ROBOT DECISIONS BASED ON MAXIMIZING UTILITY

by Walter Jacobs and Maxine Kiefer
The American University
Department of Mathematics and Statistics
Washington/ D. c. 20016

ABSTRACT

The decision-making component of a robot that operates in a poorly known environment is considered. The usual problem-solving approach to handling a task is not suitable when each decision may turn out in several ways, and many decisions are needed to complete the task. An alternative approach, which is based on maximizing the estimated utility resulting from each decision, is illustrated. The example describes the plans, utility functions and decision procedures of a simulated insect-like robot called PERCY. In spite of its limited ability to perceive and store information about the environment, PERCY can achieve satisfactory performance on its task.

1. Introduction

Robot experiments are currently carried on in limited task environments. In such circumstances, it is feasible to provide the system with comprehensive, if not perfectly accurate, information covering those features of the environment that relate to its tasks. But to depend on planning methods that require such comprehensive information will hinder the extension of the work that is now being done. Systems will have to be able to operate in environments where much of the knowledge that would help in carrying out their tasks cannot be provided beforehand, and where the approach of storing detailed information as it is received is unsatisfactory, because there is more information than can be stored, or because it changes too often.

Comprehensive models of the environment are needed when planning is based on a problem-solving approach. This approach makes an exploratory analysis of its model, and lays out an intended course of action before the first action is taken. But if it is likely that the course of action will not reach its intended outcome, because the system's knowledge of its environment is incomplete or inaccurate, planning as though the outcome were certain can be a waBte of effort.

The possibility of using multiple-outcome operators in planning has been discussed by Munson[5] and by Fikes, Hart and Nilsson[1]. However, these discussions make it appear that multiple outcomes do not mesh well with a problem-solving approach. Every plan with a desired goal state as one of its possible outcomes may have to assign low probability to that outcome, because the conditions to be met in getting there are not known, and not because it is really improbable that the goal will be reached. It is not obvious how to formulate the decision process when that case is likely to come up.

A different approach to the use of multiple-outcome operators has been described in a series of articles[2,3,4] This type of planning focuses on intermediate objectives, and uses probability estimates to take account of its inability to predict how a course of action will turn out. The approach has been tested in experiments with a simulated robot that has only crude knowledge of its environment.

The design of the robot involves a hierarchy of control, with selection of plans handled at the highest level, and the direct control of perception and action delegated to the lowest level. Various types of planning can function with this hierarchical organization. In the experiments, the selection of plans has been based on a policy of maximizing utility. Thus, the robot's decision-making component has some features of the "cost-effective executive" described by Munson. But there are differences. For one thing, utility is associated not with a specific goal, but with events that occur in the course of the task, and contribute to or detract from the overall measure of the success in performing the task.

The system that is simulated, in the experiments with this approach, is an insect-like robot called PERCY. Its task is to build a nest, and that calls for a number of trips about the environment to locate material and add it to the evolving nest. PERCY can complete the nest in a very large number of ways; its problem is to find a way that keeps it adequately fed, and still does not excessively delay the completion, in the latest experiment, PERCY must also minimize encounters with an enemy — the stinger; the risk of painful stings can be totally eliminated only at the cost of an unacceptable slowing of progress on the nest.

PERCY has very poor knowledge of its environment. Its ability to perceive the objects of importance in its task is limited, and it has no map of the environment to use in locating those objects. It finds its way about by means of landmarks, which it can perceive at a distance. These landmarks help it to locate the nest site, several places where material can be found, and the entrances to the area where its food can be hunted. A diagram of the environment is shown in Figure 1.

In this task, there is no experimenter whose commands set problems for the robot to solve. The situation is more like the playing of a game, in which the course of action is limited by and responsive to the moves of the opponent. And in fact, decisions are treated as the system's moves in a game played against the environment. In this game, the environment replies to a decision by choosing one of its possible outcomes, and that outcome identifies the stage of the task at which the next decision must be made. A game tree summarizes the set of complete sequences of alternating decisions and outcomes; each sequence describes a possible instance of performing the system's task. Each such complete sequence has associated with it a utility value, which measures the quality of performance when the task effort has followed the course described by the sequence. A component of the system produces the system's strategy in the game, and the success of the strategy is measured by the average utility over a series of performances of the task.

