Expert System Consultation Control Strategy

James Slagle and Michael Gaynor *

Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, D.C. 20375

Abstract

User interfaces to expert systems represent a bottleneck since consultation time is proportional to the amount of information the system asks the user to supply. An efficient, rather than exhaustive, strategy to direct user questioning will reduce consultation time and effort. An intelligent strategy to minimize questioning, the merit system, has been successfully implemented in *Battle*, an expert consultant system developed for the Marine Corps. The merit strategy enables *Battle* to focus the consultation process on the most meritorious questions allowing the military commander to respond quickly awith the most pertinent information. The merit system, originally defined for logical functions in the *Multiple* program, has been extended to the *Mycin* style of propagation and to the method of subjective Bayesian assignments used by *Prospector*. A procedure for merit calculations with any differentiable, real-valued assignment function is presented. Our experience has shown that merit values provide an efficient flow of control for expert consultation.

I Introduction

This paper reports on the consultation control strategy of a computer based intelligent decision aid system called *Battle* [10], developed for the United States Marine Corps. The objective of *Battle* is to improve the Marine Integrated Fire and Air Support System (MIFASS) by providing timely recommendations for the allocation of a set of weapons to a set of targets.

In a time -critical expert consultant system, the consultation must be quick yet relevant to the decision being made. Many military expert consultant systems are time-critical, for example, systems such as *Battle* that allocate weapons to targets. Other time-critical military systems would include systems for classifying images, submarine combat systems, multisensor information integration systems, and operational planning systems. An expert system in mineral exploration, for example *Prospector* [2], is not time-critical since the mineral being sought has been in the ground millions of years and will not go away soon. Some expert systems in medicine are not time-critical, *Mycin* [4] for example, but an emergency system would be.

When a system such as *Mycin* or *Prospector* questions the user, it uses a depth-first (local) search strategy. It will persist with a line of questioning that has become seemingly irrelevant. However, when a time critical expert consultant system is questioning a user it is essential that it asks questions that are highly relevant and quickly answered. Ideally it would use a best-first (global) strategy. The user of a time-critical system may know he has only five minues to make a decision. He would become highly frustrated with a system asking seemingly irrelevant or time consuming questions when he knows there are better questions to be asked. When expert consultant systems have thousands rather than hundreds of rules even non-time-critical systems will need some means of asking relevant and quickly answered questions. The merit system presented in this paper allows an expert consultant system of the *Battle*, *Mycin*, or *Prospector* type to ask questions in a best-first manner.

User interfaces present a bottleneck for most expert systems; consultation time is roughly proportional to the number of questions directed to the user. Older consultant systems generally follow an exhaustive depth-first network traversal to direct the consultation process that could ask hundreds of questions of which a handful would be really pertinent. A system that asks only pertinent questions, however, allows substantial savings of time by avoiding unnecessary questioning. The *Battle* decision aid [5] uses the merit system, a best-first strategy, to direct its consultation sessions efficiently. This is quite important for time critical applications such as the U.S.Marine Corps commander using the *Battle* weapon assignment program in combat.

II Other Expert System Consultation Strategies

Expert systems such as *Battle*, *Mycin*, and *Prospector* represent knowledge as a set of *propositions*. Each proposition has a *value* representing its likelihood. A proposition may have *antecedent* propositions from which its value may be inferred, and may itself be an antecedent of *consequent* propositions. We call the numerical dependence of the value of a consequent proposition on the values of its antecedents the *assignment function* of the consequent. A proposition with no consequents, called a *top proposition*, represents the result of the inferencing process. The data from which the result is calculated are represented by *askable* propositions in the network, whose values may be supplied by the user. Other propositions in the network, whose values the user is unlikely to know, are *unaskable*. Typically top propositions are unaskable.

The distinction between askable and unaskable propositions is not always clear, since users differ in expertise and different information is available in different instances. An askable proposition may have antecedents and an assignment function for those cases when the user cannot supply its value.

Several techniques have been adopted to try to optimize the expert consultation process within the framework of a depth-first traversal. The simpler of these methods generally eliminate questioning about any node whose final value is established. The MARK IV control strategy of *Prospector* [2] first chooses a propo-

Also of Bloomburg University, Bloomsburg, PA

sition for consideration whose antecedents are then evaluated by a function. The MARK IV control strategy apparently works well for the *Prospector* system. It suffers from several shortcomings :

1. Optimization is within the framework of a depth-first traversal. A node, once traversed by the depth-first mechanism, will never be reconsidered.
2. The node selected may not be the optimal proposition for consideration within the entire network.
3. The four criteria evaluated by the function do not identify the antecedent with the largest potential to produce changes in the consequent probability.

