

AD-A124 738

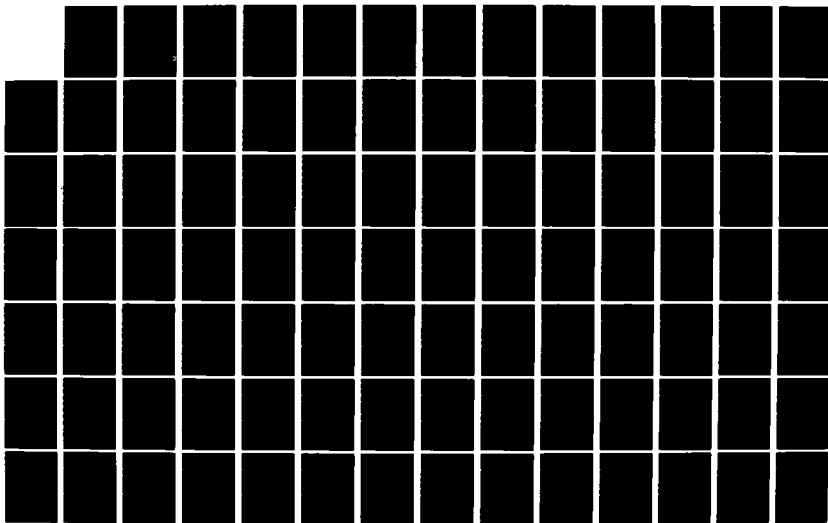
A DECISION ANALYSIS ALGORITHM FOR THE SAC WARNING AND
CONTROL SYSTEM(0) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING D E LEE
DEC 82 AFIT/GOR/05/82D-9

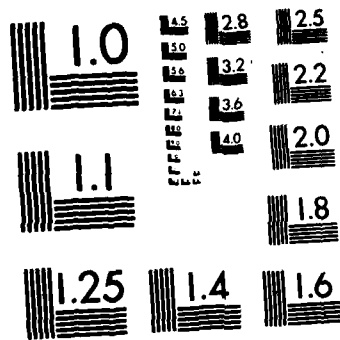
1/2

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1



AD A124730

DECISION ANALYSIS ALGORITHM
FOR THE
SAC WARNING AND CONTROL SYSTEM

THESIS

AFIT/GOR/OS/82 9 DOUGLAS E. LEE
CAPTAIN USAF

DTIC
ELECTE
FEB 22 1983
S D E

DTIC FILE COPY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Approved
for release

83 02 022 076

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GOR/OS/62D-9	2. GOVT ACCESSION NO. A124 730	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A DECISION ANALYSIS ALGORITHM FOR THE SAC WARNING AND CONTROL SYSTEM		5. TYPE OF REPORT & PERIOD COVERED MS THESIS
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Douglas E. Lee Captain		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT - EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1982
		13. NUMBER OF PAGES 122
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 Approved for public release; IAW AFR 190-1 19 JAN 1983 FREDERIC C. LYNCH, Major, USAF Director of Public Affairs <i>John E. WCLANER</i> JOHN E. WCLANER Dean for Research and Development Air Force Institute of Technology Wright-Patterson AFB, Ohio		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Decision Analysis Worth Assessment Multiple Objectives Multiple Attribute Utility Theory Decision Making		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Recent efforts in decision analysis have produced on-line, real-time, computer-based decision aids for assisting decision makers in clarifying preferences in the decision environment. This thesis was created to implement a decision maker's preference structure into the SAC warning system. The tool used to implement the decision maker's preferences is an additive worth assessment function designed to maximize the number of aircraft that can escape in a threatening environment while minimizing the cost associated with maintaining a given alert status. A sensitivity analysis package is		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

also provided showing the changes in variable or attribute levels that are needed to move from one alert status to another. Also provided is a user's/programmer's guide to facilitate the implementation of a decision maker's preference function.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

1

A
DECISION ANALYSIS ALGORITHM
FOR THE
SAC WARNING AND CONTROL SYSTEM

THESIS

AFIT/GOR/OS/82D-9 DOUGLAS E. LEE
CAPTAIN USAF

DTIC

Approved for public release; distribution unlimited.