The game played against the environment differs in important respects from the usual games played by programs*. For one thing, the environment as opponent is not trying to win. Therefore, it is not appropriate to plan as though the opponent will make the move that is least advantageous to the system. A policy of maximizing expected utility takes the place of minimaxing. Secondly, the payoffs occur during the course of the task rather than at the end. PERCY does not win or lose; its utility rises when things go well, and falls in the contrary case. And finally, it is not necessary for the system to make a new decision each time it must act. Each decision picks an operator that controls a course of action, and a new decision is called for only when the outcome of the previous one is determined.

2. The PERCY Simulations

There are three types of stages at which PERCY must make a decision. The first type occurs when an immediate objective of a preceding decision has been realized: material has been picked up, or placed at the nest, or food has been eaten. In the second type, new information has been obtained: food has been spotted while looking for material, or material has been seen while hunting food, or the Stinger has been sighted. The third type of decision stage is reached when food is being tracked and is lost to sight.

Decisions are again of three types. PERCY can hunt for food. If it iB carrying material, it can head for the nest to place the material. And if it is not carrying material, it can try to locate some.

A decision is fully specified by giving the list of targets that are of interest while the decision is being executed. The list contains a primary target — food, material, or the nest; it usually contains one or more secondary targets. A secondary target may be one of the landmarks used by PERCY in finding its way around the environment. Also, either food or material may be a secondary target when the other is the primary one, and the Stinger will be a secondary target when seeing it will call for a decision whether to avoid it or not.

The outcomes of these decisions have descriptions similar to those of the decision stages that arise when the outcomes are reached. In addition, there are two outcomes that end the task; completion of the nest, or abandonment of the task. Note that in general a decision stage can be named by the outcome that triggers it, and we will make no distinction hereafter between outcomes and decision stages.

Table (1) gives a complete list of outcomes for PERCY'S decisions. Also, for each outcome that does not end the task, the decisions open at that stage are listed.

TABLE (1) PERCY'S Outcome and Decisions

Type of Outcome	Decisions Available
1. Material added to nest, task continues	la. Find material near landmark 1 lb. Find material near landmark 2 lc. Find material near landmark 3, watch for food and Stinger ld. Hunt food, watch for material. Stinger
2. Material taken *near* landmark 1	2a. Add material to nest 2b. Hunt food via landmark 2
3. Material taken near landmark 2	3a. Add material to nest 3b. Hunt food
4. Material taken near landmark 3	4a. Add material to nest 4b. Hunt food
5. Food eaten, material not held	5a. Find material near landmark 2 5b. Find material near landmark 3
6. Food eaten, material held	6a. Add material to nest via landmark 2 6b. Add material to nest via landmark 3
7. Material seen while hunting food	7a. Take material 7b. Hunt food
8. Food seen while finding material	8a. Find material near landmark 3 8b. Hunt food
9. Stinger seen	9a. Find material near landmark 2 9b. Find material near landmark 3 9c. Hunt food via landmark 2 9d. Hunt food via landmark 3
10. Food lost to sight, material not held	10a. Find material near landmark 3 10b. Hunt food via landmark 3
11. Material added to nest, task done	11a. Wait for new task
12. Task abandoned	12a. Wait for new task

An important point is brought out by this table. There are in fact many more decision stages in PERCY'S task then are shown, for each outcome represents a class of <u>equivalent</u> decision stages. Two stages are equivalent when they have identical lists of decisions, and each decision uses the same operator in both stages. This does not mean that the same decision will be made at the two stages, for equivalent stages may differ in certain parameters that enter into the making of decisions.

The use of equivalence among decision stages means that the system is applying generalization or abstraction to its task. The decision stage when material is first added to the nest is evidently different from that where the addition of material leaves the nest nearly completed, but PERCY'S decisions are not affected by the difference. In fact, it cannot recognize the difference.

Abstraction is important because it can greatly simplify the problem of providing a strategy. Clearly, the fewer the distinguishable outcomes, the less information will be needed in making decisions. The importance of this will be evident when the way PERCY makes its decisions is described. Also, abstraction aids learning, as will be seen. *On* the other hand, abstraction may omit information that is relevant to decisions, and so may adversely affect performance of the task.

