Antonio Leal
[61]

Technical Report NAVTRAEQUIPCEN 78-C-0107-1

KNOWLEDGEABLE OPPONENT MODELS FOR ENEMY SUBMARINE TACTICS IN TRAINING SIMULATORS

Don May, Efraim Shaket, Antonio Leal

Quarterly Progress Report September 1978 - November 1978

November 1978

DoD Distribution Statement

Approved for public release;
distribution unlimited.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NAVTRAEQUIPCEN 78-C-0107	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Knowledgeable Opponent Models for Enemy Submarine Tactics in Training Simulators		5. TYPE OF REPORT & PERIOD COVERED September 1978 - November '78
		6. PERFORMING ORG. REPORT NUMBER PQTR-1071-78-11
7. AUTHOR(s) Don May, Efraim Shaket, Antonio Leal		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Perceptronics, Inc. 6271 Variel Avenue Woodland Hills, CA 91367		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Code N-71 Orlando, Florida 32813		12. REPORT DATE November 1978
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes four models which show promise for simulating a knowledgeable opponent for enemy submarine tactics in training simulators. While the models are primarily designed to simulate an opponent by selecting his decision alternatives, the models can also be used to simulate friendly forces as well. The four approaches are: (a) the elicited probability approach, (b) the adaptive decision modeling approach, (c) the heuristic search approach, (d) the production rules approach.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The elicited probability approach is based on a transition matrix which relates the current state vector to a set of state transformation operators. The matrix elements are conditional probabilities elicited from experts (or can be determined by collecting statistics). The state transformation operators are rules which dynamically change the state of the simulation when selected by the application of Bayesian algorithms. The basic mechanism can be used to select operators in a hierarchic manner by incorporating them in higher level transformation operators.

The adaptive decision appraoch uses pattern recognition to learn opponent behavior from instructor opponent controllers (operator). This approach is based on decision modeling and utililty theory. The adaptive learning algorithms act as a pattern classifier and is used to identify biases in operator decision policy as a response to classes or patterns in the input data. The Multi-attribute Utility (MAU) model is used to capture the decision behavior of the operator. In the MAU model, the consequences of every action are considered to be decomposable according to a single common set of attributes.

The heuristic search approach provides a mechanism by which the opponent responds to actions taken by friendly forces with a course of action which leads to the achievement of some enemy goal. A state space model is used to represent the problem domain. The states are a complete description of the tactical situations as they exist at a particular instant of time. An action converts one state into another. The opponent asks the question, "What sequence of actions can transform the current state into a desired goal state?" The basic search algorithm begins at a start node and expands successive nodes until a goal node is encountered. Then the path from the initial node to that goal node is the solution sought. Heuristic search algorithms use domain specific knowledge to guide the search. Heuristic knowledge may apply to node expansion or to path evaluation. In either case heuristic knowledge is used to reduce the searching effort. Specific heuristic search algorithms are discussed.

The production rules approach uses sets of situation-action pairs, called "productions" to transform the current state to the next state. The productions represent the problem specific knowledge. In addition to productions, the production rule system contains a triggering mechanism that applies those that are applicable-causing the situation to change. AND/OR graphs represent one kind of production rule system. Production rule systems resemble the human reasoning process, and can be used to answer the questions of how or why a particular conclusion was reached by the system. Also, the user can hypothesize a conclusion or desired final state and use the productions to work backward toward an enumeration of the facts that would support the hypothesis.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

SECTION I

INTRODUCTION

GENERAL

This report gives an overview of four types of models for simulating an intelligent opponent for enemy submarine maneuvers within the Navy Submarine Combat System Trainers (SCST). These models were selected for discussion because they are most promising. With variations of each model, the possibility of combining models, and the use of sub-models, many combinations are possible.

The specific objectives of the program are:

a. Analysis of current requirements of Navy submarine tactical trainers with respect to the tactical behavior of simulated enemy submarines.

b. Identification of potential knowledgeable opponent model algorithms and techniques applicable to submarine tactics.

c. Evaluation of each potential model in terms of tactical maneuver capabilities, model trainability, software requirements, trainee performance measurement, and required research and development.

REQUIREMENTS ANALYSIS

The requirements for knowledgeable opponent models were given in an earlier report (Leal, Purcell, Thomas, 1978). Figure 1-1 is reproduced from this report. The flowchart shows a decision diagram for submarine commander.

The following is a summary of the decisions that must be made by the knowledgeable opponent model:

a. Overall Mission

Transit
Patrol (barrier, broad, choke point shipping or sub-routes)
Type of sub (fixed decision)

b. Has friend been detected.

c. Has friend detected enemy (legitimate opponent).

d. Evade or not (or hide in deep water).

e. Tactic selection (course, depth, heading, maneuvers, etc.)

f. Strategy selection

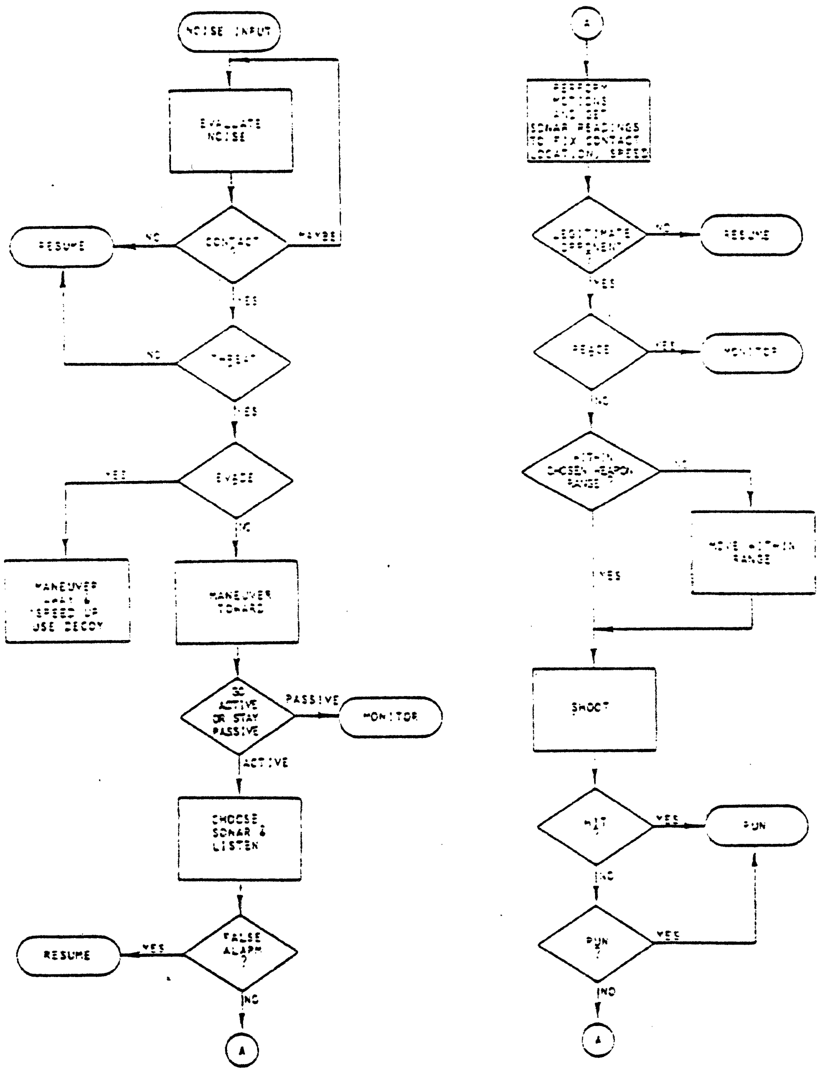


FIGURE 3-1. DECISION DIAGRAM FOR SUBMARINE COMMANDER.

g. Subgoals.

h. What sensors to use (active or passive).

i. What weapons to use if at all, realtack.

j. Use decoy or not.

k. False alarm?

l. Move within range, close to investigate.

m. Track friend.

n. Clear its baffles, turn and look behind it.

REQUIREMENTS ANALYSIS (continued)

Since the opponent and friend have essentially the same decision structure, the same model which is developed for the opponent can also model the friend. This brings up a number of interesting and useful possibilities:

a. Play one model against the other. By doing this, it will be easier to debug the software. Also, it is possible to develop a set of performance baselines which can be used for further model development and to develop evaluation guidelines.

b. The opponent model easily contains a model of the friend. Further levels of recursion are possible. For example, the friend can be aided by an opponent model which contains a friend model.

c. Different models can play each other to evaluate which model is best.

d. Different parameter values can be set for each model and the models can play each other in order to evaluate the effectiveness of various strategies and various assumptions regarding opponent capabilities.

It should be emphasized that when the same model is used for several purposes, different behavior can be created by varying model parameters.

POTENTIAL MODELS

From an analysis of the requirements of the knowledgeable opponent model and from an analysis of existing simulation and modeling techniques, four major approaches have been identified which show potential for model implementation. These approaches are:

a. Elicited probability approach

b. Adaptive decision modeling approach

c. Heuristic search approach

d. Production rules approach

The remainder of this report discribes each approach and shows its applicability to the ASW problem. The next phase of the program will establish specific design and implementation relationships between each approach and the simulation requirements, in addition to the evaluation of each model.

SECTION II

THE ELICITED PROBABILITY APPROACH

INTRODUCTION

The elicited probability approach to scenario generation is logically similar to Bayesian information processing. However, instead of aggregating expert opinions to estimate the probabilities of complex events in a real world, we use those probabilities to simulate the real world. The scenario can represent a knowledgeable opponent as well as the environmental conditions.

The technique is based on a transition matrix which relates the current state vector to a set of state transformation operators. The components of the transition matrix are the conditional probabilities of each state transformation operator (or rule which dynamically changes the environment), given the value of each state vector variable. These conditional probabilities can be estimated by experts on the behavior of the environment being simulated, or statistics can be collected to determine them.

The next step is to compute the conditional probabilities of each state transformation operator, given the current state vector. The actual state transformation operators applied to the current state vector are chosen on the basis of these probabilities by means of a Monte Carlo selection procedure. Alternatively, the transformation operator with the highest probability could be selected. The state transformation operators are then executed to obtain a new state vector.

The basic mechanism can be used to select independently n state transformation operators, one from each of n sets of operators, to be applied in parallel. The basic mechanism can also be used to select operators in a hierarchic (dependent) manner by incorporating them in higher level transformation operators.

THE BAYESIAN MODEL

The Bayesian model provides a mechanism for transforming a state space representation of the environment at discrete times. Figure 2-1 schematically represents the model. The mechanism is similar to that used to aggregate expert opinion in Bayesian information processing systems. The current state vector is used to select probabilities from a conditional probability matrix relating a set of state transformation operators to the current state vector. These probabilities are aggregated using Bayes' theorem to obtain a transformation probability vector. This vector contains the probability that, given the current state vector, each transformation operator in the set will be selected. Operators are selected from the set using a Monte Carlo selection procedure based on these probabilities. Finally, the selected operator is invoked to transform the current state vector to the next state.