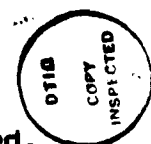
A
DECISION ANALYSIS ALGORITHM
FOR THE
SAC WARNING AND CONTROL SYSTEM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Masters of Science

by
Douglas E. Lee, B.S.
Captain USAF
Graduate Operations Research
December 1982

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



Approved for public release; distribution unlimited.

PREFACE

I wish to thank my advisor, Lieutenant Colonel Ivy Cook, who gave me a balance of freedom and professional guidance which allowed me to develop this thesis. My thanks also go to Captain Aaron Dewispelare, my reader, who gave me good advice.

My classmates, whose sense of humor and professionalism made the past eighteen months bearable, also deserve my thanks.

I would like to extend my appreciation to Captain Bruce Wayne Morlan for his invaluable assistance in not only helping me select a thesis topic, but also, as my interface with SAC.

Finally, I would like to thank my wife Cathy, and my son Jack, who helped me through the difficult times and ensured that I kept the proper perspectives.

Table of Contents

	Page
Preface	ii
List of Figures	iv
Abstract.	v
I. Introduction.	1
Problem Statement	3
Objective	3
II. Decision Analysis	5
Systems Analysis Paradigm	5
Systems Analysis Paradigm and SWCS.	6
Decision Theories	9
Multiple Attribute Utility Theory	9
III. Methodology	11
Worth Assessment Definition	11
Assumptions	12
Worth Assessment Procedure.	13
Scenario.	14
Scenario Implementation	15
Computer Implementation	16
IV. Sensitivity Analysis.	18
Attribute Weight Sensitivity.	18
Attribute Level Sensitivity	20
Scenario Sensitivity Analysis	21
Weight Sensitivity.	21
Level Sensitivity	22
Scenario Sensitivity Design	24
V. Conclusions and Recommendations	26
Conclusions	26
Recommendations	28
Bibliography	30
Appendix A: User's/Programmer's Manual	
Appendix B: Sample Output	

List of Figures

Figure	Page
2.1 SAC Warning System Objectives Tree.	8

Salvo Air Control
ABSTRACT

Recent efforts in decision analysis have produced on-line, real-time, computer-based decision aids for assisting decision makers in clarifying preferences in the decision environment. This thesis was created to implement a decision maker's preference structure into the SAC warning system. The tool used to implement the decision maker's preferences is an additive worth assessment function designed to maximize the number of aircraft that can escape in a threatening environment while minimizing the cost associated with maintaining a given alert status. A sensitivity analysis package is also provided showing the changes in variable or attribute levels that are needed to move from one alert status to another. Also provided is a user's/programmer's guide to facilitate the implementation of a decision maker's preference function.

↑

A
DECISION ANALYSIS ALGORITHM
FOR THE
SAC WARNING AND CONTROL SYSTEM

I. INTRODUCTION

One of the most important tools that the Commander-in-Chief of the Strategic Air Command (CINCSAC) has to aid in his decisions pertaining to nuclear war is the SAC Warning and Control System (SWCS). This tool performs many functions, among them, predicting the impact times of nuclear warheads targeted against Air Force bases in the United States. Currently, this portion of SWCS has no capability to incorporate any decision preferences that the CINCSAC may have pertaining to aircraft located at those bases.

There are many reasons why a decision analysis algorithm should be included in such a tool -- among these reasons are cost and complexity. Any time that a decision is made, some sort of cost is assessed. In the case of the SAC warning system, money and manpower are exhausted while

aircraft life is shortened if aircraft are put on increased alert status or launched to protect them from a threat that does not materialize. However, if the alert status is not increased or aircraft are not launched and the threat is real, equipment vital to our nation's defense is lost.