The way PERCY assigns utility values will now be described. Four kinds of payoffs occur during a trial of its task. Each placement of material at the nest, each feeding, each pickup of material, and each sting, all make their contribution — positive or negative — to the measure of utility for the trial. The utility measure $u_A(t)$ for the placement of material depends on the time t that has elapsed since the previous placement. One may think of this measure as combining two offsetting factors; a fixed increment of satisfaction when the material is placed, and a small fixed charge against satisfaction at every instant of time that placing of material is delayed. Similarly, the utility measure $U_E(t)$ for the consumption of food depends on how long it has been since PERCY ate last. The dependence on the elapsed time is more complicated than in the case of placement, because the satisfaction in eating is small if PERCY is not yet hungry, and the dissatisfaction of not eating increases sharply as PERCY gets hungrier. When the elapsed time is so great that the pleasure of eating is outweighed by the pains of hunger before eating, the net utility u_E is negative. Finally, the positive utility U_M gained when material is picked up and the negative utility U_g that accompanies a sting are constant. The utility functions are summarized in Figure 2.

The measure of utility for a trial of PERCY'S task is the sum of the utilities for the eight pickups of material, for their additions to the nest, for the various feedings, and for the stings received. The problem of achieving good performance is not a trivial one. PERCY will get its best results by properly spacing its feedings, and making quick round trips to the nest whenever the time since it last ate permits this. To complicate matters, when it sees the Stinger it may have to decide whether it is better in the circumstances to accept the sting and save time, or avoid the pain and instead put up with a substantial delay. There are two important parameters that distinguish equivalent decision stages. These are the time since PERCY last ate, and the time since material was last added to the nest. They must be involved in the way decisions are made if good performance is to be attained.

3. <u>Planning Strategies</u>

We consider strategies that make decisions by exploring the game tree. Exploration involves the evaluation of <u>plan</u>s.

A plan P at a decision stage D consists of one decision d available at D, together with some or all of its possible outcomes, and perhaps further decisions and outcomes. In the game against the environment, the plan is analogous to a protocol that begins: "If I make this decision, the following outcomes may occur; if this outcome should happen, I can then make one of these decisions; ..." It may be formally represented as a subtree of the game tree; this subtree is rooted at the node corresponding to D and it contains a single branch from that node — the branch for the decision d. The nodes that terminate the plans examined at a decision stage *D* mark the limits of the planning horizon at that stage. Values or utilities are estimated for the stages of the task that correspond to these terminal nodes, and on the basis of these values, a value is derived for each plan — or equally, for the decision implied by the plan.

When a planning strategy is used, at each decision stage D one or more plans are evaluated, until a decision with a suitable value is obtained. This may be the decision with the highest value, among those available at D; or the first decision with a value exceeding some appropriate threshold; or even the best decision found among the first n evaluated, for some suitable n.

Thus a planning strategy involves four elements. There must be a way of providing a set of plans for each decision stage. A rule for assigning values to the situations that terminate a plan must be present. A way of deriving from these terminal values the value of the decision that initiates the plan is necessary. And finally, there must be a rule for selecting a decision at a decision stage, based on the values that are obtained for the individual decisions.

In a planning strategy that uses the policy of maximizing utility in making its decisions, the last three of these elements take specific forms. A utility estimate is provided for every situation that terminates a plan. Also, for every decision d, a set of estimates of the probabilities $\{q_a(0)\}$ of the possible outcomes 0 must be available. Then, utilities can be obtained by working backwards from the terminal stages of a plan, using the fact that the utility assigned to a set of uncertain outcomes is the expected value of their utilities. That is,

$$U(d)=\sum_{0'} q_d(0')\,u(0')$$

where the sum is over all outcomes 0' of the decision d that have non-zero probabilities. And because of the policy of selecting the decision that has the highest utility,

$$u(0)*\max_{d} U(d)$$

where the maximum is taken over the decisions open at the decision stage that corresponds to outcome 0. Continued application of these two rules, to the utilities of the outcomes that end the plan, will arrive at the utility of the plan and its decision, and will also decide among the available plans.

Both problem-solving strategies and minimax policies can be treated as special cases of this approach, although they are not usually so considered. In problem solving, uncertainty in the environment is usually ignored; this means that the most probable outcome of a decision is estimated to have probability 1, and all others probability 0. Application of a minimax policy is equivalent to estimating probability 1 for the outcome with the lowest utility, and probability 0 for other outcomes, when working back from the situations that terminate a plan.