The *Casnet* (causal-associational network) system [14] provides a more extensive search of the inference network than does the *Prospector* MARK IV strategy and considers costs as well. The two control strategies used by *Casnet* are :

1. Selection of the node with the maximum weight-to-cost ratio, and
2. Selection of the node with the maximum weight subject to certain constraints on cost.

Casnet concentrates on nodes that seem to be most consistent with the remaining nodes in the network. Since the objective of expert consultation is to infer the value of a top proposition or propositions, a more appropriate estimate of the weight of a node can be determined by its potential influence on a top proposition. The merit control strategy of the *Battle* system assigns a weight to each node corresponding to its ability to alter the value of a top proposition.

III. The *Multiple* Control Strategy

Multiple (MULTIpurpose Program that LEarns) has been implemented for the game of Kalah and for theorem proving [7]. With a two step algorithm, *Multiple* uses merit values to select the next proposition using merit values :

1. The system *sprouts* from an untried proposition with the largest merit value on its proposition tree and calculates merit values for all its children.
2. At each level only the best merit value is *backed up* to the top proposition. At the top level, the untried proposition with the highest merit value is identified.

Assume that there exists a proposition tree with a top proposition G, and antecedents G_i. Each G_i may have antecedents designated G_{ij}. Each subscript indicates an additional level down the tree. The values stored at G, G_i, and G_{ij} are named P, P_i, and P_{ij}. Each value P is given by the assignment function $f(P_1, P_2, \cdots P_n)$ applied to its antecedent values P_i. The merit of an untried proposition $G_{ij...st}$ is defined as :

$$\textit{Merit Value of } G_{ij...st} \equiv \left| \frac{\partial P}{\partial (C_{ij...st})} \right| \qquad 3.1$$

where ∂P is the change in the value (generally, but not restricted to, a probability) of the top proposition G, and $\partial(C_{ij...st})$ is the cost of expanding the untried proposition $G_{ij...st}$. Both positive and negative values are equally significant. The merit of proposition H is the expected ratio of two terms if H is expanded:

1. The absolute value if the change in value of the top proposition.
2. The cost of expanding H.

Thus expanding a proposition with maximum merit should lead to good results.

A more useful form of the merit formula is derived by application of the chain rule :

$$\left| \frac{\partial P}{\partial (C_{ij...st})} \right| = \left| \frac{\partial P}{\partial (P_i)} \frac{\partial (P_i)}{\partial (P_{ij})} \cdots \frac{\partial (P_{ij...st})}{\partial (C_{ij...st})} \right| . \qquad 3.2$$

The last factor, called self-merit, introduces cost considerations. The self-merit of proposition $G_{ij...st}$ is a measure of the expected change in the proposition's value $P_{ij...st}$ with respect to the cost of considering that proposition, $C_{ij...st}$. Each of the remaining factors in the chain rule expansion is called an edge-merit. It measures the change in the value of a consequent proposition due to the change in the value of an antecedent proposition. An edge-merit value for a specific antecedent/consequent pair may be calculated by evaluating the derivative of the assignment function associated with the edge linking that pair.

Multiple algorithm merit calculations require time proportional to tree-depth since the merits of only the newly sprouted propositions need to be computed and backed up. Merit calculation is completely analogous to moving up a tree of winners.

IV Merit In An Inference Network

The most meritorious propositions in a network are those propositions that are likely to have the most cost-effective influence on a top proposition. Using the *Multiple* algorithm, *Battle* explores the most meritorious propositions until it encounters an askable one. The user is prompted for a value for this proposition. After receiving this information or discovering the user cannot provide the information, the system proceeds to discover the next unasked, askable proposition of highest merit. Such a process is iterated until no more propositions remain with a merit greater than some cutoff value.

The cutoff merit value is a user-defined parameter used to limit the number of questions asked. A cutoff value does not alter the order of questioning. The user may vary this value during the consultation process. Consultation continues only while a proposition whose merit value exceeds the cutoff can be found.

Merit values are calculated for a small set of nodes with a common parent in each sprouting operation. These values are maintained in a tree of winners, and each newly calculated value is compared to the best values from all previously traversed nodes. The merit system thus allows an unconstrained network traversal that moves to a most meritorious node wherever it may be in the network.