Complexity is another reason for a decision analysis tool. In a warning environment, deciding on the aircraft alert status has to be made quickly. Without any sort of aid to address all the variables involved in a decision of this magnitude, the decision made could be incomplete or incorrect because it was based on one or two major variables instead of all the salient variables affecting the decision. This problem of incorporating all salient variables can be compounded when the alert status has to be determined for many bases, such as all the SAC bases, instead of only one base. With a decision algorithm, the decision maker is helped in two ways: first, he has access to the salient variables involved with making a decision while he is in a nonthreatening environment; and second, when the environment becomes threatening, the tool can be used as a basis for the decision that has to be made.

The problem addressed by this thesis, then, is developing the framework for a decision analysis algorithm that can be incorporated into the SAC Warning and Control System. This algorithm will incorporate the decision maker's preferences to minimize the alert status while maximizing aircraft survivability. Again, it should be

stressed that this decision analysis algorithm is a tool to aid in decision making -- not an algorithm to replace the decision maker.

Problem Statement

The primary purpose of this thesis is to provide a decision analysis algorithm that can be incorporated into the SAC Warning and Control System. This algorithm will aid the SAC decision makers in assessing the "optimal" alert status given the decision maker's preferences.

The algorithm provided must be one that can be easily followed in order to be implemented on the SAC resident computer without complication. The implementation not only includes the actual program coding, but also includes designing the decision maker's preference structure. Another restriction for this algorithm is that it must be a "real time" program that can be processed in terms of milliseconds.

Objective

The objective of this thesis is to develop a decision analysis algorithm that the SAC warning personnel can understand and implement. In order to attain this objective, a brief description of decision analysis will be presented. Next, the methodology developing both a decision analysis algorithm and a hypothetical scenario will be discussed. A sensitivity analysis showing how the input parameters for the decision maker's preference function

would have to be changed in order to move from the "optimal" or "proposed" alert status to any other alert status will then be presented. The "final" product that will be given to the SAC personnel will consist of the thesis, a user's/programmer's manual, and sample output. A copy of Lt. Wayne Stimpson's thesis, MADAM: Multiple Attribute Decision Analysis Model, will also be included as an alternative for developing a decision maker's preference function.

II. DECISION ANALYSIS

In his book, Goal Programming for Decision Analysis, Dr. S. M. Lee defines decision analysis as:

" . . . the analytical process by which one selects specific courses of action from a set of possible courses of action in order to achieve his goals."
(Lee, 1972: 3)

This process is an approach to problem solving that attempts to structure the problem by dividing it into smaller subunits that can be dealt with on an individual basis (Morlan, 1979: 8).

Systems Analysis Paradigm

In order to implement decision analysis, a seven step systems analysis paradigm has been developed (Sage, 1977: 66). The seven steps making up this paradigm are

- 1) problem definition,
- 2) value system design,
- 3) system synthesis,
- 4) system analysis,
- 5) optimization,
- 6) decision making, and,
- 7) planning for action.

Of these seven steps, the first three are common to any decision analysis model. In the problem definition phase,

the variables or elements of the problem are identified as are the needs of the decision maker. The major constraints are also defined in this step. Attribute or performance measure definition is the major activity in value system design. One of the most effective methods for value system design is the objectives tree. With an objectives tree, the major objectives that define the problem are divided into subobjectives. This division process continues until the subobjectives can be used as measurements to determine the degree of attainment of their respective major objectives. The reason that an objectives tree is so effective is that when presented to the decision maker, there is no doubt in communication. The objectives tree represents the analyst's conception of the problem and any changes the decision maker has can easily be reflected on the tree. The final common step is system synthesis. In system synthesis, alternatives to solving the problem are listed (Sage, 1977: 73). The techniques used in the remaining four steps are dependent upon the decision analysis tool that is used.