In a stable environment where the probability distributions $\{q_d(0)\}$ accurately reflect the responses of the environment to the execution of decision d, there exists a set of ideal utilities $\{u(0)\}$ for every outcome that can occur in the course of handling the task. These are derived from the utilities of the payoffs in the following way: Each outcome that terminates the task is assigned a utility equal to the sum of the utilities of the payoffs that occur in the corresponding trial. Then, by repeated application of the two equations given above, utilities for all other outcomes are obtained. In the process, the strategy that yields the highest expected utility against an environment with the specified probabilities is also determined.

In actual systems dealing with tasks involving fairly large game trees, the probability and utility estimates will at best crudely approximate the ideal values just described. In a dynamic environment, a stable set of probability values need not exist. And even when they do, the estimates of the values available to the system may be inaccurate. In addition, the calculation of utility estimates from the payoffs, in the manner just described, may be far beyond the capabilities of the system.

Nevertheless, given such a task, there may be no practical alternative to the policy of maximizing utility as a basis for designing a strategy component for the system that must deal with it. Thus, the experiments with PERCY are useful in showing what can be accomplished with such a strategy, despite the reliance on approximations to probabilities, utilities, and other parameters that enter into the making of decisions.

4. How PERCY Makes Decisions

In the PERCY experiments, the planning horizon extends only as far as the outcomes of each single decision. The use of the most limited possible exploration of the future in decision making is consistent with the system's crude perceptual capabilities and poor knowledge of its environment, its lack of memory of earlier stages of a trial, and its consequent inability to distinguish among equivalent decision stages.

However, when the capacity for cognition that PERCY is given is limited in these ways, it is possible to use a very simple structure and a minimum of data in making its decisions. Each decision becomes a plan merely by appending a list of the possible outcomes. Only one probability distribution {qa(0)> is needed in finding the utility of the plan that involves the decision d. And a uniform method of estimating the utility of any outcome takes care of the remaining requirement for decision making. The utility estimates given by this method do not satisfy the theoretical relationships stated earlier for ideal utilities. But in view of the inaccuracy of PERCY'S information, they may well be little worse than those that would result from more extensive calculations calling for a much more complicated strategy component.

The utility estimate is closely related to the utility function for payoffs described in section 2. Suppose PERCY is at the decision stage following outcome 0, that a time h has elapsed since it last ate, and that a time n has passed since material was last added to the nest. PERCY must estimate a utility ^{u}df°') for outcome 0', when that outcome is reached after making decision d. The estimate is based on the utility functions u_E, u_A u_M, u_s given in Figure 2, and on three time estimates:

- t_d — the estimated time to reach outcome 0' if decision d is executec and 0' occurs
- t_e — the estimated time between occurrence of 0' and the next feeding (0 if eating occurs at 0')
- t_a — the estimated time between occurrence of 0' and the next addition to the nest (0 if placement occurs at 0*)

Then (see Figure 3):

$$u_d(0')=u_E(h+t_d+t_e)+u_A(n+t_d+t_a)+q_s u_s+e u_M$$

where q_g is the estimated probability that PERCY will be stung in reaching outcome 0', also e is 1 if material is picked up at 0', and is 0 otherwise.

This utility estimate is an approximation to the utility of the payoffs that will take place until the next addition to the nest and the next feeding. The approximation is rough, because the values estimated by t_e and t_a depend on h and n and also on subsequent decisions, but t_e and t_a are treated as constants.*

Since PERCY has estimates of t_e and t_a for every outcome 0', and also has estimates of ti, q_s and q_afO') for every decision-outcome pair (d,0'), it can calculate the utility of its decisions d, using the equation

* The method of estimating utilities described in this section is the one used in the most recent experiments with PERCY. It differs in several respects from the method reported in earlier articles.

$0'$

At a decision stage it makes the calculation for each available decision and selects the decision with the highest utility.

Of course, the probabilities and elapsed times are characteristic of the particular environment in which PERCY finds itself. PERCY is not endowed with good estimates of these quantities. Rather, initial estimates are supposed to have been developed as PERCY explored its environment in tasks engaged in earlier; the estimates are improved on the basis of its actual experience in the nest-building and related tasks. That is, PERCY has a capability to <u>learn</u> by developing better values on which to base its decisions.