In most situations, a given operator will transform only a subset of the state vector. Therefore, it will be necessary to invoke a number of transformation operators "simultaneously." These operators are selected by

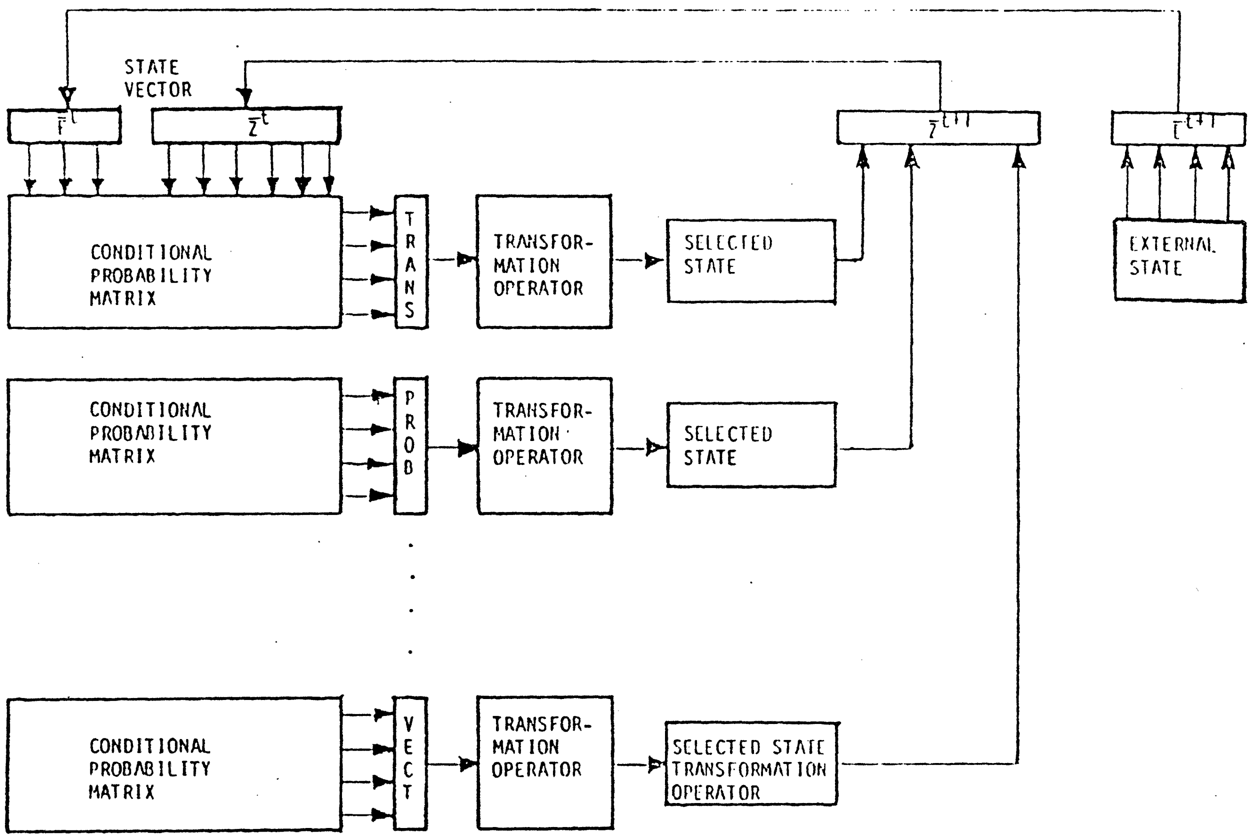


Figure 2-1. Bayesian Scenario Generation Model Schematic

repeating the above process, using different sets of operators and correspondingly different conditional probability matrices.

THE STATE VECTOR

The state of the scenario generator (and, thus, the scenario) at time t is represented by a state vector $\overline{Z}^t$. This vector is comprised of discrete state variables:

$$\overline{Z}^t = [z_1^t, z_2^t, \ldots, z_n^t] \tag{2-1}$$

The values of all state variables define the state of the scenario at any time, t.

THE CONDITIONAL PROBABILITY MATRIX

The conditional probability matrix (P-Matrix) relates the current state vector, $\overline{Z}^t$, to a set of state transformation operators. The components of the matrix are the conditional probabilities that, given the occurrence of the state transformation operator T_j, the ith state variable had a value z_i^t: $p(z_i^t|T_2)$. These probabilities are estimated by experts on the behavior of the environment being simulated. The expert estimates can be based upon experience, upon real world measurements, upon theoretical models, etc. It is also possible to determine the conditional probabilities by collecting statistics during an actual training session in which the instructors are controlling opponent actions.

Two vectors of *a priori* probabilities, also estimated by experts, are required. The components of the first vector, $\overline{P}_T$, are the *a priori* probabilities that each state transformation operator will be selected. They are represented thusly:

$$\overline{P}_T = [p(T_1), p(T_2), \ldots, p(T_m)] \tag{2-2}$$

The components of the second vector, $\overline{P}_z$, are the *a priori* probabilities of the occurrence of each state component of the $\overline{Z}$ vector. They are represented as follows:

$$\overline{P}_z = [p(z_1), p(z_2), \ldots, p(z_n)] \tag{2-3}$$

The *a priori* probabilities don't have to be estimated with great precision because, as the scenario unfolds, they have less and less effect over the behavior of the scenario.

PROBABILITY AGGREGATION

The probability that a state transformation will be selected is computed by aggregating the conditional probabilities according to Bayes' rule:

$$p(T_j|\overline{Z}^t) = \frac{p(T_j)\ p(\overline{Z}^t|T_j)}{p(\overline{Z}^t)} \tag{2-4}$$

If we assume that the variables which comprise the state vector are independent, then

$$p(\overline{Z}^t) = \prod_{i=1}^{n} p(z_i^t) \tag{2-5}$$

and

$$p(\overline{Z}^t|T_j) = \prod_{i=1}^{n} p(z_i^t|T_j) \tag{2-6}$$

Thus, equation 2-4 becomes

$$p(T_j|\overline{Z}^t) = \frac{p(T_j)\ \prod_i p(z_i^t|T_j)}{\prod p(z_i^t)} \tag{2-7}$$

When equation 2-7 is implemented, the $p(T_j|\overline{Z}^t)$ are normalized; thus, the denominator in equation 2-7 is not needed.

The assumption that the variables which comprise the state vector are independent is a crucial one. The most practical way to meet this condition is to take care to define the state vector such that it is independent. If there are dependencies in the state vector, they may not noticeably affect the behavior of the scenario (e.g., environment, opponent's actions). This can be tested by using the model to generate behavior which is viewed by the person from whom the probabilities were elicited. If the behavior is not as desired, the elicited probability values can be fine-tuned until the proper behavior is obtained.

One technique of handling dependencies in the state vector is to also elicit the covariance matrix representing the correlation among state variables. This matrix can then be used in one of two methods:

a. The problem is transformed into a domain where independence holds.

b. The covariance matrices are used to derive weights to compensate for dependence.

Both methods have several disadvantages:

a. The covariance matrices are dependent on the order of processing state variables; a different covariance matrix must be used for each order.

b. The covariance matrices involve either asking people to estimate means and standard deviations, or polling a group of experts and collecting these statistics.

c. When the probabilities are subjectively determined (by elicitation), the precision of the problem is such that the covariance matrices may be meaningless.

In general, the complexity of using the covariance matrices seems to exceed that justified by meaning and relevance.

Another method of handling dependencies in the state vector is to construct a new set of variables based on permutations of some of the dependent variables. This approach is simple, but leads to a rapid increase in the size of the state space.

STATE TRANSFORMATIONS

There are virtually no restrictions on the kinds of state transformation operators which can be defined. A transformation operator may affect a single state variable and generate a constant output. It may also affect a large number of state variables and make use of a complex decision stragegy to determine their values. The transformation operator may even determine the value of a variable for several subsequent time cycles.

A transformation operator may make use of subsets of $\overline{Z}^t$ which were not used in selecting the operator. An operator may also make internal use of Bayesian aggregations based upon additional conditional probability matrices and subsets of $\overline{Z}^t$. Thus, hierarchies of transformatin operators can be established.

Each transformation operator affects a set of one or more state variables. The operators, in turn, are grouped according to which set of variables they affect. These sets of variables must be disjoint because, after a single operator is selected from each set, the selected operators are assumed to be invoked simultaneously. If the sets of variables are not disjoint, the order in which the selected operators are actually invoked will affect the value of the transformed state vector. However, non-disjoint sets of variables can be handled by establishing a hierarchy of operators within a "higher level" operator.

The selection of one state transformation operator from each operator set is made by means of a Monte Carlo selection procedure. The probabilities of occurrence of each operator in the set are normalized to obtain a discrete cumulative distribution function. A uniformly distributed pseudorandom number in the range [0,1] is then generated and its position in the distribution function is used to select the operator. Alternatively, the operator with the highest probability could be selected.

In some experimental applications, it may be useful or necessary to know the probability that a state variable will have a particular value. $p(z_k^{t+1}|\overline{Z}^t)$. By restricting the kinds of allowable state transformation operators to those that generate a constant (and unque) result, it is possible to obtain these probabilities directly from the scenario generator. If state transformation operator, T_j, outputs the same value for z_k^{t+1} whenever it is invoked, and only T_j outputs that vaue, then

$$p(z_k^{t+1}|\overline{Z}^t) = p(T_j|\overline{Z}^t) \tag{2-8}$$

If more complex transformation operators are used, $p(z^{t+1}|\overline{Z}^t)$ becomes more difficult to compute. A value can always be obtained, however, by making statistical measurements of the behavior of the scenario generator.

The current state vector, $\overline{Z}^t$, is transformed into $\overline{Z}^{t+1}$ by the (assumed) simultaneous invocation of all of the selected state transformation operators. If the state vector is properly designed, it is possible to use the Bayesian/Monte Carlo selection mechanism to choose all of these operators. However, in many instances it may be more convenient to use "external" mechanisms to select transformation operators for certain subsets of the state vector. These externally controlled state vector subsets will be collectively referred to as the $\overline{E}^t$ subvector (see Figure 2-1). Examples of externally controlled state variables would include clock driven variables such as day and night, high and low tides, and events which occur on a fixed schedule.

PROBABILITY ELICITATION

Previous research has shown that human experts are good at estimating conditional probabilities, but poor at aggregating them (e.g., Edwards, 1962). Accordingly, the present scenario generator uses conditional probabilities elicited from experts and aggregates them automatically. First, expert inputs are used to:

a. Describe the environment to be modeled in terms of relevant state variables.

b. Determine which variables are externally controlled and which are controlled by the Bayesian model.

c. Define all of the transformations which change the state variables.

Then, the expert is queried in detail to:

d. Estimate the a priori probabilities and the individual conditional probability which constitute the entire matrix.

The method of elicitation is simply to interview the expert and ask him the probabilities. Bond and Rigney (1966) were able to elicit almost 650 conditional probabilities associated with electronic trouble shooting in one hour using a simple questionnaire.

The process of probability elicitation is an iterative one which allows the expert to refine his estimates. That is, once the initial estimates are made, test scenarios are generated which allow the expert to see the consequences of his estimates. He is then asked to modify his estimates to make them more consistent with the desired behavior of the scenario generator.

ELICITED PROBABILITY APPROACH

Advantages of Strong Points

a. Simplicity; easy to develop, maintain, implement.

b. Generates a probabilistic opponent and environment.

c. Weights representing behavior are easy to elicit and to alter.

d. State oriented; easy to switch between manual and automatic operation.

Disadvantages or Weak Points

a. It is difficult to alter structural aspects due to the need to avoid dependencies in the state vector.

b. Difficult to insert logical statements to control the scenario.

c. The application of state transformation operators may be order dependent.

Examples of Elicited Probabilities. The following is a table of a priori and conditional probabilities of state elements given detection or non-detection. Note: Conditional probabilities are used as weights. Opponent's decision: Have I been detected by the home submarine?

State Elements		A Priori Prob of State	Have Been Detected	Have Not Been Detected
Home gone active Opponent detects signals	- Yes	.10	.75	.25
	- No	.90	.25	.75
Has opponent detected home sub	- Yes	.10	.40	.50
	- No	.90	.60	.50
Proximity of home submarine	- Very Close	.05	.40	.15
	- Near	.05	.25	.25
	- Mid-Range	.10	.25	.25
	- Far	.80	.15	.40
Environment noisy	- Yes	.5	.40	.60
	- No	.5	.60	.40
Thermal layers	- Several	.05	.20	.50
	- One	.15	.30	.30
	- No	.80	.50	.50
A priori probability of detection			.10	.90

Decision: What maneuver should I select?

State		A Priori Prob	Hide 1	Maneuver run	zigzag 3	proceed 4	attack 5
How far is home:	Undetected	.60	0	etc.			
	Very Close	.15	.25				
	Near	.15	.30				
	Mid-range	.05	.25				
	Far	.05	.20				
Has opponent decided that home has detected him?	- Yes	.15	etc.	etc.			
	- No	.85					
War or peace?	War	.01	etc.	etc.			
	Peace	.99					
Water depth:	Shallow	.10	etc.	etc.			
	Normal	.50					
	Deep	.20					
	Very Deep	.20					
Noise:	Yes	.50	etc.	etc.			
	No	.50					
Opponent gone active?	Yes	.10	etc.	etc.			
	No	.90					
How long on current maneuver?	Short	.20	etc.	etc.			
	Middle	.30					
	End	.50					

AUXILIARY SYSTEMS

The above discussion covers the part of the elicited probability approach to generating scenarios. In the system developed by Perceptronics, a sensor system and an Intelligence Analysis System (IAS) were also developed.

The sensor system, shown in Figure 2-2 would enable the opponent model to deploy sensors and get sensor information. In the present application, however, it doesn't seem necessary to go into that level of detail.

The function of the Intelligence Analysis System is to simulate an intelligence expert who knows how the environment and the opponent behave, but must rely upon status reports from the ASW trainee for data about the current status of the fleet. The IAS provides the trainee with an intelligence report (i.e., the probabilities of possible actions by the opponent based upon the information in the trainee's status report).