Systems Analysis Paradigm and SWCS

Before the final four steps of the systems analysis paradigm can be accomplished, the first three steps should be applied to the SAC warning system. In the hypothetical scenario, the problem is to find the best alert posture for each SAC base possessing aircraft. The variables involved in this problem include:

- 1) aircraft,
- 2) the impact time of the threat,
- 3) the time it takes the decision maker to release the aircraft,
- 4) spacing time between aircraft taking off,
- 5) crew reaction time, and,
- 6) costs involved with maintaining an alert status.

The major constraints in this problem are system related. The first constraint is that of execution time. In the SAC warning system there is a need for a "real time" program in order to make effective decisions, so that every piece of software defining the SAC warning system must run in fractions of a second. The other constraint is that of system compatibility. The current system is deterministic so that any decision analysis software should also be deterministic in nature.

The next step in the systems analysis paradigm is to find performance measures or attributes for the system. In the case of the SAC warning system variables, while the aircraft can be defined in terms of numbers and all the variables associated with time can be defined in seconds, defining alert costs is more difficult. One approach would be to actually define all the costs involved with an alert status (such as cost of fuel, aircraft parts, etc.); however, there are some costs that are very difficult to define. These costs include morale of the crew members and value of aircraft life. Another method is to apply a Direct Worth Estimate (DWE) to each possible alert status (Sage, 1977: 356). With DWEs, the decision maker places a worth on

each alert posture. The primary reason for having alert statuses is to insure aircraft survivability. With the variables presented, aircraft escaping is a function of impact time, decision time, alert time or crew reaction time, and spacing time between aircraft. Aircraft escaping can be determined using Equation 2.1.

$$ACES = INT \left[1 + \frac{IMPTME - DECTME - ALTTME}{SPCTME} \right] \quad (\text{Eq 2.1})$$

where,

ACES = number of aircraft escaping
 IMPTME = impact time of threat
 DECTME = decision time
 ALTTME = crew reaction time due to alert posture
 SPCTME = spacing time.

The objectives (shown in Figure 2.1) for the SAC warning problem are to maximize the number of aircraft that survive (escape) while minimizing the alert costs for each base.

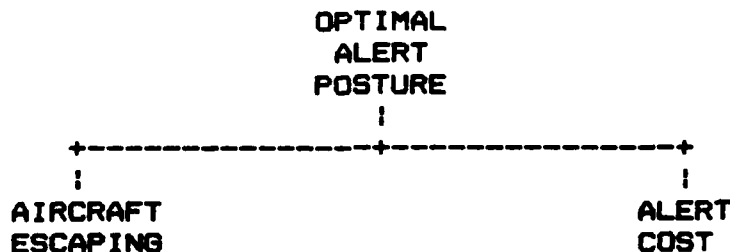


Figure 2.1. SAC Warning System Objectives Tree

The final common step in the systems analysis paradigm is system synthesis in which the alternatives for the problem are listed. The alternatives that are in the SAC warning

system are the possible alert postures. The best alert status will be the one that is the most effective in attaining the decision maker's desires.

Decision Theories

There are many techniques that can be used in the remaining four steps of the systems analysis paradigm, but they are dependent upon the decision analysis approach implemented. Among these approaches are goal programming (Lee, 1972), statistical analyses (Bowman, 1963; Goldberg, 1970), and multiple objective optimization theory (MacCrimmon, 1973; DeWispelare & Sage, 1979). Multiple Attribute Utility Theory (MAUT) is one approach that has been used effectively in the military environment (Chinnis, et al, 1975; Allen, et al, 1977).

Multiple Attribute Utility Theory

Multiple attribute utility theory is defined as:

"A type of decision theory; requires the analyst to elicit preference information concerning the attributes of proposed alternative policy of the decision maker; utilizing the decision maker's preferences, the analyst forms a scalar choice function (SCF). The SCF is used to evaluate the outcomes of the alternatives, score, and subsequently rank the alternative policies for the decision making step." (DeWispelare, 1980)

Advantages to MAUT are its ability to solve multiple conflicting objectives that have noncommensurable units and its final product of a complete ordering of the alternatives. Disadvantages include the elicitation process

with the decision maker to develop the scoring functions and criterion weights along with the time it takes for implementation.