The estimates for $q_d(0')$, t_d and q_s are provided by means of a set of totals -- one set for each decision stage 0:

N(d,0') -- the number of times decision d has resulted in outcome O"

T(d,0') -- the total time elapsed in reaching outcome 0' on these occasions

S(d,0') — the number of times that a sting was received in reaching 0'

At a point where the estimates are needed, they are calculated by means of the relations

$$t_d = T(d,0')/N(d,0'),$$

$$q_s - s<d,0')/N(d,0') ,$$

$$qd(O') = rN\{d,0')+i]/Z I N(d,0')+I);$$
$$0^*$$

the sum in the last of these expressions is over all possible outcomes of decision d.*

Estimates of t_e and t_a are stored for each outcome. When $t_e(0)$ is not zero — i.e., when eating has not occurred at 0 — its value is revised each time a decision d, made at 0, arrives at an outcome 0'. The new estimate is obtained by adding to the old one the quantity

$$k[t(0')+t_e(0')-t_6(0)],$$

where t(O') is the actual time it took to reach O' after decision d, and k is a small *constant,* say .2, The corresponding adjustment for $t_a\{0)$, when its value is not fixed at zero, is

$$k I t(0')+t_a(0')-t_a(0)]$$

The derivation of these revision formulas is omitted. They are approximations, which are used in order to live with the constraints of limited memory and ability to calculate that are assumed for PERCY.

We have now described how PERCY, in the course of carrying on its task, continually revises its estimates of time and probability.

This process of revision constitutes PERCY'S learning, or adjustment to the environment. This is the only type of learning provided in the simulations thus far, although other types can readily be added to the general structure that underlies PERCY.

5. <u>PERCY'S Performance</u>

The simulation as described has been run with a number of different sets of utility parameters. Various sets of values were chosen for the constants k_i in the utility functions given in Figure 2. The relative weights given to eating regularly, making progress on the nest, and avoiding stings were changed thereby. With each set of values, a series of trials of the nest-building task was run. As soon as a nest was finished, it was wiped out, and a new trial began. The estimates with which PERCY began this new trial were the ones arrived at by the end of the preceding trial, so that there was an opportunity to demonstrate learning during each series. All series started with the same relatively poor estimates.

It was evident at the start of the experiment that PERCY'S efforts were doomed to frustration if its utility functions were incompatible with the realities of the environment. For example, when both k3 and k5 are less than the time required for a round trip to get material and food and bring it back to the nest, a successful performance of the task is impossible. Ideally, the parameters should make more than one satisfactory strategy available, so as to test whether the learning feature in the system helps to offset the limited horizon used in planning.

Tables (2) and (3) give information about the kind of behavior PERCY exhibited in these experiments. They describe the first trial of a series, and the seventh trial of the same series, by which time the behavior had converged to a locally optimal strategy which was quite good, though not the best available. In the initial trial, PERCY often made decisions that exposed it to the risk of a sting, and with rather baa luck Buffered a total of five stings• By the seventh trial, it had substantially reduced the likelihood of a sting. It did this by going the long way round to get material and food, and returning the short way. Whenever it got back sufficiently quickly, it made a rapid round trip to the nearby material location before going after food again.

The convergence after a half dozen or so trials to a rather rigid pattern of decisions occurred with each set of utility parameters used. It appears that adjusting the estimates t_a and t., when this is done as if the values being estimated are constants, cannot fully make up for the limitations PERCY has in its ability to plan. One of the several directions for further work is to see whether these characteristics of its decision making can be improved without an appreciable increase in PERCY'S memory and ability to calculate.

* This is the so-called Bayes estimate for the probability of O'. It gives a non-zero estimate for the probability of events that are possible but not yet observed.

PERCY'S BEHAVIOR

Table (2)

