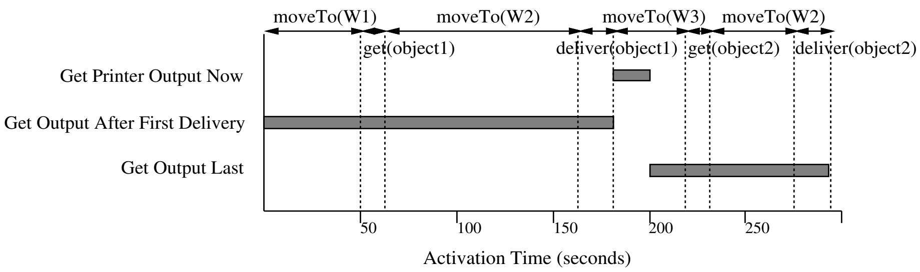


Figure 12: **Best Goal Ordering versus Activation Time. (Delivery Example)**

Figure 12 shows how the preferred strategy changes with the printer request time. Inserting the printer output request between the two deliveries reduces the total amount of travel needed. The robot takes advantage of the fact that the printer output goes to workstation 3, the location where the second delivery begins. Note that the robot will initially go back to the printer even after starting toward workstation 3 to do the second delivery. It is advantageous to do so as long as the robot has not moved too far away from the printer.

A priority based scheme, if used in this example, would not be able to take advantage of the spatial relationships between the goals. The printer request would always be serviced last since it has the lowest utility. A heuristic-based approach could be used to suggest ordering goals such that the destination of one was the start of the next. However, this would not take into account situations as in the example where workstation 2 is only near the printer and not at the same location. In any event, it would be difficult to encode the spatial information that would determine when it is advantageous to return to the printer and when to continue on to the workstation.

4.3 Contingency Example

In the course of working with the Hero Robot, an informal experiment was run to see how people handle the same tasks as the robot. It was observed that sometimes people would elect to "recharge" rather than go collect a cup, even though they had sufficient "battery charge". Invariably, the reason given was that they wanted to have enough charge in reserve to be able to handle a possible printer or delivery request.

The techniques described in this report can be applied to model this type of contingency planning. The situation in the experiment was modeled as a choice between two plans: $plan_1$, to collect the cup first, and $plan_2$, to recharge first. These plans were evaluated taking the possibility of a printer request into account. It was assumed that there was a constant probability $P(printer)$ of a printer request arriving in any minute, (the possibility that two or more requests would arrive was ignored). Let $Plan_1'(t)$ be the net present value at time t of the plan that would be selected if $plan_1$ were used and a printer request arrived at time t. Multiplying $Plan_1'(t)$ by the probability that a request will arrive at time t and discounting it back to time zero gives the current net present value weighted by its probability. Integrating gives the total net present value (equation 8). A similar calculation gives the result for $plan_2$.

Using the utility values selected for the domain and a 20% probability of a printer request arriving from workstation 3 in any minute results in a preference for the plan that recharges. If the possibility of a printer request is not taken into account, the plan to collect the cup is preferred. Obtaining this result using full numerical integration in Maple is computationally expensive requiring a few minutes of elapsed time on a SPARC II workstation. The example does serve to suggest, however, that the approach may be applicable using a more efficient implementation and further approximations.

$$NPV_1 = \int_0^T \frac{P(printer) * Plan_1'(t)}{(1+d)^t} dt + (1 - (P(printer) * T)) * NPV(Plan_1) \tag{8}$$

5 Ambler Domain

The Ambler is a six-legged prototype planetary exploration rover [Simmons and Krotkov, 1991]. Its proposed tasks include investigating sites of potential interest, taking samples and building terrain maps. Sites of interest will be identified from existing satellite images or from the images sent back to earth by the robot. The robot moves very slowly, on the order of half a meter a minute, and the distances between sites can be relatively large, so travel time dominates estimated plan execution times.

The utility of completing one of the Ambler tasks is essentially time independent. The value of investigating a particular site or taking a particular sample does not vary with time. For this reason, the Ambler utility functions are simple constants. Figure 13 gives the values used for the simulation.

The estimated action times for the Ambler are shown in Figure 14. As with the Hero robot, Euclidean distance and average speed are used to estimate travel times. A discount rate of 0.2% per minute was chosen. This rate discounts values six hours in the future by 1/2, which is a suitable planning window for the Ambler.

$$Utility(investigate, delay) = 1 \quad (9)$$
$$Utility(sample, delay) = 10 \quad (10)$$

Figure 13: **Ambler Activity Utilities.**

Primitive Action Times		
Action	Time (min)	Description
Investigate	10	Investigate a site, decide whether to sample or not.
CollectSample	100	Collect and store a sample.

Locomotion Time
$MoveTime(a, b) = \frac{distance(a,b)}{AmblerSpeed}$
$AmblerSpeed = 0.5$

Figure 14: **Ambler Expected Action Times**, (minutes and meters/minute).