Two areas that MAUT can be further divided into are risk and certainty. MAUT associated with risk establishes attribute values and alternatives with respect to the decision maker's attitude toward risk. The solutions derived incorporate utility functions measuring the decision maker's risk averseness or proneness to determine alternative ranking. While the problem solution is much more complex, a truer representation of the real world is attained (Lee, 1981: 5).

Decision making under certainty or riskless decision making allows the decision maker to state with complete certainty the outcomes associated with each action. The weights and attribute values are determined exactly, making the problem easily solvable. Treating an uncertain decision as a riskless decision may, in some cases, be justified because the precision lost is far outweighed by the reduction in effort (Fischer et al, 1978: 61-62).

The decision analysis tool used as the algorithm in the SAC Warning and Control System is worth assessment -- a MAUT, riskless decision making tool. The major reason for the use of this method is the time constraint imposed by SWCS. Another reason is that the current software in SWCS does not incorporate any probabilistic functions.

III. METHODOLOGY

The decision analysis algorithm designed for the SAC Warning and Control System was a worth assessment procedure. The major reasons for this method were system related. First, was the need for an algorithm that could be executed in a fraction of a second, and second, was the need for a program similar to the software currently in SWCS -- specifically, a deterministic program. Presented in this section are the steps for implementing worth assessment, the hypothetical scenario designed for this algorithm, and a brief description on how to implement worth assessment on the SAC warning computer.

Worth Assessment Definition

Worth assessment is a decision analysis procedure that finds the worth or value for each possible course of action (alternative) in a problem. This procedure first decomposes the problem into measurable factors or attributes and assigns a worth to the level of each attribute that the alternative attains. Next, these individual worths are weighted with respect to their importance to the problem, and then, all the worths are summed to form an overall worth for that particular alternative. This measure of worth indicates preferences among the alternatives (Farris & Sage,

1975: 1160).

Assumptions

Before an additive worth assessment algorithm can be used in SWCS, two major assumptions should be met. These assumptions are independence and constant tradeoff.

Independence. For the worth assessment function in SWCS, mutual preferential independence is assumed. Milan Zeleny states:

"The pair of attributes X and Y is preferentially independent of attribute Z if the value trade-off between X and Y is not affected by a given level of Z." (Zeleny, 1982: 420)

In order for mutual preferential independence to exist, for a given set of attributes $X_1, \dots, X_n$, every subset of these attributes must be preferentially independent of its complementary set. One problem with mutual preferential independence is that as the number of attributes increases, the number of preferential independence conditions that must be satisfied increases astronomically. If there are multiple attributes, not only do the pairs of attributes have to be preferentially independent, but also the triples of attributes, etc (Keeney & Raiffa, 1976: 111-112). To lessen the amount of work in proving preferential independence, a weakened set of conditions can be used to prove preferential independence. These conditions state that if every pair of attributes is preferentially independent of its complement, then the attributes are

preferentially independent (Keeney & Raiffa, 1976: 112). For this application, simple attribute independence was assumed because there were only two attributes.

Constant Tradeoff. The other assumption needed for implementing a worth assessment algorithm is constant tradeoff. For constant tradeoff to exist, the importance of all the attributes to the decision maker must remain constant over their respective ranges (Morlan, 1979: 20). While the aircraft escaping attribute was designed as a linear function (which results in constant tradeoff), the alert cost attribute was nonlinear, and violated the constant tradeoff assumption.

These assumptions are not presented to justify the worth assessment model; they are presented to validate the model. The importance of the model designed is not how well it adheres to the assumptions; but rather, how well it models the decision maker's value function (DeWispelare, 1982).