The IAS is identical to the scenario generator, with two vital exceptions. First, it uses a "current status state vector," $\overline{Z}^t_s$, which is generated from the trainee's status report, instead of the vector which represents the actual state of the environment, $\overline{Z}^t$. This means that the IAS intelligence report will be only as accurate as the trainee's status report. Second, the intelligence report is based upon the aggregated probabilities. Thus, no Monte Carlo selection is made and no transformation operators are invoked. By limiting the kinds of transformation operators used in the scenario generator to those which output a constant output, we insure that the aggregated probabilities of state transformations are the same as the probabilities of the generated states (see Section 2.6). A functional description of the Intelligence Analysis System is provided in Figure 2-3.

DECOY MODEL

<u>The Basic Concept</u>. A decoy is a relatively cheap target which simulates the target of values to the detectors. Decoys are characterized by cost and their ability to simulate the target of vaue to various types of detectors. A convenient number to represent this is the probability of fooling the detector. The opponent would only deploy decoys if he knew he was being hunted because otherwise he would increase the chance of detection.

<u>The Representation</u>.

Type of Detector Type of Decoy	ESM	SONAR	....	DETECTOR N
Expensive	P_{11}	P_{12}	...	P_{1n}
Cheep	P_{21}	P_{22}	...	P_{2n}

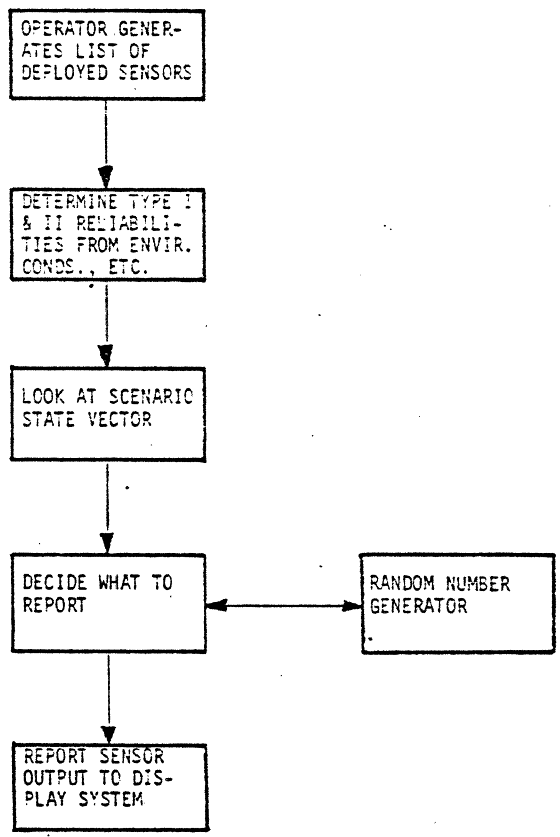


Figure 2-2. Flowchart of Sensor System

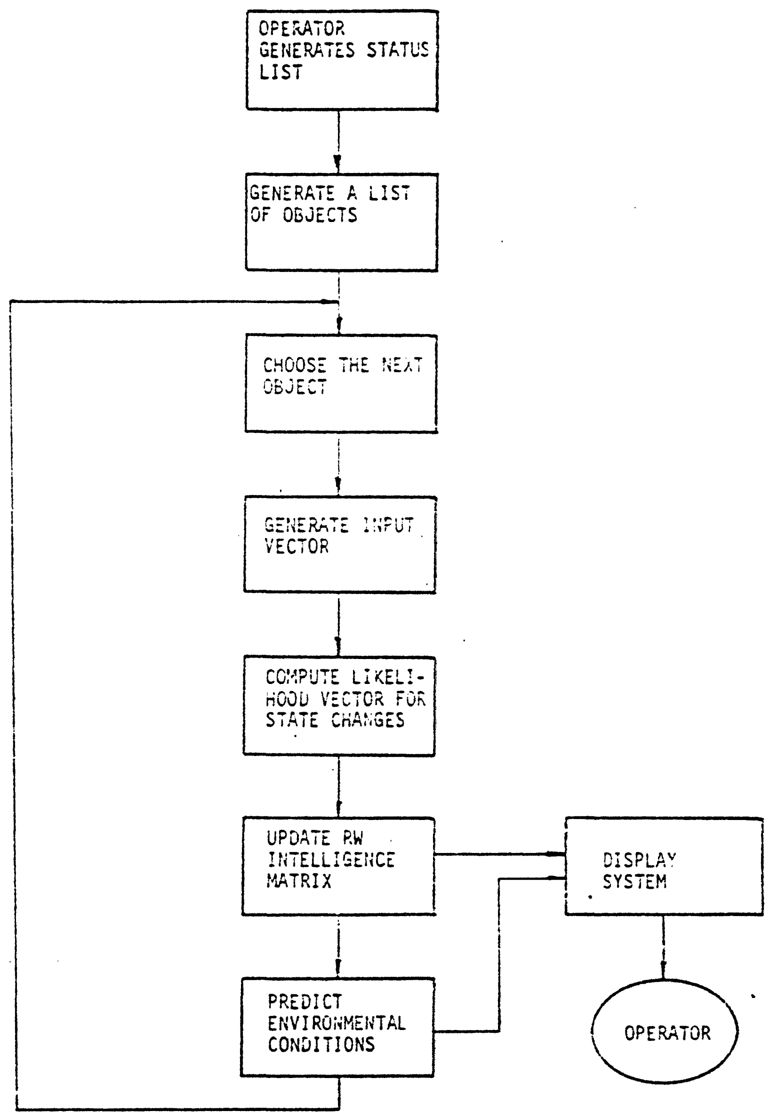


Figure 2-3. Flowchart of Real World Intelligence System (RWIS)

Decoy Deployment Decisions. The opponent must decide what if any decoy to deploy. Once it is decided to deploy a decoy it must be decided what behavior to program for the decoy; this includes:

- trajectory for decoy to follow;
- counter measures for the decoy to use.

SECTION III

THE ADAPTIVE DECISION MODELING APPROACH

INTRODUCTION

The adaptive decision approach to generating knowledgeable opponent behavior -- which uses pattern recognition -- is based on learning opponent behavior from instructor opponent controllers. This approach is based on decision modeling and utility theory. In the present application, all of the relevant information for selecting the opponent's next action is immediately available at the time it's needed.

ADAPTIVE DECISION MODELING

Work on adaptive decision-making is derived from the areas of behavioral decision research and AI experience with learning networks. The unique aspect of this approach is the capability to adjust model parameters on-line and change decision strategy accordingly. In essence, the learning system attempts to identify the decision process of the human operator on-line by (a) successive observation of his actions, and (b) establishment of an interim relationship between the input data set and the output decision (the model). Learning in this context refers to a training process for adjusting model parameters according to a criteria function. The object is to improve model performance as a function of experience, or to match the model characteristics to that of the operator.

Learning techniques have been used to model the decision strategy and to identify the sources of cognitive constraints on the human operator performing a dynamic prediction task (Rouse, 1972). Another example of an adaptive model of the human operator through real time parameter tracing has been reported by Gilstad and Fu (1970). Linear and piecewise-linear discriminant functions were used to classify system gains, errors and error rate. The decision boundaries for classification were determined through a process of on-line learning, observing operator performance and parameter adjustment. The specific model used was applicable only to very limited tasks, and merely illustrated the feasibility of the technique.

A unique advantage of using a learning system lies in its capability to act as a pattern classification mechanism. As such, it can be used to identify biases in operator decision policy as a response to classes or patterns in the input data (Tversky, et al, 1972). In conventional Bayesian technique, the pattern of events is decomposed into elementary data points. With the assumption of independence, the elementary data points are aggregated to revise the hypothesis. Effects of the data pattern do not bear on the decision.

In dynamic decision making, however, the temporal and spatial nature of the data are highly significant. Since decision data appear as a pattern of individual events, it is reasonable to assume that the subject responds to the pattern as well as to the individual values. In fact, the pattern may contain the greater amount of information. Classification of input patterns by the learning mechanism can be accomplished by programmed congnizance of such data features as: data with non-independent events, data with correlated events, data with events which continuously vary with time, the number of elements of decision data and the rate of change in the data points.

THE MAU MODEL

Multi-attribute decision analysis is the most widely used approach for making evaluations involving multiple criteria. MAU methods decompose the complex overall evaluation problem into more manageable sub-problems of scaling, weighting, and combining criteria. In doing so, the MAU methods provide a rich framework for analysis, discussion, and feedback. This "divide and conquer" approach to evaluation involves defining the problem, identifying relevant dimensions of value, scaling and weighting the dimensions, and finally aggregating the dimensions into a single figure of merit for the system.

The power of the multi-attribute approach lies in its level of analysis and flexibility. Sensitivity analyses of the level and weight of each dimension can provide indications of what aspects to concentrate tests on, or what system elements to modify. Flexibility is present, since criteria can be added or deleted as necessary. Also, the weights and levels can be quickly adjusted according to new functional requirements and capabilities.

In the MAU model, the consequences of every action are considered to be decomposable according to a single common set of attributes. The model computes an aggregate multi-attribute utility (MAU) as a weighted sum of each consequence attribute level (A_i) multiplied by the importance or utility of the attribute (W_i). The calculated MAU of each action is used as the selection criterion:

$$MAU_j = \sum_i W_i \ A_{ik}$$

where

MAU_j = the aggregate utility of option j

W_i = the importance weight of attribute i, and

A_{ik} = the level of attribute i for action k.

Figure 3-1 shows the major components of the MAU model in block diagram form. Possible actions are parameterized in terms of attribute levels. The MAU calculator uses as inputs (1) the attribute levels of the given action, and (2) a vector of "attribute weights" which have been dynamically estimated for a given operator by an adaptive model.

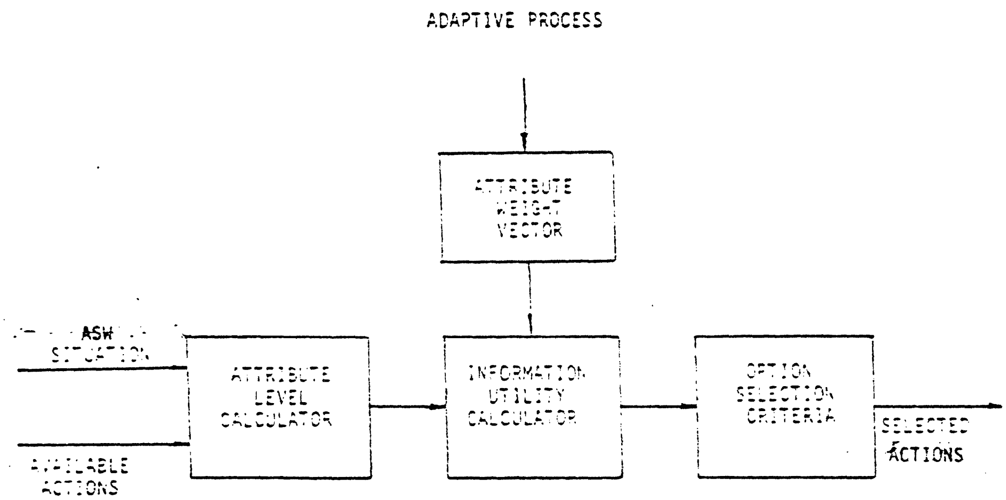


FIGURE 3-1
OVERVIEW OF ACTION SELECTION MODEL

Calculation of the multi-attribute utility for each action is central to the operation of the model. The MAU calculation is shown in Figure 3-2. The dot-product of the attribute level vector and the attribute weight vector provides the aggregate MAU value. The attributes are scaled so that each attribute level ranges from 0 to 1. Further, the orientation is arranged such that each attribute contributes positively to the overall aggregate MAU. That is, holding all other attribute levels constant, an increase in any attribute level increases the MAU.

ATTRIBUTE CHOICE.

The determination of attributes to include in the decision model is probably of greater importance than the accurate assessment of the importance weights (Dawes, 1975). The following list of desirable characteristics for the attributes expands on Raiffa's (1969) recommendations of attribute independence, set completeness, and minimum dimensionality:

a. Accessible. The levels of each factor should be easily and accurately measurable.

b. Conditionally Monotonic. The factor level should be monotonic with the criterion (preference) regardless of the constant values of other factors.

c. Value Independent. The level of one attribute should not depend on the levels of the other attributes. This is to some extent a consequence of recommendation b.

d. Complete. The set of attributes should present the operator's behavior as completely as possible.

e. Meaningful. The attributes should be reliable and should demonstrate construct validity. Feedback based on the model attributes should be understandable to the operator.

For the most part, these recommendations result in an attribute set that is measurable, predictive, and in accord with the axioms of utility theory. The recommendations also imply a limitation on the number of possible attributes. The requirements of independence and meaningfulness render any large set of attributes unrealizable, because of the cognitive limitations of the human operator.

ADVANTAGES OF THE MULTI-ATTRIBUTE UTILITY MODEL.