5.1 Exploration Example

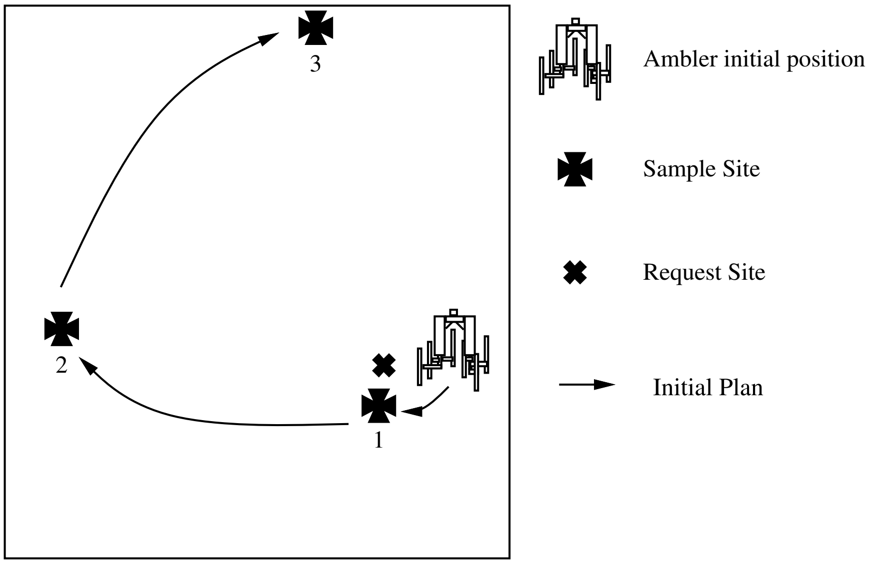


Figure 15: **Ambler Example 1 : Sample Collection.**

Simulations using the Ambler domain were used to investigate the affect of information gathering. As the rover moves around, it gathers more information about the local environment. This information would be transmitted back to earth where specialist will use it to identify new sites of interest and update the probabilities that chosen sites are likely to prove interesting. Transmitting and analyzing the data would take considerable time, and the robot may have moved significantly in the duration. This example explores the tradeoffs in goal ordering as a function of this delay.

The sample collection example (Figure 15) consists of an initial plan of three site investigations. Each investigation consists of moving to the chosen site and examining it for interesting rock formations. Each site has a 60% probability of being found interesting, in which case a sample is collected. Suppose that from the information gathered while moving to the first site, the geology specialist on earth identify a new site near the first site. Since this site has been seen to some extent, it is given an 80% chance of proving interesting enough for a sample to be taken.

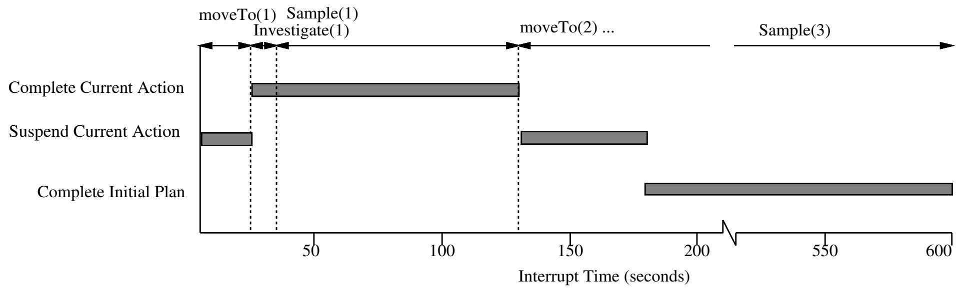


Figure 16: **Best goal ordering versus Activation Time.**

The preferred plan as a function of time the robot receives the new goal is given in Figure 16. The results of the evaluation can be understood as follows: If the new request arrives before the Ambler reaches the first site, it will service the new request first, since it has a higher likelihood of proving interesting. Once sampling at the first site has begun, it will be completed before investigating the new request. After the robot has completed sampling at the first site it will visit the new site before visiting sites 2 and 3, as long as the robot has not moved too far away from site 1. If the Ambler has moved far enough away, it will delay servicing the new request until it has completed the other investigations.

This example serves to show how analysis of new information can be used to activate new goals which can then be incorporated into the executing plan.

6 Limitations

The plan representation chosen imposes a number of limitations on the types of plans that can be expressed. For one, there is no way to express partially ordered plans: The representation requires a linear ordering of abstract actions and primitive actions. Also, there is no way of expressing concurrent execution of actions.

The limitation imposed on concurrent action execution could be removed since the system need only determine the expected time to complete a given sequence of actions. This is currently done by summing the expected time for each primitive action. If concurrent execution were allowed, the expected time calculations would have to take this into account.

The current system never considers reordering the actions in the original plan. In some circumstances, this leads to the adoption of a plan that is significantly sub-optimal. What is needed is some type of over-ride mechanism as used in the IRMA architecture[Bratman *et al.*, 1988]. One possible approach would be to find the best place to insert the new goal and then consider reordering the goals scheduled to be achieved after the new goal. The rational behind this would be that the plan up until the new goal remains unchanged, and hopefully nearly optimal. This is not true for the remainder of the plan. The initial conditions for the portion of the plan after the new goal could have changed significantly, providing an opportunity for further optimization.