Worth Assessment Procedure

Listed below are the steps used to implement a worth assessment algorithm. An indepth discussion of this procedure is presented in Appendix A of this thesis.

- 1) List overall performance objectives.
- 2) Construct a performance criteria hierarchy.
- 3) Select physical performance measures.
- 4) Define worth function for each attribute.
- 5) Establish relative importance between subobjectives.
- 6) Adjust weights to reflect confidence in the measures.

Once these steps have been accomplished, the best

alternative is determined by substituting the values of the attribute levels associated with a given alternative into the worth assessment equation (Eq 3.1), and finding the overall worth for each alternative. The alternative with the highest worth score is deemed the best alternative.

$$ALTWTH = \sum_i ATTWGT_i * VALLVLi \quad (Eq\ 3.1)$$

where,

ALTWTH = alternative worth score
 ATTWGT_i = weight associated with the ith attribute
 VALLV_{Li} = value for the level of the ith attribute

In the worth assessment procedure, ATTWGT_i is derived by following steps 1, 2, 3, 5, and 6, while VALLV_{Li} is developed in step 4. It should be noted that step 6 does not have to be performed, and was not used in the hypothetical scenario.

Scenario

Because of the nature of this thesis and the classification of the material that would be involved, a hypothetical scenario was developed to show how a worth assessment algorithm would be implemented in the SAC warning system. This scenario, even though hypothetical, was designed for easy modification so that actual inputs could be used with minimal effort. A detailed description of this scenario is given in the User's/Programmer's Manual (Appendix A).

Scenario Implementation

The purpose of this scenario was to find the best alert status for a given SAC base using the decision maker's preference function. The performance objectives associated with this problem were aircraft survivability and alert cost. The performance criteria hierarchy was relatively simple because of the small number of factors used to describe the problem. The physical performance measure for aircraft survivability was the number of aircraft that could escape while the measure for alert cost was established relatively between alert statuses with the lower alert statuses receiving a higher measure than the higher alert postures.

The worth function defined for aircraft escaping was a linear function using percentage of aircraft that could escape from a base and value for escaping aircraft. A value of 1.0 was given to an aircraft escaping attribute that had 100% escaping while a 0.0 was assigned for 0% aircraft escaping. The percentage of aircraft escaping was used because the function could then be applied to any base, regardless of the number of aircraft located at the base. The worth function for alert cost was discrete, and was designed so that the lower the alert status, the higher the value. Methods for deriving nonlinear functions, presented by Keeney and Raiffa (1976: 91-96), are the conjoint scaling technique and the midvalue splitting technique. With the relatively simple decision function, the weight assignment

process was trivial. For this scenario, the aircraft escaping attribute was considered three times as important as the alert cost attribute, producing weights of .75 and .25, respectively. Because of the small number of attributes, no adjustments in weights to reflect the decision maker's confidence were made.

Finally, the best alert status was derived by addressing all the alert statuses (using Eq 3.1) and finding the worth scores associated with each. The alert status with the highest worth score was then presented as the best or proposed alert status.

Computer Implementation

The worth assessment software presented in this thesis assumes that the user is concerned with identifying an optimal alert status for each base in the database file. The model designed addresses one base at a time, looking at all the possible alert postures for that base. The values for each attribute are calculated and then summed to find the overall worth for the particular alert status. After the worths for all the postures for a given base are calculated, the posture with the highest worth or value is selected as the proposed alert status.

The software is designed for easy modification. The first step in changing the software is to develop a value function for each attribute in the decision maker's overall worth function. Next, these functions are then placed into the software where the attribute values are calculated.

Finally, the weights used for ranking the attributes are defined and placed where the overall worth for a given alert posture is calculated.

An indepth procedure on implementing the worth assessment software into SWCS is presented in Appendix A of this thesis along with the flowchart and coding for the hypothetical scenario software. Sample output is presented in Appendix B.