The multi-attribute information utility model presented here is characterized by several attractive features. These features, itemized below, offer substantial advantage over the EU decision model. The advantages arise out of the theoretical structure of the model, especially its decomposition property, and have all been empirically demonstrated to some degree in ongoing Perceptronics programs, (Samet, Weltman, and Davis, 1976; Steeb, Chen and Freedy, 1977).

$$\begin{array}{cccc} \text{Attribute Level Vector} & \text{Attribute Weight Vector} & = & \text{MAU} \\ \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{bmatrix} & \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \end{bmatrix} & = & \Sigma A_i W_i \end{array}$$

FIGURE 3-2
CALCULATION OF INFORMATION UTILITY

a. Generality. The adaptive, multi-attribute model for information selection holds a considerable amount of generality. It can be applied in situations where diagnostic actions can be decomposed into a small set of manageable, quantifiable attributes which have two critical characteristics. First, they must be logically related to the situation-specific demands. That is, their relevance to specific situations must be known. Second, they must directly impact upon a decision maker's choices among competing options. A number of military decision-making environments have already been demonstrated to fit this paradigm (e.g., Coats and McCourt, 1976; Hayes, 1964; McKendry, Enderwick and Harrison, 1971; Samet, 1975).

b. Parsimony. The model is parsimonious; it need only assess an operator's weights for a limited number of information dimensions or attributes. Besides significantly minimizing the model's computational needs and software complexity, this feature reflects findings of psychological experiments (e.g., Hayes, 1964; Slovic, 1975; Wright, 1974) and is in agreement with contemporary decision theory (e.g., Tversky and Kahneman, 1974), all of which suggest that a decision maker can only perform weighting and aggregation on a relatively small number of the important dimensions in the decision task. Also, when decisions are based on a manageable number of information dimensions, they are easier to communicate and rationalize -- especially in group decision-making situations (Gardiner and Edwards, 1975). In complex situations, therefore, the reduction in the number of model parameters in the proposed MAU model as compared to the expected utility model are of major importance.

c. Robustness. Like other linear composition models, the multi-attribute decision model is robust; that is, its performance is not significantly degraded by small perturbations in the model's parameters (Dawes and Corrigan, 1974). Such robustness probably contributes to the finding that multi-attribute utility assessment techniques have proven, in certain instances, to be more reliable and valid than direct assessment procedures (Newman, 1975; Samet, 1976).

d. Speed of Adaptation. The adaptive model adjusts all parameters with each incorrrectly predicted operator decision (i.e., action selection). Thus, weights for specific troubleshooting attributes can be trained rapidly during sessions in which the operator performs the diagnostic task. This is in contrast to the current model, in which only the parameters of the chosen and predicted actions are adjusted in a given decision.

e. Flexibility. The multi-attribute utility model is inherently flexible. If accurate prediction of troubleshooting behavior is not sufficient (i.e., if attribute weights cannot be trained to stable values), additional features or attributes can be added and inappropriate ones deleted. The response to dynamic changes in conditions is similarly flexible. In instances where conditions change rapidly and radically, new sets of weights trained for the new conditions can be substituted. Such weight vectors could be prepared ahead of time by training them either in actual operational situations or in step-through simulations.

UTILITY ESTIMATOR

The dynamic utility estimation technique is based on a trainable pattern classifier. Figure 3-3 illustrates the mechanism. As the operator performs the task, the on-line utility estimator observes his choice among the available actions at each point in the sequence and views his decision-making as a process of classifying patterns consisting of varying attribute levels. The utility estimator attempts to classify the attribute patterns by means of a linear evaluation (discriminant) function. These classifications are compared with the operator's choices. Whenever they are incorrect, an adaptive, error-correction training algorithm is used to adjust the utilities. A comprehensive discussion of this technique can be found in Freedy, Davis, Steeb, Samet, and Gardiner (1976).

TRAINING ALGORITHM

On each trial, the model uses the previous utility weights (W_i) for each attribute (i) to compute the multi-attribute utilities (MAU_k) for each action (k). Thus,

$$MAU_k = \sum_{i=1} W_i \; A_{ik} \tag{3-1}$$

where

W_i is the weight of the attribute, and

A_{ik} is the level of the i^{th} attribute associated with action k.

The model predicts that the operator will always prefer the action with the maximum MAU value. If the prediction is correct (i.e., the operator chooses the action with the highest MAU), no adjustments are made to the utility weights. However, if the operator chooses an action haveing a lower MAU value the model adjusts the utility weights by paring the chosen action with the predicted action and applying the error correction training algorithm. In this manner, the utility estimator "tracks" the operator's diagnostic strategy and learns his utilities or weights for information attributes. The training rule used to adjust the weights associated with each of the attributes is illustrated in Figure 3-3.

Actual in-task estimation appears feasible using pattern recognition techniques. Instead of batch processing, the pattern recognition methods refine the model decision-by-decision. Briefly, the technique considers the decision maker to respond to the characteristics of the various alternatives as patterns, classifying them according to preference. A linear discriminant function is

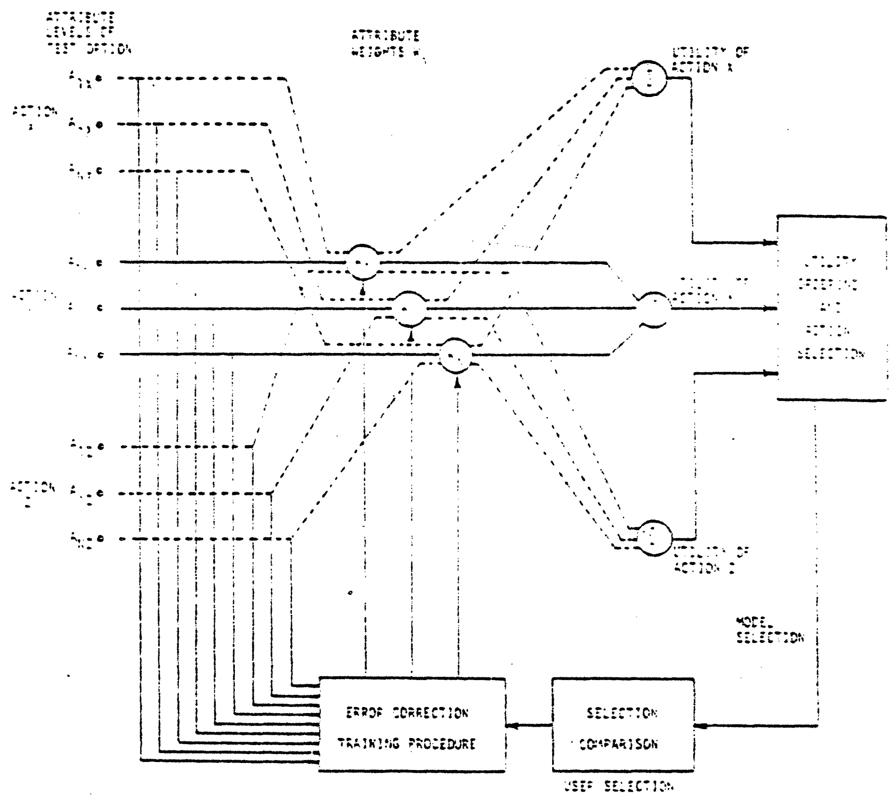


FIGURE 3-3
SCHEMATIC REPRESENTATION OF ADAPTIVE
MULTI-ATTRIBUTE INFORMATION UTILITY MODEL

used to predict this ordinal response behavior, and when amiss, is adjusted using error correcting procedures. This use of pattern recognition as a method for estimation of decision model parameters was apparently first suggested by Slagle, (1971). He made the key observation that the process of expected utility maximization involved a linear evaluation function that could be learned from a person's choices.

The suggested technique was soon applied by Freedy, Weisbrod, and Weltman (1973) to the modeling of decision behavior in a simulated intelligence gathering context. Freedy and his associates assumed the decision maker to maximize expected utility on each decision. They assigned a distinct utility, $U(x_{jk})$, to each possible combination of action and outcome, as shown in the decision tree in Figure 3-4. The probabilities of occurrence of each outcome j given each action k were determined using Bayesian techniques. These patterns of probability were used as inputs to the estimation program (Figure 3-5). The expected utility of each action A_k was then calculated by forming the dot product of the input probability vector and the respective utility vector. This operation is equivalent to the expected utility calculation:

$$EU(A_k) = \sum_j P(x_{jk}) \cdot U(x_{jk}) \qquad (3\text{-}2)$$

The classification weight vector W_{jk} in the pattern recognition program acts as the utility $U(x_{jk})$. The alternative A_k having the maximum expected utility is selected by the model and compared with the decision maker's choice. If a discrepancy is observed an adjustment is made, as shown in Figure 3-4. The adjustment moves the utility vectors of the chose - and predicted actions (W_c and W_p, respectively) in the direction minimizing the prediction error. The adjustment consists of the following:

$$W_c' = W_c - d \cdot P_p \qquad (3\text{-}3)$$

$$W_p' = W_p + d \cdot P_c \qquad (3\text{-}4)$$

where

W_c' is the new vector of weights $[W(x_{1c}), W(X_{2c})]$ for action c

W_c is the previous weight vector for action c

d is the correction increment

P_i is the probability vector describing the distribution of outcomes

$[P_{1k}, P_{2k}, \ldots P_{nk}]$ resulting from action k

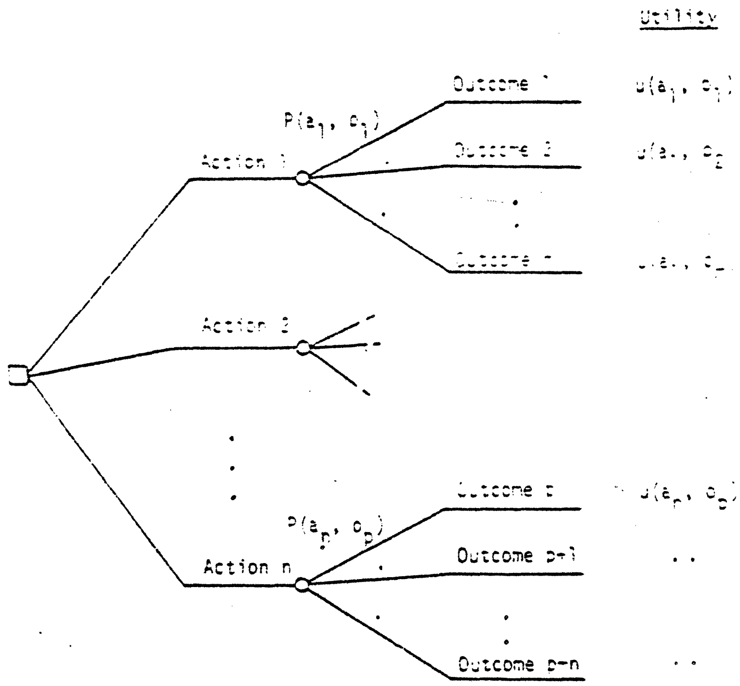


FIGURE 3-4. DECISION TREE OF UTILITY ESTIMATION PROGRAM DEVELOPED BY FREEDY ET. AL.

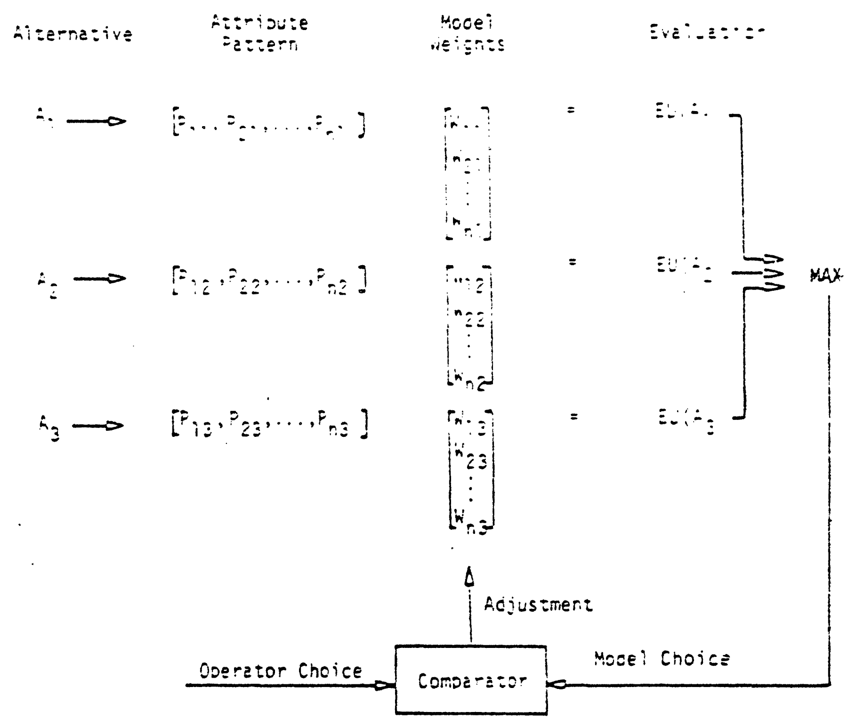


FIGURE 3-5 STRUCTURE OF UTILITY ESTIMATION PROGRAM OF FREEDY ET. AL.(1973)

This model is an adaptation of the R-category linear machine (Nilsson, 1965). The pattern classifier receives patterns of descriptive data (outcome probabilities) and responds with a decision to classify each of the patterns in one of R categories (actions). The classification is made on the basis of R linear discriminant functions, each of which corresponds to one of the R categories. The discriminant functions are of the form:

$$g_i(x) = W_i \cdot x \text{ for } i=1, 2, \ldots, R \tag{3-5}$$

where x is the pattern vector and W_i is the weight vector. The pattern classifier computes the value of each discriminant function and selects the category i such that

$$g_i(x) > g_j(x) \tag{3-6}$$

for all j=1, 2, ..., R; i≠j

A geometric interpretation of the R-category linear machine is shown in Figure 3-6 (Nilsson, 1965). Decisions involving two possible consequences, x_1 and x_2, are evaluated according to three discriminant functions $G_1(\underline{x})$, $G_2(\underline{x})$, and $G_3(\underline{x})$. The lines of intersection between the discriminant hyperplanes are the points of indifference between actions. Mappings of these lines of intersection to the attribute plane are shown in the figure. The resulting regions R_1, R_2, and R_3 correspond to the actions maximizing the (expected utility) evaluation function.