7 Future Work

There are a number of open questions that remain to be addressed. These include how to select utility functions, how to select an appropriate discount rate, and how to deal with uncertainty in action time estimates.

For the example domains used in this report, the form and parameters of the utility functions were first formulated with the hope that they would produce the desired behaviour. Experimentation allowed the parameters to be tuned. Further work remains to be done on how to map desired behaviour to specific utility functions. To do this involves understanding how the set of utility functions interact to influence overall behaviour. This is important both for selecting the form of the utility functions and for adjusting the parameters.

The method of plan selection used does not take into account any measure of the confidence in the accuracy of the action times and utility estimates. This could be especially important in circumstances where confidence levels vary significantly. Less credence should be given to plans whose utility is sensitive to small changes in parameter values for which there is little confidence.

Decision criteria such as net present value that depend on a discount rate are highly sensitive to that rate. Choosing a discount rate is still a matter of experimentation. Further work is needed to determine the characteristics of the domain that should be taken into account when selecting a discount rate. One possibility is to have the robot adjust its discount rate as it refines its time estimates and its estimates about the probability of future events.

8 Conclusions

This report has presented some initial results on rational planning for mobile robots. The examples presented show that a mobile robot can take advantage of opportunities as they arise if it can interrupt and reformulate its plan of action. A decision theoretic approach to plan reformulation is more general than heuristic based methods and produces more rational results than do fixed priority schemes. The use of a net present value decision criterion for the mobile robot domain has some advantages over benefit-cost ratio and net value criterion when dealing with limited resources and non-independent alternatives.

A decision theoretic approach to plan evaluation is useful when dynamically reordering multiple active goals. Coupled with the use of net present value and consideration of opportunity costs, it provides the basis for effective operation of a mobile robot.

Acknowledgments

We wish to express our thanks to Long-Ji Lin and Lonnie Chrisman for their work on the initial design and implementation of the Hero robot system.

References

[Boddy and Dean, 1989] Mark Boddy and Thomas Dean. Solving time-dependent planning problems. In *IJCAI-89 Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. AAAI, August 1989.

[Bratman *et al.*, 1988] M. E. Bratman, D. J. Isreal, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4), 1988.

[Chapman, 1987] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32, 1987.

[Char, 1987] Bruce W. Char. A tutorial introduction to maple. Technical Report CS-85-56, University of Waterloo, 1987.

[Chrisman and Simmons, 1991] Lonnie Chrisman and Reid Simmons. Senseful planning: Focusing perceptual attention. In *Proceedings, Ninth National Conference on Artificial Intelligence*. AAAI, July 1991.

[Currie and Tate, 1985] K. Currie and A. Tate. O-plan-control in the open planning architecture. *Expert Systems*, 1985.

[Doyle, 1988] Jon Doyle. Artificial intelligence and rational self-government. Technical Report CMU-CS-88-124, Carnegie Mellon University, March 1988.

[Etzioni, 1989] Oren Etzioni. Tractable decision-analytic control. In *Proceedings, First International Conference on Principles of Knowledge Representation and Reasoning*, May 1989.

[Feldman and Sproull, 1977] Jerome A. Feldman and Robert F. Sproull. Decision theory and artificial intelligence ii: The hungry monkey. *Cognitive Science*, 1, 1977.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[Hansson *et al.*, 1990] Othar Hansson, Andrew Mayer, and Stuart Russell. Decision theoretic planning in bps. In *Proceedings of the AAAI Symposium on Planning*. AAAI, 1990.

[Horvitz *et al.*, 1988] Eric J. Horvitz, John S. Breese, and Max Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2, 1988.

[Raiffa, 1968] Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, Reading Mass., 1968.

[Russell and Wefald, 1991] Stuart Russell and Eric Wefald. *Do the Right Thing*. MIT Press, 1991.

[Sacerdoti, 1977] Earl Sacerdoti. *A Structure for Plans and Behavior*. American Elsevier, New York, NY, 1977.

[Sassone and Schaffer, 1979] Peter G. Sassone and William A. Schaffer. *Cost-Benefit Analysis A Handbook*. Academic Press Inc, 1979.

[Simmons and Krotkov, 1991] Reid Simmons and Eric Krotkov. An integrated walking system for the Ambler planetary rover. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2086–2091, Sacramento, CA, April 1991.

[Simmons *et al.*, 1990] Reid Simmons, Long Ji Lin, and Chris Fedor. Autonomous task control for mobile robots. In *Proc. IEEE Symposium on Intelligent Control*, Philadelphia, PA, September 1990.

[Simon, 1982] Herbert A. Simon. *Theories of Decision-Making in Economics and Behavioral Science*, pages 287–317. The MIT Press, 1982.

[Wellman, 1988] Michael P. Wellman. Formulation of tradeoffs in planning under uncertainty. Technical Report MIT/LCS/TR-427, MIT, August 1988.