We recognize this traversal may be disconcerting to a naive user, but then exhaustive questioning may be tedious or even dangerous to a military commander with a time-critical task to execute. It is a tradeoff between time to question and the apparent completeness of the questioning. When it is more desirable to question the user thoroughly on a specific topic before moving on to the next issue the user should order a depth-first traversal. **V Self-Merits**

How is a merit value determined? Two processes are involved: assignment of a self-merit value and calculation of the edge-merits. The product of the edge-merits and the final self-merit along a path from the top proposition to a node provide the merit value of the node (see equation 3.2).

Assignment of self-merit values to nodes is initially the responsibility of the network designer (domain expert). Large self-merit values should be assigned to nodes whose parameters are easily specified by a user

(low cost), and whose value is likely to change a great deal. Self-merits of unaskable nodes should reflect the expected change in the node's associated value with respect to the cost of calculating that node's value or expanding the traversal to its antecedents.

Several sets of self-merits to describe accurately the benefit/cost ratio for examining various nodes by different subsets of users may be needed. It is important that self-merits be assigned reasonable values relative to each other in the initial implementation. After a consultant system has been running for a reasonable period of time, empirical data may yield self-merit values. Although the self-merits are generally assigned in an ad-hoc fashion, our experience has shown that it is beneficial to use precise mathematical formulas to complete the merit value calculation. Some edge-merit formulas are derived in the following sections. **VI Logical Function Edge-Merits**

A consequent whose truth is contingent on verification of all its antecedents is the logical **AND** of those antecedents. In a general probabilistic approach, assuming all antecedents are independent, the **AND** function may be described as :

$$P(H) = P(E_1)\,P(E_2)\cdots P(E_n) \qquad 6.1$$

The probability assigned to a consequent, H, given the current probabilities for each of its antecedents, E_j ($j = 1, \cdots, n$), is the product of those antecedent probabilities. Differentiating the consequent probability with respect to a single antecedent and substituting back from equation 6.1, the formula for **AND**-edge-merit is derived as :

$$\frac{\partial P(H)}{\partial P(E_j)} = \frac{P(H)}{P(E_j)} \qquad 6.2$$

Equation 6.2 depends on the values of only the antecedent/consequent pair of the edge in consideration. This is a most convenient form to express edge-merit calculations.

An **OR** function is logically true when any of its antecedents are true. Again assuming the independence of antecedent values, the probabilistic **OR** function may be written as :

$$P(H) = 1 - [1 - P(E_1)] \cdots [1 - P(E_n)] \qquad 6.3$$

The consequent probability is the complement of the product of the complements of all current antecedent probabilities. Differentiating with respect to an individual antecedent, and substituting back from equation 6.3, the **OR**-edge-merit is found to be :

$$\frac{\partial P(H)}{\partial P(E_j)} = \frac{1 - P(H)}{1 - P(E_j)} \qquad 6.4$$

It may be shown that both the **AND**-edge-merit and the **OR**-edge-merit approach finite limits as $P(E_j)$ approaches 0 or 1.

In a probabilistic scheme, the logical **NOT** may be defined as :

$$P(H) = 1 - P(E) \qquad 6.5$$

Although not required for choosing among antecedents (since such an edge has but one antecedent), the **NOT**-edge-merit becomes important in networks with multiple consequents of propositions (see section 8).

VII Subjective Bayesian Edge-Merits

Prospector uses a subjective Bayesian method of assignment [1], relating each antecedent to its consequent as an independent piece of evidence. When the set of top propositions is mutually exclusive and exhaustive, the subjective Bayesian method is not practical [3]. In general, however, subjective Bayesian assignments provide a useful method for the evaluation of evidence by an expert consultant system. A brief review of the subjective Bayesian method as well as a derivation of the edge-merit for that assignment procedure is now presented.

Assume that there exists a hypothesis, H, and n independent sources of evidence E_j (for $j = 1, \cdots, n$) that may either support or deny the hypothesis. The hypothesis H is called the consequent of each E_j, and each E_j is an antecedent of H. We suppose that for each antecedent E_j the current (probability) value $P(E_j)$, the prior probability $P_0(E_j)$, and prior probabilities $P(H|E_j)$ and $P(H|\bar{E}_j)$ of the consequent given the antecedent and its negation are known, and that the prior probability $P_0(H)$ of the consequent is known. We summarize the procedure derived in [1] for calculating the current probability $P(H)$ of the consequent.