The R-category technique becomes somewhat cumbersome if a large number of actions are possible or if the decision circumstances change rapidly. This problem is a result of the assignment of a distinct, holistic utility to each tip of the decision tree. The number of model parameters thus increases rapidly with an increase in the number of actions possible. Also, the only weight vectors adjusted in a given decision are those corresponding to the model-predicted and the actually chosen actions. This partial adjustment makes the system somewhat unresponsive to change.

A natural extension of Freedy's approach is to adapt the single discriminant, multi-attribute approach to the modeling of objective choice behavior. Each possible outcome of a decision can be associated with a set of attributes or objectives of the decision maker. An importance weight vector defined over the various attributes can then be adjusted to predict behavior. The mechanism is simply that of a threshold logic unit. The adjustment rule following an incorrect prediction is

$$W' = W + d(x_c - x_p) \tag{3-7}$$

where

W' is the updated weighting vector

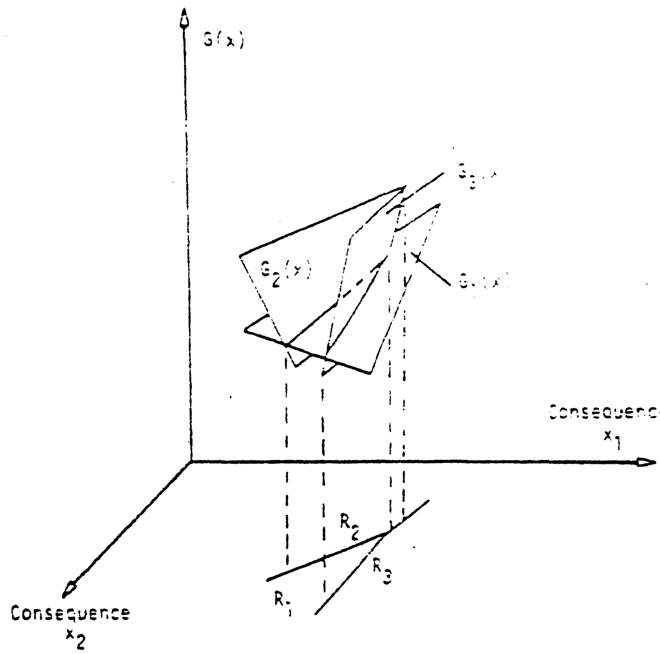


FIGURE 3-6 GEOMETRIC INTERPRETATION OF R-CATEGORY LINEAR MACHINE
(ADAPTED FROM NILSSON, 1965)

W is the previous weighting vector

x_p is the attribute pattern of the model-predicted choice

x_c is the attribute pattern of the decision maker's choice

d is the adjustment factor

A possible advantage of the pattern recognition technique over many of the other forms of estimation is its flexibility of adjustment. Several types of error correction are possible for the TLU, each with a different combination of speed, stability, and complexity. The three principle forms are the fixed increment rule, the obsolute correction rule, and the fractional correction rule. These differ solely in their formulation of the adjustment factor d in Equation 3-7.

The fixed increment rule simply assigns a non-zero constant to d. Thus the movement of the weight vector is a constant proportion of the difference in the predicted and chosen patterns. The correction may not be sufficient to avoid subsequent errors with the same pattern, but the process is eventually convergent (Duda and Hart, 1973). The fixed increment rule has the advantages of simplicity and relative insensitivity to inconsistent behavior.

A more rapid but also more potentially unstable rule is the absolute correction rule. This method sets d to be the smallest integer at which the error of the pattern is corrected. In the decision modeling situation, this becomes:

$$d \text{ - smallest interger} > \frac{|k \cdot (x_c - x_p)|}{(x_c - x_p) \cdot (x_c - x_p)} \tag{3-8}$$

in which

x_c is the attribute level vector of the operator selected choice

x_p is the attribute vector of the predicted choice

The fractional correction rule is similar to the abosolute rule but is typically less extreme. The franctional rule moves the weight point some fraction of the above distance:

$$d = \frac{\lambda |k \cdot (x_c - x_p)|}{(x_c - x_p)(x_c - x_p)} \tag{3-9}$$

where λ is a constant $0 < \lambda < 2$.

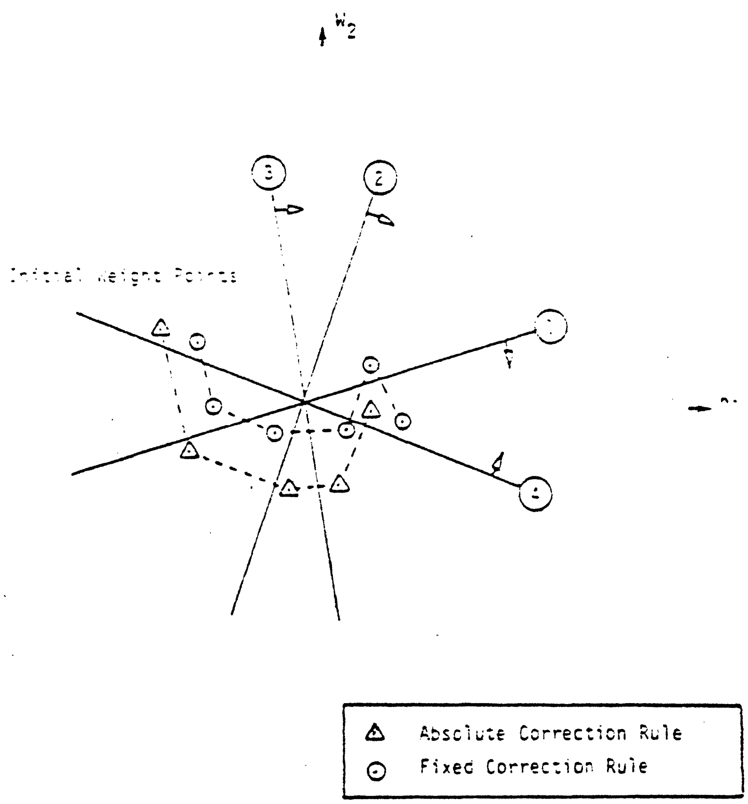


FIGURE 3-7 COMPARISON OF BEHAVIOR OF CONVERGENCE RULES

All three of the adjustment rules have been proven convergent with linearly separable patterns (Nilsson, 1965). The speed of convergence is normally fastest with the absolute rule. This is illustrated for an example series of adjustments in Figure 3-7. The set of four numbered lines in the figure are a sequence of patterns. These patterns are shown as hyperplanes in a 2-dimensional weight space. Each hyperplane represents the difference between two multi-attribute vectors. The operator choice is shown by the direction of the arrow at each pattern. The absolute rule, (the triangles in the figure) is seen to achieve correct prediction after four observations, while the fixed rule (the circles) requires five. Unfortunately, the absolute rule is expected to be less forgiving of inconsistent behavior than the fixed or fractional rules. This is because of the large responses the absolute rule makes to operator inconsistencies. The fixed and fractional rules may exhibit a greater tendency to smooth or average the behavior.

SECTION IV

THE HEURISTIC SEARCH APPROACH

STATE SPACE MODEL

The overall objective of knowledgeable opponent scenario generation is to provide a realistic simulation of an active enemy. The enemy would react to events and actions taken by the friendly forces and choose a course of action that would lead to the achievement of some enemy goal, which usually means a bad outcome for the friendly forces. The heuristic search approach provides such a mechanism.

In the underlying model, which is called the "state space" model, the problem domain (such as underwater warfare) is expressed in terms of "states", which are complete descriptions of the tactical situations as they exist at some particular instant of time (Nilsson, 1971). An "action" is a transformation which, when applicable, converts one state into another. Thus, a sequence of actions ("plan" or "allocation") converts some initial state into a final, or goal, state. The enemy submarine commander asks the question, "What sequence of actions can transform the current state into a goal state which satisfies my overall objectives?" In other words, "How do I get from where I am to where I want to go?" Before a system can perform properly, it must know what actions are available, under what circumstances they can be applied, what their effects are, and what possible states can arise from their use.

BASIC SEARCH TECHNIQUES

The most basic search techniques are systematic expansions of the state space. Starting from the start node (labeled 1 - the current state), the search algorithm expands all its possible successive nodes. When a goal node is encountered, the path from the initial node to that goal node is the solution sought. In the ASW case, it is the strategy, or sequence of actions, the commander has to take to reach his objective.

Figure 4-1 and Figure 4-2 show the most elementary algorithms - the "breadth-first" and the "depth-first" algorithms, respectively. In the "breadth-first" algorithm, each node is expanded completely - all its "sons" identified - before the next one is expanded, and a given layer of nodes is expanded before the next is started. This method is guaranteed to find the shortest path from the start to the goal nodes. The numbers in Figure 4-1 indicate the order of node expansion.

In the "depth-first" algorithm, each alternative line of inquiry is sought to the fullest depth before other alternatives are evaluated. When such a search fails, the algorithm tries the next deepest possibility. Figure 4-2 shows the order of node expansion in this algorithm. The depth first algorithm does not guarantee the shortest path to a goal if more than one goal node exists.

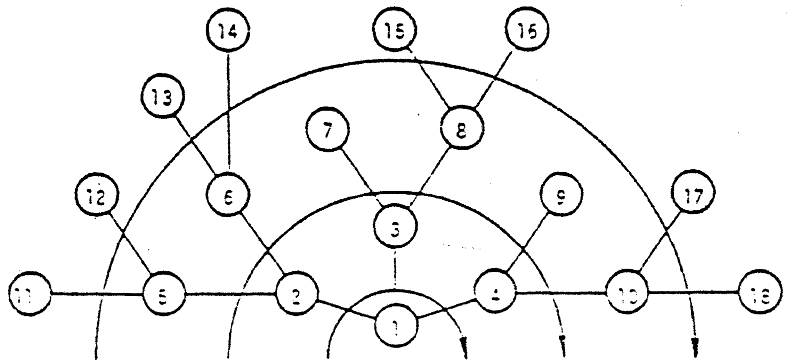


FIGURE 4-1 BREADTH-FIRST EXPANSION ORDER

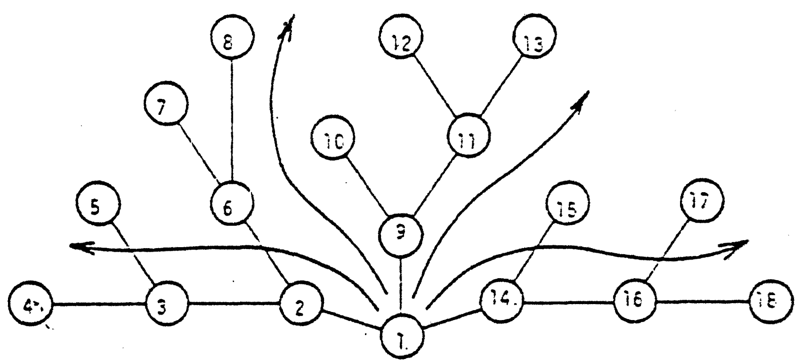


FIGURE 4-2 DEPTH-FIRST EXPANSION ORDER

These search methods are "blind" methods because they develop systematically every node in the state space without using any information which may be known in advance about the particular problem domain or the particular knowledge found in the nodes that has already been expanded to guide the search process. The heuristic search approach is the class of algorithms that uses such domain specific knowledge to guide the search.