IV. SENSITIVITY ANALYSIS

The purpose of this chapter is to aid the decision maker in understanding how the parameters -- specifically, the weights in the worth assessment function and the attribute levels for a given outcome -- affect the problem. The parameters that will be addressed are the weights assigned to a performance measure with respect to the overall problem objective and the attribute levels for a particular alternative when compared to another alternative. The sensitivity analysis designed addresses the changes in the weights for the performance measures and the changes in the attribute levels that are necessary to make the value or worth of two alternatives equal. Limitations in a sensitivity analysis of this nature include the inability to conceptualize the changes in an alternative score due to simultaneous changes in a set of attributes, and the inability to deal with ambiguous rate changes in a set of attributes. One approach to accomplishing this type of sensitivity analysis is to consider the alternative score response to modifications on the attributes taken independently.

Attribute Weight Sensitivity (Stimpson, 1981: 39-41)

The purpose of the attribute weight sensitivity

analysis is to determine how the overall worth scores assigned to alternatives or outcomes change relative to one another when the weight of a particular attribute or performance measure is modified. Referring to Equation 3.1, the worth score assigned to an alternative is the sum of the products of the weights and values associated with each attribute. To find how the alternative worth score changes with respect to a particular attribute weight change, the current attribute weight is replaced by the new attribute weight. Simply changing the attribute weight will not work because the hierarchy will no longer be normalized. For a new attribute to be introduced without losing the normalized decision tree, the cumulative weights of the remaining attributes must equal one minus the new attribute weight. The new alternative worth score can be calculated by using Equation 4.1.

$$\text{ALTWTH}' = \text{VALLVL}_j * \text{ATTWGT}_j' + [(1 - \text{ATTWGT}_j') / (1 - \text{ATTWGT}_j)] * (\text{ALTWTH} - \text{VALLVL}_j * \text{ATTWGT}_j) \quad (\text{Eq. 4.1})$$

where,

ALTWTH' = new alternative worth score
 ALTWTH = old alternative worth score
 VALLVL_j = value for the level of the jth performance measure (attribute)
 ATTWGT_j' = new weight associated with the jth performance measure (attribute)
 ATTWGT_j = old weight associated with the jth performance measure (attribute).

In the hypothetical scenario, which is a two attribute case, the weight of the second attribute can be calculated by

subtracting the first attribute weight from 1 ($1 - \text{ATTWGT}_j$)).

Attribute Level Sensitivity (Stimpson, 1981: 42-43)

An attribute level sensitivity analysis is used to examine the robustness of the "optimal" solution to changes in the level of a particular attribute. In order to use the attribute level sensitivity analysis, two assumptions must be made. First, it is assumed that all the alternatives are independent of each other, that is, the changes in the attribute levels of one alternative have no influence on the attribute levels of any other alternatives. The second assumption is that the weights of the attributes are not affected by changes in an alternative's attribute level. With these assumptions, only an alternative that has a change in an attribute level will experience a change in the alternative worth. If an alternative has a change in an attribute level, the change in the alternative worth will be a direct function of the weight associated with the attribute (Eq. 4.2).

$$\begin{aligned} \text{ALTWTH}' &= \text{ALTWTH} - (\text{ATTWGT}_j * \text{VALLVL}_j) \\ &\quad + (\text{ATTWGT}_j * \text{VALLVL}_j') \end{aligned} \quad (\text{Eq 4.2})$$

where,

ALTWTH' = new alternative score
 ALTWTH = old alternative score
 ATTWGT_j = weight associated with attribute j
 VALLVL_j' = new value for the level of the j th attribute
 VALLVL_j = old value for the level of the j th attribute.