The probability estimator $P_j(H)$ of the consequent given the current value of E_j is calculated by linear interpolation between the known values.

$$P_j(H) = P_0(H) + M_j(P(E_j) - P_0(E_j)), \text{ where} \qquad 7.1$$

$$M_j = \begin{cases} \dfrac{P_0(H) - P(H|\bar{E}_j)}{P_0(E_j)} & \text{if } P(E_j) \le P_0(E_j), \\ \dfrac{P_0(H) - P(H|E_j)}{P_0(E_j) - 1} & \text{if } P(E_j) > P_0(E_j). \end{cases} \qquad 7.1a$$

The probability estimators are combined more conveniently by transforming them to odds estimators. The combined odds $O(H)$ of the consequent is transformed to an expression of $P(H)$ in terms of $P_j(H)$ and $P_0(H)$.

The edge-merit for subjective Bayesian assignment may be expanded with the chain rule as

$$\frac{\partial P(H)}{\partial P(E_j)} = \frac{d\,P(H)}{d\,O(H)} \frac{\partial O(H)}{\partial O_j(H)} \frac{d\,O_j(H)}{d\,P_j(H)} \frac{\partial P_j(H)}{\partial P(E_j)}. \qquad 7.2$$

The first and third factors of Equation 7.2 may be simplified by differentiating the odds equation (see [11] for details). The second factor in the edge-merit expansion may be found by differentiation of the combined odds equation [11]. since all factors except $O_j(H)$ are constant with respect to $O_j(H)$. The final factor of the edge-merit expansion corresponds to the slope M_j of the linear interpolation in Equation 7.1. Substituting these yields the edge-merit for subjective Bayesian assignment,

$$\frac{\partial P(H)}{\partial P(E_j)} = \left[\frac{1 + O_j(H)}{1 + O(H)}\right]^2 \frac{O(H)}{O_j(H)} M_j \qquad 7.3$$

$$= \frac{(1 - P(H))\,P(H)\,M_j}{(1 - P_j(H))\,P_j(H)}$$

Some boundary conditions in this formulation are notable. If $P_j(H)$ approaches zero or one, the value of the edge-merit will approach a finite limit, although it is undefined at the limit points. In practice, a small offset of $P_j(H)$ will simplify the calculation. Also, the slope M_j is discontinuous at $P(E_j)=P_0(E_j)$. Currently, a value intermediate between the interpolant slopes is used.

VIII Multiple Consequents

In an inference network individual nodes may have any number of consequents. Suppose that a top proposition G has two antecedents, G_1 and G_2, and that both G_1 and G_2 share a common antecedent, G', as illustrated in figure 2. Assume that G' is independently chosen as the

most meritorious antecedent of each of G_1 and G_2. The *Multiple* algorithm backs up merit values in a tree of winners, always selecting the maximum value. When the merit value backed up at G_1 is compared to the value backed up at G_2, both nodes will possess the backed up merit of G'.

It would be inaccurate to simply back up to G the maximum of the merit values backed up at G_1 and G_2 since either choice represents the selection of proposition G'. The value backed up to G should represent the combined influence of G' on G. Since the effects of G' through its parents might be synergistic or antagonistic, the sum of the signed merit values for G' calculated independently through each of G_1 and G_2 should be backed up to G. When these values are of opposite signs, the antecedents of both G_1 and G_2 must be reexamined. A sibling of G' initially thought to have a lower merit value than G' might back up a larger merit value to G.

To see that adding the signed merit values is correct mathematically as well as intuitively, we may apply the chain rule for functions of several variables to see that the merit of node G' is

$$\left|\frac{\partial P(G)}{\partial C(G')}\right| = \left|\left[\frac{\partial P(G)}{\partial P(G_1)}\frac{\partial P(G_1)}{\partial P(G')} + \frac{\partial P(G)}{\partial P(G_2)}\frac{\partial P(G_2)}{\partial P(G')}\right]\frac{\partial P(G')}{\partial C(G')}\right|$$

while the signed merit values of G' as calculated through G_1 and G_2 are

$$\frac{\partial P(G)}{\partial P(G_1)}\frac{\partial P(G_1)}{\partial P(G')}\frac{\partial P(G')}{\partial C(G')} \text{ and } \frac{\partial P(G)}{\partial P(G_2)}\frac{\partial P(G_2)}{\partial P(G')}\frac{\partial P(G')}{\partial C(G')}.$$

Thus to select a most meritorious node below G we form the set of most meritorious nodes below the antecedents G_j of G, and sum the merits of nodes that appear more than once. As before, we select the merit that is largest in absolute value to get a most meritorious node below G.