HEURISTIC SEARCH METHODS

Heuristic search methods try to utilize any information known about the problem domain to guide the search for a solution in the state space. The added information helps avoid the combinatorial explosion of computer resources (time and memory) needed for the basic search techniques. Figure 4-3 illustrates the basic idea of the heuristic search approach by comparing it to depth first and breadth first searches. The contours of node expansion are directed toward the goals G1 and G2, in contrast to the blind search algorithm. Applying a heuristic search usually leads to the discovery of optimal or suboptimal solutions in cases that would be too big to handle by standard techniques. Many achievements of heuristic search are known. For example,

a. Computer Aided Design (Powers, 1973; Hagendorf et al, 1975)

b. Test Sequence Generation for Detection of Failures in Clockmode Sequential Circuits (Hill and Huey), 1977

c. Edge and Contour Detection (Martelli, 1976)

d. Chromosome Matching (Montanari, 1970)

e. Organic Chemical Synthesis (Sridharan, 1973)

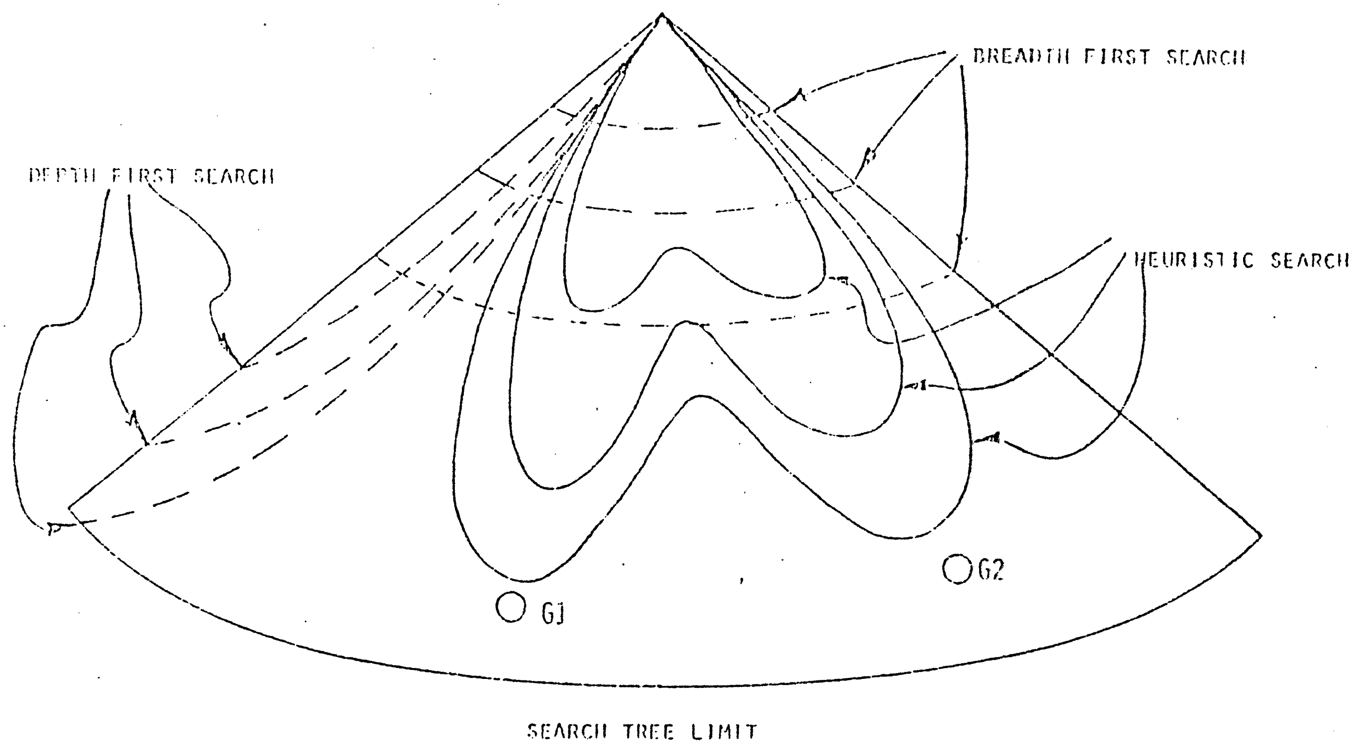
f. Ballistic Missile Defense (Leal, 1977)

g. Discovery of Mathematical Concepts (Lenat, 1978)

The heuristic information can be contained in different parts of the search algorithm. If Γ is the function that generates node successors and $f(\eta)$ is an estimate of the promise of node η to be on the path to a goal node, then the heuristic information may be contained in either of them. Using knowledge in Γ, the search algorithm would generate first the more probable successors of a node. On the other hand, using knowledge in $f(\eta)$, the most promising nodes would be selected for subsequent development in the face of less promising ones.

THE MINIMAX AND $\alpha\beta$ ALGORITHMS

Two algorithms which have particular applicability to the case of military confrontation are the minimax and the $\alpha\beta$ algorithms. The minimax is applicable in zero-sum adversary confrontations where what is good for one side is bad for



G1, G2 GOAL NODES

FIGURE 4-3: EXPANSION CONTOURS OF DEPTH FIRST, BREADTH FIRST, AND HEURISTIC SEARCH METHODS.

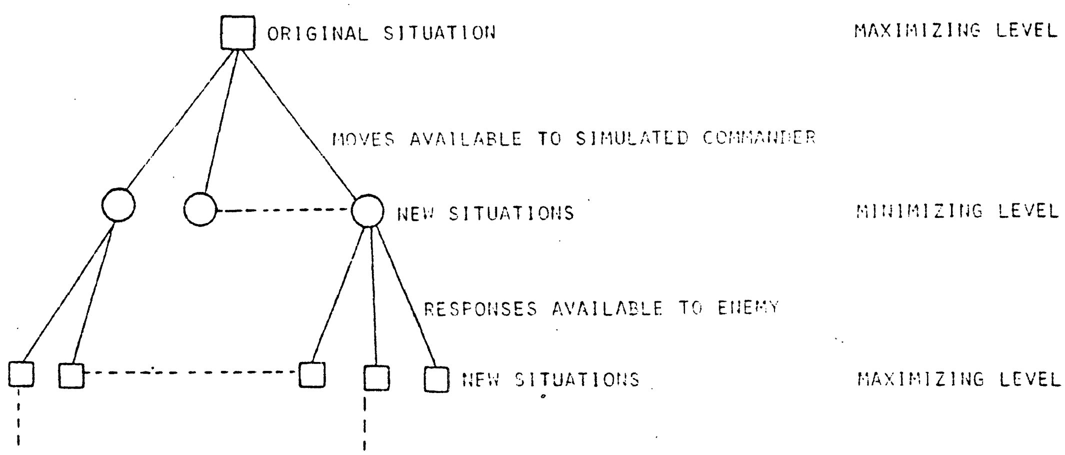


FIGURE 4-4: THE MINIMAX ALGORITHM TREE DEVELOPMENT

the other. When developing the state space of such a problem, the prudent decision maker has to assume that, when given the choice, the enemy would select the alternative which is the most damaging to the decision maker's own objectives. When expanding the search space for this problem, as shown in Figure 4-5, the commander first determines all the alternatives available to him. This is the maximinzing level because at this level the commander has the choice, and he will obviously choose the alternative that maximizes his objectives. The next level is the set of responses available to the enemy for each of the commander's choices. Here the enemy will make the choice, and he will choose the worst alternative (from the commander's point of view). Thus, this layer is called the minimizing level. The maximinzing and minimizing of layers goes on until the allocated computing resources are used up. At that point, the static value of each tip node (i.e., "worth" of the situation) is evaluated and the choices are made at each decision point. The "backed-up" values propagate upward in the state space tree until they reach the first layer. These values are the basis of the commander's choice among the alternative actions available to him. This "minimazing" algorithm is repeated for every decision the simulated commander has to make; thus, it takes into account the dynamics of the situation, and it finds the best tactical move foreseeing the best choice of the enemy. In this algorithm, the heuristic information is contained in the tip node evaluation function $f(\eta)$ in the previous section.

The alpha-beta algorithm is an improved version of the basic minimax algorithm. It uses a common sense argument to prune the tree that has to be developed. It can be shown that although the algorithm allows a large part of the search tree to be completely ignored, it will not lose any solution that the basic minimax algorithm would find.

The alpha-beta algorithm starts with a depth-first search down to some level η (see Figure 4-5). When the depth limit is reached, the nodes are evaluated and temporary values are backed-up in the tree. The alpha-beta technique takes advantage of these preliminary values. Consider, in Figure 4-5, the maximizing node A in the tree after nodes 4-9 have been developed below it. A has been assigned a temporary value of 0.2 (propagated from node 5). B, which is a minimizing node, has been assigned a temporary value of 0.1 (propagated from node 9).

At this time, there is no point developing any other successor to the node b (such as C) because, since it is a minimizing node, the best value B can get is 0.1 or lower, and node A, being a maximizing node, will always select 0.2 over 0.1. This argument is the "alpha" half of the alpha-beta pruning. The dashed lines in Figure 4-5 show all the subtrees that will be pruned off and the order of node generation.

The "beta" half operates in precisely the reverse for nodes in the minimum layers. By using the alpha-beta algorithm, the tree can be explored approximately twice as deep as a simple minimax algorith, while expanding the

EXAMPLE OF ALPHA-BETA PRUNING

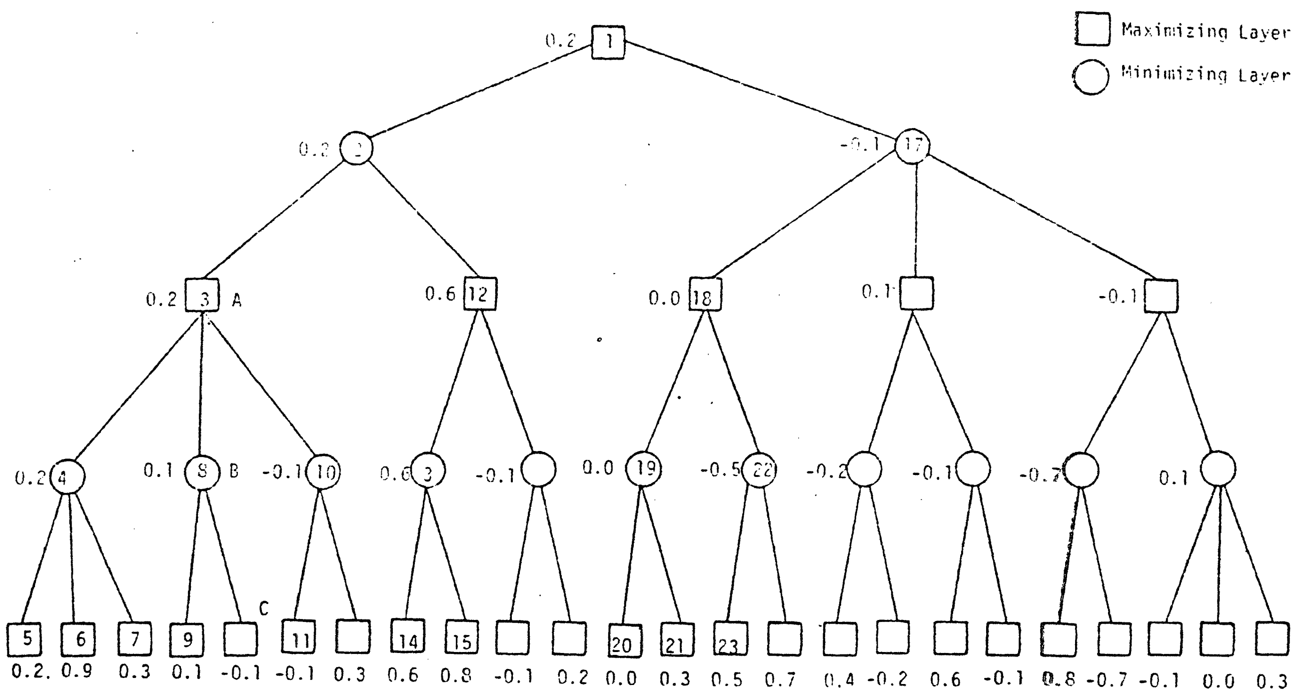


FIGURE 4-5: EXAMPLE OF ALPHA-BETA PRUNING

same number of nodes. The algorithm is somewhat slower, inasmuch as it has to do the bookkeeping for the temporary alpha and beta values. The alpha-beta algorithm is a very promising potential opponent model.