First Trial

Decision Stage	Decision Made	Outcome	Number of Time Units	Utility At Outcome
At nest	M F 3 S	Sees S	51	0
Stinger seen	M 2	Gets M at 2	89	100
Material taken at L2	F	Gets F	57	423
Food eaten, material held	N 2	Places M at N, stung	174	81
At nest	M F 3 S	Sees F	51	81
Food seen	M 3	Gets M at 3	9	181
Material taken at L3	F	Gets F	44	568
Food eaten, material held	N 2	Places M at N	184	392
At nest	M F 3 S	Sees F	58	392
Food seen	F 3	Gets F	37	780
Food eaten, no material	M 3	Gets M at 3	44	880
Material taken at L3	N	Places M at N, stung	69	464
At nest	M 1	Gets M at 1	37	564
Material taken at L1	N	Places M at N	33	824
At nest	M F 3 S	Sees F	58	824
Food seen	F 3	Gets F	31	1202
Food eaten, no material	M 3	Gets M at 3	45	1302
Material taken at L3	N	Places M at N, stung	68	898
At nest	M 1	Gets M at 1	33	998
Material taken at L1	N	Places M at N	36	1260
At nest	M F 3 S	Sees S	52	1260
Stinger seen	M F 3	Gets M at 3, stung	16	960
Material taken at L3	F	Gets F	46	1360
Food eaten, material held	N 2	Places M at N	195	1142
At nest	M F 3 S	Sees S	47	1142
Stinger seen	M F 3	Sees F, stung	10	742
Food seen	F	Gets F	36	1138
Food eaten, no material	M 3	Gets M at 3	43	1238
Material taken at L3	N	Finishes N, stung	66	828

****

Table (3)

Seventh Trial

At nest	M 1	Gets M at 1	35	100
Material taken at L1	N	Places M at N	35	360
At nest	M 2	Gets M at 2	129	460
Material taken at L2	F	Gets F	54	753
Food eaten, material held	N 3	Places M at N	114	559
At nest	M 2	Gets M at 2	116	659
Material taken at L2	F	Gets F	60	1056
Food eaten, material held	N 3	Places M at N	106	892
At nest	M 1	Gets M at 1	38	992
Material taken at L1	N	Places M at N	38	1240
At nest	M 2	Gets M at 2	118	1340
Material taken at L2	F	Gets F	58	1619
Food eaten, material held	N 3	Places M at N	101	1465
At nest	M 1	Gets M at 1	33	1565
Material taken at L1	N	Places M at N	36	1827
At nest	M 2	Gets M at 2	129	1927
Material taken at L2	F	Gets F	61	2198
Food eaten, material held	N 3	Places M at N, stung	106	1606
At nest	M 2	Gets M at 2	123	1706
Material taken at L2	F	Gets F	54	2098
Food eaten, material held	N 3	Finishes N	99	1946

****

6. Concluding Remarks

The way PERCY makes decisions has now been explained *in* detail. This account completes the series of articles that have described the PERCY simulation as an example of the purposive system. The term refers to a general structure for integrated systems, with a logical organization that reflects the game against the environment. The structure disengages the making of decisions from their execution. It bases decision making on utility evaluations. It clarifies the role of generalization in handling tasks. And above all, its way of operating faces squarely the problem of incomplete knowledge of the task environment.

Each of these points is illustrated by the PERCY simulation. The present article, which treats the problem and the process of decision making without concern for the course of action touched off by a decision, reflects the separation of these two aspects of system operation. PERCY'S utility evaluations are based explicitly on the payoffs associated with feeding, progress on the nest, and meetings with the Stinger; they are the only factors influencing its decisions. And generalization in dealing with the task takes the form of equivalences among decision stages that select the same set of targets for immediate attention.

In consequence of this design, PERCY achieves good performance in a task calling for a long series of decisions. Moreover, its success is realized in spite of its very limited capacity to obtain or store information about the environment, or to explore the implications of the little information it possesses.

To turn this last point around, the simulation shows how an organism with a rather primitive Central Nervous System can operate purposefully in an environment even though the program it is born with cannot contain specific information about that environment. Thus the simulation may be treated as a theory of how creatures that are relatively low in the evolutionary scale can show selective behavior on tasks of moderate complexity. Utility functions based on payoffs can readily account for the "drives" that are assumed to guide animal — and sometimes human — behavior.

Indeed, the way people tackle well understood tasks may be closer to the PERCY approach than to the problem-solving framework that underlies much of *the work* on *robots*. Artificial intelligence, like human intelligence, undoubtedly ought to incorporate both ways of planning.

* * * * *

REFERENCES:

(1) Fikes, R. E., Hart, P. E. and Nilsson, N. J. "Some New Directions in Robot Problem Solving," Machine Intelligence 7. (Edinburgh University Press, 1972).

(2) Jacobs, W. "A Structure For Systems That Plan Abstractly," AFIPS Conference Proceedings, Vol. 38 (1971), pp. 357-364.

(3) ________. "Control Systems in Robots," Proceedings ACM 25th Anniversary Conference, Vol. 1 (1972), pp. 110-117.