Whenever a summation of signed merit values occurs, the merit values backed up the tree before the point of summation might not represent the most meritorious propositions. One possible solution is to back up the K-best merit values at each level, in the hope that the most meritorious value is included among them. In our experience this has not been necessary. Even though *Battle* backs up only the single best merit value at each level, the merit calculation guides it to appropriate questions.

IX Concluding Remarks

Merit calculations may be performed for an inference network whose assignment functions are of many different types. We have extended merit to handle both *Mycin* and *Prospector* inference mechanisms, as described in reference [11]. A node whose value is assigned by a simple logical function, probabalistic **AND** for example, might assign a value to its consequent through the subjective Bayesian method. At each edge the edge-merit function appropriate to that edge is applied. The units of the various edge-merit values all cancel, leaving a final merit value expressed in the units used by the top proposition over cost. Differentiable, real-valued expert-defined assignment functions are easily incorporated into the inference network with the merit control strategy.

We have compared merit with the J^* algorithm used by *Prospector* by generating the values of two antecedents on a single node over a range of 26 probabilities. In no case did J^* choose the most meritorious antecedent, see [6] for more detail.

In a future consultant system, propositions and assignment functions supplied by an expert will be linked into an inference network. Commonly used assignment methods will be available as system defined functions. The expert, however, will be able to introduce new assignment functions wherever necessary. The system will derive the form of the edge-merit function for these expert-defined assignment functions.

Merit values may be employed to order antecedents within a depth-first traversal of an inference network, or to guide a best-first strategy. The two-step *Multiple* algorithm for locating a most meritorious node was designed for implementation with large trees where an exhaustive search is not practical. In an inference network, however, an expert system might do an exhaustive merit analysis, examining each askable proposition on the network in search of the most appropriate one for investigation. Such a searching procedure requires more time to find a most meritorious proposition on the network, but it guarantees that consultation will focus on a most meritorious node.

A reduction of inconsequential propositional values requested from the user will increase the effectiveness of the consultation process, especially in time-critical applications such as the tasks faced by military commanders. The *Battle* system uses merit values to direct such an intelligent consultation session. Since merit, a function of both the cost and potential benefits of considering a proposition, is easily calculated by a computer, introduction of the merit value heuristic should result in reduction of consultation time and effort.

Acknowledgement

This work was sponsored by the Office of Naval Research.

References

[1] Duda, R.O., Hart, P.E., and Nilsson, N.J. Subjective Methods for Rule-based Inference Systems. Proc. National Computer Conference 45, AFIPS Press, pgs. 1075-1082. Artificial Intelligence Center, SRI Int., Menlo Park, CA., 1976.

[2] Duda, R.O., Hart, P.E., Konolige, K., and Reboh, R., A Computer-based Consultant for Mineral Exploration. Artificial Intelligence Center, SRI Int., Menlo Park, CA., (Sept. 1979).

[3] Pednault, E.P.D., Zucker, S.W., and Muresan, L.V., On the Independence Assumption Underlying Subjective Updating. Artificial Intelligence 16 (May, 1981), pgs. 213-222.

[4] Shortliffe, E.H. Computer-based Medical Consultations: *mycin*, American Elsevier, New York, 1976.

[5] Slagle, J.R., Cantone, R., and Halpern, E. Battle : An Expert Decision Aid for Fire Support Command and Control. NRL Memorandum Report 4847, (July 8, 1982).

[6] Slagle, J.R., and Halpern, E. An Intelligent Control Strategy for Computer Consultation. NRL Memorandum Report 4789, (April 8, 1982).

[7] Slagle, J.R. and Farell, C.D. Experiments in Automatic Learning for a Multipurpose Heuristic Program. Comm. of the ACM, 14, (February, 1971), pgs. 91-99.

[8] Weiss, S.A., Kulikowski, C.A., Amarel, S. and Safer, A. A Model-based Method for Computer-aided Medical Decision Making. Artificial Intelligence 11, (August, 1978), pgs. 145-172.