ADVANTAGES

a. Heuristic search techniques have a wide range applicability, as can be seen from the examples mentioned above.

b. The underlying structure (state-space, AND/OR graphs) is very general and fits naturally all problems of a combinational nature and all hierarchical problems which can be decomposed into goals and subgoals (this includes decision trees).

c. General theoretical results are available.

d. It is universally accepted that heuristics are crucial to cope with intractable problems.

SCOPE AND LIMITATIONS

a. Heuristic search techniques are designed for problems of a particular nature only, with well-defined states, subgoals or subproblems. Problems with a continuous nature, for instance planning in a continuum, cannot be solved via heuristic search.

b. The use of heuristic search poses itself a problem. The more specific a heuristic function, the more efficient it is in guiding the search. How well designed and problem-specific heuristics are will therefore determine their efficiency.

c. Heuristic search might be subject to catastrophes (if no solution is found after the computational resources are exhausted or an insufficiently good solution is found).

SECTION V.

PRODUCTION RULES APPROACH

OVERVIEW

Production rule systems represent another successful approach for knowledge representation and deductive mechanisms. This approach is similar to the heuristic search approach in that it uses a modification of the state space model as the underlying conceptualization. The technique of representing the knowledge is different, however, and so is the mechanism which finds the path from the current state to the goal state. The problem specific knowledge (heuristics) is packaged in production-rule systems as small modular "chunks" called productions.

A production is a rule which consists of a situation recognition part and an action part. Thus a production is a "situation - action" pair in which the left side is a list of things to watch for in the description of the current state of the world, and the right side is the list of things to do in that case.

In the case of submarine warfare, a production that guides the commander's actions may be something like:

If

AND

Enemy dominates area

Enemy has not yet detected you

You are out of his torpedo range

You are in very shallow water

Then

Escape by sinking to bottom in silence

The effect of such a production is to respond to the situation when all the aspects combined by the AND are present and change the current action from whatever it was before to ESCAPE.

In addition to the large set of such productions, the production rule system contains a triggering mechanism that uniformly checks all the productions that apply in a given situation (by testing for truth of the left hand side of each production) and applies those that are applicable - causing the situation to change.

The main advantages of the production rule approach are the ease and modularity of the knowledge representation. Consequently, it is easy to elicit information from experts without requiring that they be programmers. The information is incremental; thus it is easily modified, updated and expanded into new areas of expertise. It is also usually argued by production rule proponents that this form of knowledge representation is highly compatible with human congnition, making it a very useful and powerful training tool. For example, suppose an opponent commander model is built as a production rule system. It becomes very easy to communicate with the system and ask "why have you done that?" meaning what aspects of the situation or what actions of the trainee caused some unexpected response of the enemy commander.

The trainee can discover specifically where he went wrong, and he can start in the middle and try other alternatives. At the same time, this is also a powerful debugging tool allowing experts to tune the system by following its reasoning process and identifying the specific cause for a mistaken conclusion which led to an unreasonable response.

THE SYSTEM

PRODUCTIONS

As AND/OR graphs, production systems are composed of two parts: the set of productions and a mechanism to find a solution in a given situation. We will discuss first a graphic representation of the productions themselves. A simple production specifies a single conclusion which follows from the simultaneous satisfaction of the situation recognition conditions. Any particular conclusion may spring from any production. The conclusion specified in a production follows from the AND or "*conjuction*" of the facts specified in the premise recognition part. A conclusion reached by more than one production is said to be the OR or "*disjuction*" of those productions. Depicting these relationships graphically produces an AND/OR graph with directed edges. Figure 5-1 shows an AND/OR graph which reaches from base tactical facts (F_i) at the bottom, through the different productions (P_i), to a conclusion or an act to be taken at the top. Any collection of productions implies such a graph. In Figure 5-1 we used the set of submarine warfare productions given in Figure 5-2. These productions should be taken as an example of the capabilities of this approach.

The arrangement of nodes in this graph focuses on how the conclusion can be reached by various combinations of basic facts. As with ordinary AND/OR trees, a conclusion is verified if it is possible to connect it with basic facts through a set of satisfied AND/OR nodes. Different sets of facts can be used to reach a given conclusion by selecting different branches at OR nodes.

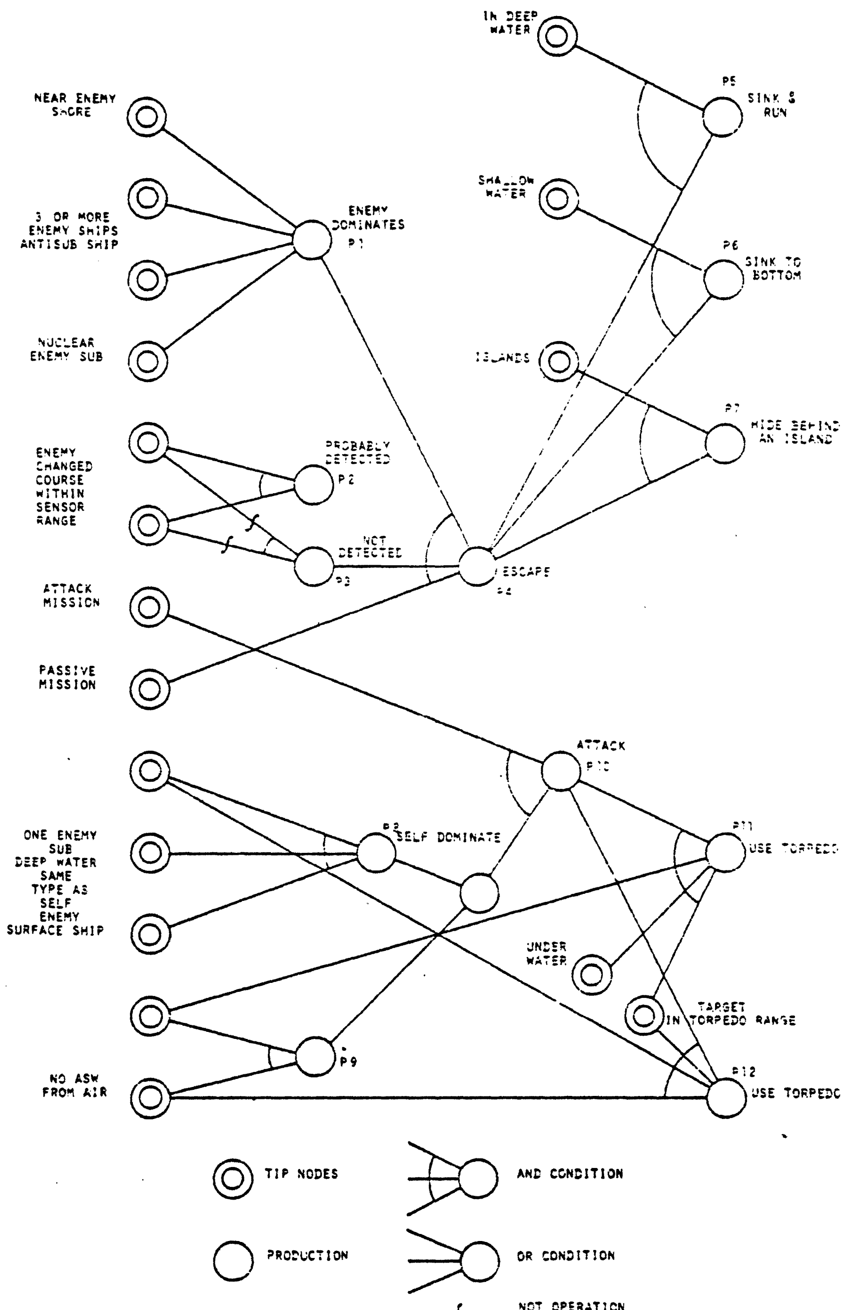


FIGURE 5-1 AND/OR GRAPH

P4

IF

AND

SELF on passive mission

SELF not detected

Enemy dominates area

THEN

ESCAPE

P5

IF

AND

ESCAPE

SELF in deep water

THEN

Sink deep and run

P6

IF

AND

ESCAPE

Self in shallow water

THEN

Sink to bottom in silence

Figure 5-2 (Continued)

P1

IF

OR

Location near enemy shore

3 or more enemy ships in the area

Anti sub ship in area

Nuclear enemy sub

THEN

ENEMY DOMINATES SCENE

P2, P3

IF

AND

SELF WITHIN SENSOR RANGE

ENEMY CHANGED COURSE

THEN

SELF DETECTED

ELSE

NOT DETECTED

Figure 5-2 Production Rule Example

P7

IF

AND

ESCAPE

SELF in Islandic area

THEN

Hide behind an island

P8, P9

IF

OR

AND

One enemy sub in area

Self in deep water

Enemy sub of same type

AND

Enemy surface ship alone

NO ASW in air

THEN

SELF dominate

Figure 5-2 (Continued)

P10

IF

AND

Attack mission

Self dominate

THEN

Attack

P11

IF

AND

Attack

Enemy surface ship alone

Self under water

Target in torpedo range

THEN

Use torpedo

Figure 5-2 (Continued)

P12

IF

AND

Attack

One enemy sub

Target in torpedo range

NO ASW in air

THEN

Use torpedo

Figure 5-2 (Continued)

Sometimes it is useful to look at the implied graph to get a better feel for the problem space, noting whether the reasoning is likely to be broad and shallow, narrow and deep, or broad and deep. Again, however, caution is in order. When used prominently in discussions of goals and subgoals, and/or graph representations tend to make control look like a search problem with the various search ideas becoming applicable. This position has its good and bad features. One bad feature is that it can create a tendency to waste time with an existing problem space rather than to make a better space, where less search, if any, would be needed.

THE CONTROL MECHANISM.

The control mechanism which utilizes the set of productions takes a collection of known facts about the situation and makes new conclusions according to productions that are satisfied by the initial facts. In operation, the user would first gather up all facts available and present them to the system. The control mechanism will then scan the production list for a production which has a matching situation part, i.e., all the premises in the left hand side are satisfied. This production will be activated and its action side will change the facts known about the situation. In the example given, if P1 was activated, it adds the conclusion that the "enemy dominates the area" to the situation description.

Reasoning from base facts to a conclusion rarely entails using only a single step, however. More often, intermediate facts are generated and used, making the reasoning process more complicated and powerful. One consequence is that the individual productions involved can be small, easily understood, easily used, and easily created. Also notice that the intermediate facts added by the lower level productions are tactical facts meaningful to the military users of the system, resulting in many benefits. Using this approach, a simulated submarine commander can produce a chain of conclusions leading to intelligent tactical actions, even as a trainee commander makes his actions dynamically.

In the event many productions have premise or situation specifications that are satisfied simultaneously, there must be some way of deciding among them. Here are some of the popular methods:

a. All productions are arranged in one long list. The first matching production is the one used. The others are ignored.

b. The matching production with the toughest requirements is the one used, where "toughest" means the longest list of constraining premise or situation elements.

c. The matching production most recently used is used again.

d. Some aspects of the total situation are considered more important. Productions matching high priority situation elements are privileged.

So far, the deduction oriented production system is assumed to work from known facts to new, deduced facts. Running this way, a system exhibits forward chaining. But backward chaining is also possible, for the production system user can hypothesize a conclusion or a desired final state and use the productions to work backward toward an enumeration of the facts that would support the hypothesis. For example, (see Figure 5-1) in the case of a submarine commander, the system can start from the mission, e.g., attack enemy sub. Then chaining backward from P10, it will conclude that it has to achieve self-dominance. This can be achieved by confronting an enemy surface ship (P9) or an enemy sub of the same type in deep water (P8). Thus, by a small change of orientation, the same set of productions was used backwards. Knowing that a deduction-oriented production system can run forward or backward, which is better? The question is decided by the purpose of the reasoning and by the shape of the problem space. Certainly, if the goal is to discover all that can be deduced from a given set of facts, then the production system must run forward, On the other hand, if the purpose is to verify or deny a particular conclusion, or reach a desired situation throught a sequence of actions, then the production system is probably best run backward from that conclusion. Avoiding needless fact accumulation is one reason. Indeed, no irrelevant facts need be checked at all. The production system can run backward from all premise elements as long as suitable productions exist. Using sensory systems to supply facts is necessary only when no productions apply.

Deciding whether forward chaining or backward chaining is better depends, in part, on the shape of the space. Figure 5-3 illustrates this by way of two symmetric situations. All possible states are represented along with the operations that can change one state into a neighbor. In the first situation shown, forward chaining is better because there is a general fan in from the typical initial states toward the typical goal states. It is hard to get into a dead end. In the second situation, the shape favors backward chaining since there is fan out.