(4) ________. "How a Bug's Mind Works," Cybernetics, Artificial Intelligence, and Ecology. (Spartan Press, 1972).

(5) Munson, J. H. "Robot Planning, Execution and Monitoring in an Uncertain Environment," Advance Papers of the Second International Joint Conference on Artificial Intelligence. (The British Computer Society, 1971).

(6) Slagle, J. R. Artificial Intelligence: The Heuristic Programming Approach. (McGraw-Hill, 1972).

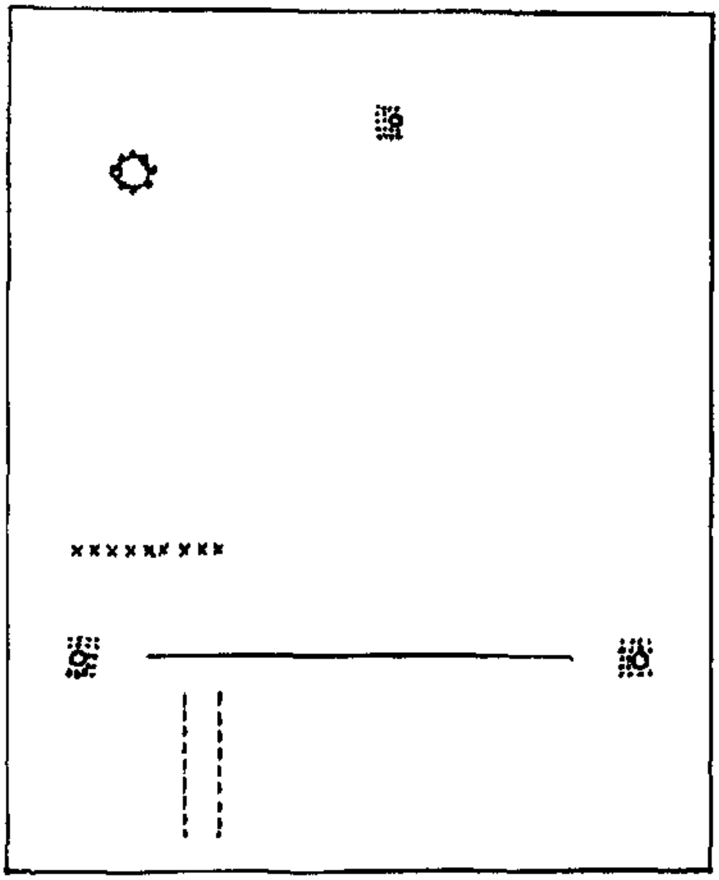

Figure 1: Diagram of PERCY's environment; symbols have the following meaning:

o landmark

::: cluster of material

nest site

— wall

--- path of food

xxx path of Stinger

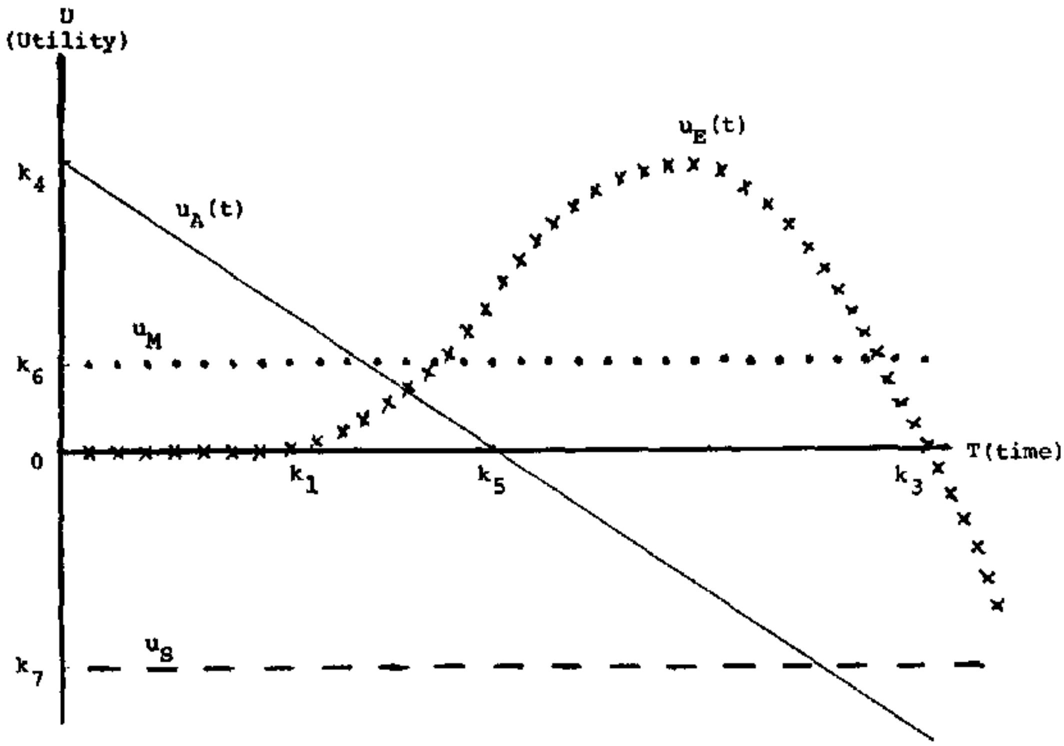

Fig. 2. PERCY's utility functions have the forms,

$$u_E(t) = 0 \qquad t<k_1$$

$$= k_2(t-k_1)^2(k_3-t) \quad t\geq k_1$$

$$u_A(t) = k_4(1-t/k_5)$$

$$u_M = k_6$$

$$u_S = k_7$$

where k_1, $k_2,\ldots,k_7$ are suitable constants.

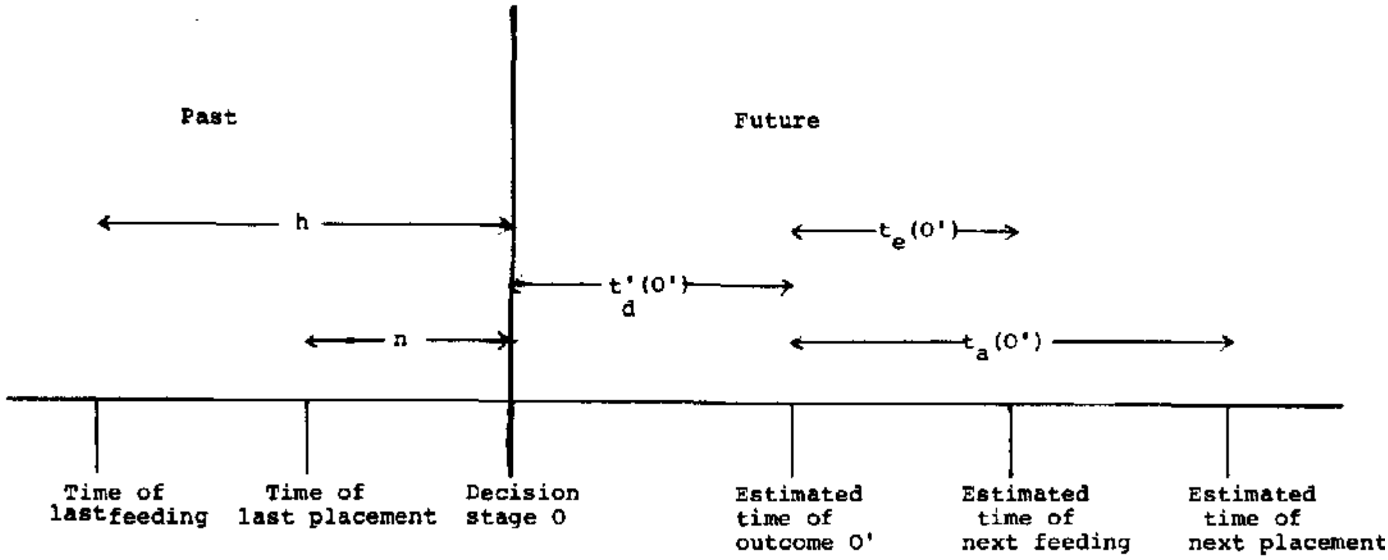

Fig. 3. The diagram shows how PERCY, at decision stage O, estimates the total elapsed time between its last and next feedings, and the total elapsed time between its last and next placements; in both cases on the assumption that O' occurs after decision d is made. The estimate for time between feedings is the sum of

h - the actual time since the last feeding; equal to zero if feeding occurred at outcome O.

t'_d - estimated time till O' occurs.

t_e - estimated time from O' till next feeding; equal to zero if feeding occurs at O'.

An analogous sum holds for placements at the nest.