ADVANTAGES

Proponents of production rule systems usually site one or more of the following advantages:

a. Production systems provide a powerful model of the basic human problem solving mechanisms. This results in easy expert elicitation user communication at the comfortable level of military tactical concepts and terms, easy trouble-shooting, and good training capability.

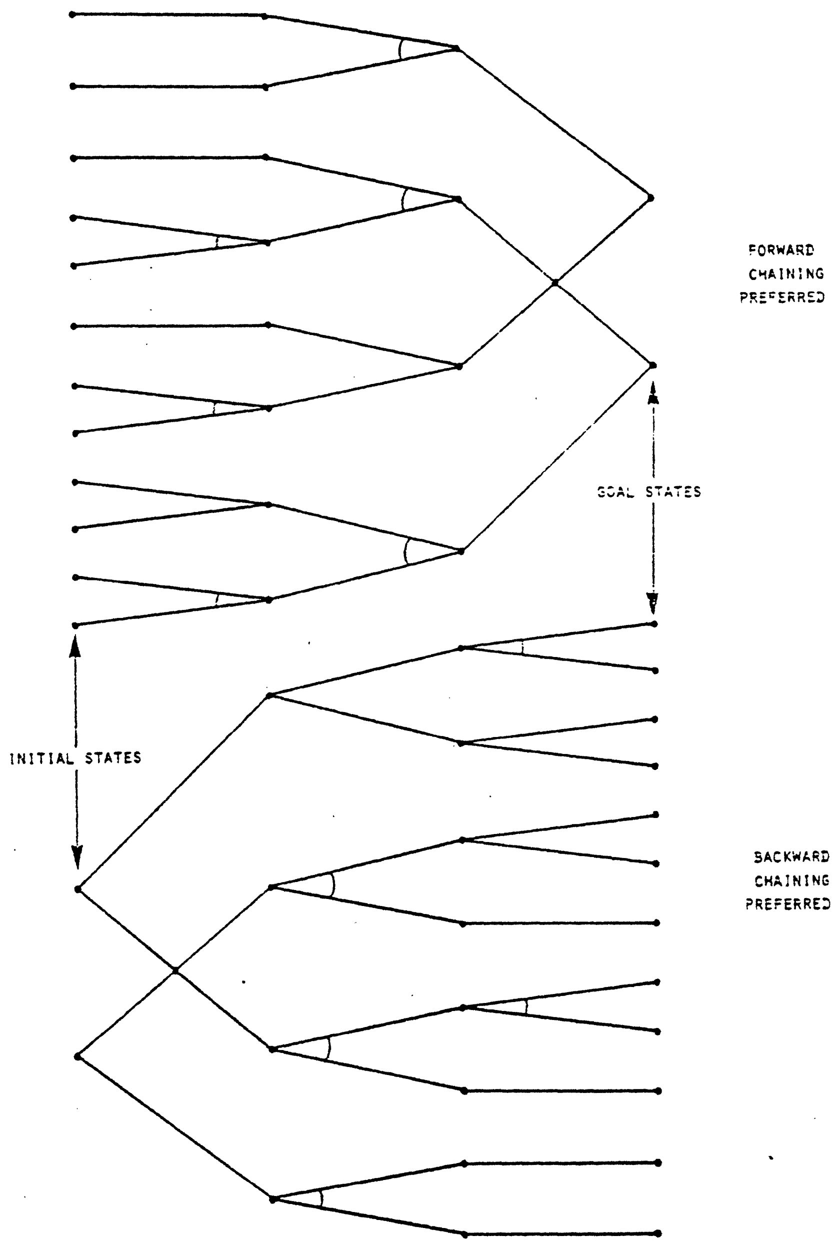


FIGURE 5-3 AND/OR GRAPH SHAPES FOR FORWARD OR BACKWARD CHAINING

b. System states are meaningful to users, debuggers, etc., thus an evaluation can be made on the tactical level rather than in the computer implementation level.

c. Production systems enforce a homogeneous representation of knowledge, effectively separating the static data representation from the uniformly applied evaluation mechanism.

d. The control mechanism is simple and explicit on what to do next, is clear from the current state what productions are available.

e. Production systems allow incremental growth through the addition of individual productions and without changes necessary to any others.

f. Production systems allow unplanned, but useful, interactions which are not possible with control structures in which all procedure interactions are determined beforehand. A piece of knowledge, or a combination of such, can be applied whenever appropriate, not just whenever a programmer predicts it can be appropriate. This can lead to highly intelligent performance by systems with a surprisingly small (several hundreds) set of productions.

g. Providing explanation capability to the system is natural to implement. When some decision is made, the system can present the sequence of productions that led to that conclusion, thus affording its "reasoning" about the situation.

h. The production rule approach is as general as any other method based on the state space model.

i. Productions can be quantified with probability information leading to applicability in decision making and risk evaluation.

DISADVANTAGES

Some of the advantages of the production rule approach can become disadvantages if care is not exercized in the design process:

a. Maintaining focus of attention: It would seem that PR systems allow knowledge to be tossed into the system homogeneously and incrementally without worry about relating new knowledge quanta to old. Thus, by relinguishing control, such systems allow unimportant productions to usurp center stage from more important productions, leading the process astray.

b. Size problems: One particular problem is that production systems may break down if the amount of knowledge is too large, for then the number of productions grows beyond reasonable bounds. The advantage of not needing to worry about the interactions among the productions can become the disadvantage of not being able to influence the interactions among the larger number of productions.

The possible solution, of course, is to partition the facts and the productions into sybsystems such that at any time only a manageable number are under consideration. Within each subsystem, some productions may be devoted to arranging transfer of information or attention to another subsystem. Curiously, some users of Hewitt's ACTORS language produce programs that have a strong resemblance to systems of communicating production subsystems.

This solution, however, goes against one of the main advantages of production rule systems, namely, modularity and independent control. If control guiding productions are added, we again have the problem of explicitly directing where control should go.

c. Global Effects: It is awkward to represent global effects using PR approach. Here, again, the modularity of the productions requires that if some global effects (such as weather in ASW) takes part in many productions, it is necessary to duplicate the whole set of productions which behave differently for each different weather state.

REFERENCES

Bond, N.A. and Rigney, J.W. "Bayesian Aspects of Troubleshooting Behavior". Human Factors, 1966, 8, 377-383.

Buchanan, B.G., and Smith, D.H. Computer assisted chemical reasoning. Proc. III international conference on computers in chemical research, education and technology, 1976.

Coates, E.N. and McCourt, A.W. A Questionnaire-Based Analysis of Order-of-Battle Elements. U.S. Army Research Institute for the Behavioral and Social Sciences (Arlington, VA) Technical Paper 271, January 1976.

Dawes, R.M. Shallow Psychology. Paper presented at the 11th Annual Carnegie Symposium on Cognition, Carnegie-Mellon University, April 24-26, 1975.

Dawes, R.M. and Corrigan, B. Linear Models in Decision Making. Psychological Bulletin, 1974, 81:95-106.

Duda, R.O. and Hart, P.E. Pattern Classification and Scene Analysis, Wiley-Interscience: New York, 1973.

Duda, R.O.; Hart, P.E.; Nilsson, N.J.; and Sutherland, G.L. Semantic network representations in rule-based inference systems, to appear in Proc. workshop on pattern-directed inference systems, 1977.

Edwards, W.D. "Dynamic Decision Theory and Probabilistic Information Processing". Human Factors, 4:59-73, 1962.

Freedy, A., Davis, K.B., Steeb, R., Samet, M.G. and Gardiner, P.C. Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation, and Application. Perceptronics, Inc. (Woodland Hills, CA) Final Technical Report PFTR-1016-76-8/30, Contract N00014-73-C-0286/NR 196-128, ARPA Order No. 2347, Code 455, Office of Naval Research, August 1976.

Freedy, A., Weisbrod, R., and Weltman, G. Self-Optimization of Task Allocation in Shared Man/Computer Control. Proc. of the IEEE Conference on Decision and Control, Paper #FP5-3, San Diego, California, December 5-7, 1973.

Gardiner, P.C., and Edwards, W. Public Values: Multi-Attribute Utility Measurement for Social Decision Making. Proceedings of the 1975 IEEE International Conference on Systems, Man and Cybernetics, Paper No. THAM4-2, September 23-25, 1975.

Gilstad, D.W. and Fu, K.S. A Two-Dimensional Pattern Recognizing Adaptive Model of the Human Operator, Proceedings, Sixth Annual Conf. Manual Control, April, 1970.

Hagendorf, H., Kolbe, W., and Sommerfeld, E. About the Solution of Combinatorial Problems with Problem Solving Methods, Proc. 4th International Joint Conference on Artificial Intelligence, 1975.

Hayes, J.R.M. Human Data Processing Limits in Decision Making. In E. Bennett (ed.), Information System Science and Engineering. Proceedings of the First Congress on the Information Systems Sciences, New York: McGraw-Hill, 1964.

Hill, F. and Huey, B. A Design Language Based Approach to Test Sequence Generation, Computer, June 1977.

Kowalski, R. AND/OR Graphs, Theorem-Proving Graphs and Bi-Directional Search. Machine Intelligence, 6, 1971.

Leal, A. Adaptive Decisions in Ballistic Missile Defense, IEEE Trans., Sys., Man, Cybernetics. Vol. SMC-7, No. 6, May 1977.

Lenat, D. The Ubiquity of Discovering. Artificial Intelligence pp. 257-285, 1978.

Martelli, A. An Application of Heuristic Search Methods to Edge and Contour Detection. Comm. ACM Vol. 19, No. 2:73-83, February 1976.

McKenry, J.M., Enderwick, T.P., and Harrison, P.C. A Subjective Value Approach to Information Utility. Human Factors, 1971, 13(6): 503-509.

Montanari, U. Heuristically Guided Search and Chromosome Matching. Artificial Intelligence, Vol 1, No. 4, 1970.

Newman, J.R. Assessing the Reliability and Validity of Multi-Attribute Utility Procedures: An Application of the Theory of Generalizability. Social Science Research Institute, University of Southern California, Technical Report 001497-2-T, SSRI Research Report 75-7, Contract N00014-75-C-0487, ARPA Order No. 2105, Code 455, Office of Naval Research, July 1975.

Nilsson, N.J. Learning Machines, New York: McGraw Hill, 1965.

Nilsson, N. Problem-Solving Methods in Artificial Intelligence. McGraw-Hill: New York, 1971.

Powers, G. Non Numerical Problem-Solving in Computer-Aided Design in Computer-Aided Design. Vliestra, T. and Wiengla, R.T., (Eds.), 1973.

Raiffa, H. Preferences for Multiattributed Alternatives. Rand Corporation (Santa Monica, CA) Memorandum RM-5868-DOT/RC, April 1969.

Rouse, W.B. Cognititive Sources of Suboptimal Human Prediction, Ph.D. Dissertation, MIT, Report DSR 70283-19, September 1972.

Samet, M.G. Quantitative Interpretation of Two Qualitative Scales Used to Rate Military Intelligence. Human Factors, 1975, 17(2):192-202.

Samet, M.G. Utility Assessment by "Interpolation Between Corners": An Application to the Evaluation of Drug Efficacy. U.S. Army Research Institute for the Behavioral and Social Sciences (Arlington, VA) Unpublished Manuscript, December 1976.

Samet, M.G., Weltman, G., and Davis, K.B. Application of Adaptive Models to Information Selection in C3 Systems. Perceptronics (Woodland Hills, CA) Technical Report PTR-1033-76-12, Contract MDA 903-76-C-0241, Defense Advanced Research Projects Agency, December 1976.

Shortliffe, E.H. Computer-based medical consultations: MYCIN. New York, N.Y.: American Elsevier, 1976.

Slagle, J.R. Artificial Intelligence: The Heuristic Programming Approach, New York: McGraw-Hill, 1971.

Slovic, P. Consistency of Choice Between Equally-Valued Alternatives. Oregon Research Institute, ORI Research Bulletin Vol. 14(11), 1975.

Sridharam, N. Search Strategies for the Task of Organic Chemical Synthesis. 3rd International Joint Conference on Artificial Intelligence, 1973.

Steeb, R., Chen, K., and Freedy, A. Adaptive Estimation of Information Values in Continuous Decision Making and Control of Remotely Piloted Vehicles. Perceptronics Technical Report PATR-1037-77-8. Woodland Hills, California, August 1977.

Tversky, A. and Kahneman, D. Judgment Uncer Uncertainty: Heuristics and Biases. Science, 1974, 185:1124-1131.

Tversky, A., P. Slovic, and S. Lichtenstein, Subjective Optimality: Conceptual Issues, Empirical Results and Decision Aids, in Application of Research on Cognition to Man-Machine System Design, Ed. W. Edwards, Engineering Psychology Laboratory, University of Michigan, Report 010342-1-F, March, 1972.