Scenario Sensitivity Analysis

The major reason for providing a sensitivity analysis package is to give the decision maker the capability to see how the weights assigned to an attribute or the level of an attribute (for a given alternative) must change in order for the value of a non-optimal alert status to exceed that of the optimal alert status. This type of sensitivity analysis gives the decision maker the exact inputs -- in terms of attribute weights and attribute levels -- that are needed to make a non-optimal alert status, optimal. This is helpful in showing the decision maker how sensitive the alert score is to its inputs, thus, showing the decision maker how easy or difficult it would be to change alert postures.

The general sensitivity analysis equations (Eq. 4.1 and Eq. 4.2) show how the alternative worth will change with respect to a change in either attribute weight or attribute level. To show how a given weight or level must change in order for their alternative worth to equal the worth of another alternative, the equations must be changed.

Weight Sensitivity

To find the attribute weight needed to change a non-optimal solution or alternative to optimal, Equation 4.1 must be modified. This modification is shown in Equation 4.3.

$$\begin{aligned} \text{ATTWGT}_j' &= [\text{ALTWTH} - (\text{VALLVL}_j * \text{ATTWGT}_j) \\ &\quad - \text{ALTWTH}' + (\text{ALTWTH}' * \text{ATTWGT}_j)] \quad (\text{Eq 4.3}) \\ &\quad / [\text{ALTWTH} - \text{VALLVL}_j] \end{aligned}$$

where,

ATTWGT_j' = new weight associated with the j th attribute
 ATTWGT_j = old weight associated with the j th attribute.
 ALTWTH' = new alternative worth score
 ALTWTH = old alternative worth score
 VALLVL_j = value for the level of the j th attribute

For the hypothetical scenario developed, ATTWGT_j' reflects the new weight given to the attribute (either aircraft escaping or alert cost) of a "non-proposed" alert status while ALTWTH' is the worth score of the proposed alert posture and ALTWTH is the worth score currently assigned to the alert posture. While Equation 4.3 adjusts one of the attribute weights, the other weight will also be changed because of the weight normalization in the decision tree. As stated previously, the weight for the second attribute in the scenario presented can be calculated by subtracting the weight of the first attribute from 1.

Level Sensitivity

As with Equation 4.1, Equation 4.2 must also be modified in order to find the attribute level that must be attained before a given alternative can become an optimal alternative. The necessary changes are shown in Equation 4.4.

$$\text{VALLVL}_j' = \frac{\text{ALTWTH}' - \text{ALTWTH} + (\text{ATTWGT}_j * \text{VALLVL}_j)}{\text{ATTWGT}_j} \quad (\text{Eq 4.4})$$

where,

VALLVL_j' = new value for the jth attribute
VALLVL_j = old value for the jth attribute
ALTWTH' = new alternative worth score
ALTWTH = old alternative worth score
ATTWGT_j = weight associated with the jth attribute.

When applying Equation 4.4 to the hypothetical scenario, VALLVL_j' would represent the attribute value level (for a "non-proposed" alert status) that must be attained in order for this alert posture to become the proposed or optimal alert status. ALTWTH' would be the worth or score of the proposed alert status, and ALTWTH would be the score associated with the "non-proposed" alert posture.

An extension to the attribute level sensitivity analysis implemented with the worth assessment software is the conversion of the attribute level change into its basic elements. Once the needed attribute value level is calculated, the attribute value is converted into the performance measure used for that attribute (in the hypothetical scenario, the number of aircraft escaping would be a performance measure) using the value function derived for that attribute. After the performance measure level is determined, the performance measure is then converted back into its basic components.

In the hypothetical scenario, the attribute level extension is applied to the aircraft escaping attribute. After the needed value of aircraft escaping for a "non-proposed" alert status is calculated, the value is then

converted into the number of aircraft that must escape through use of the aircraft escaping value function. Because the number of aircraft escaping is a function of threat impact time, decision time, alert time, and spacing time (Eq 3.1 in User's/Programmer's Manual), the times that the decision maker has control of can be modified to show how the number aircraft needed to escape can be attained. In the case of aircraft escaping, spacing time and a combination of decision time and alert time can be manipulated.

Scenario Sensitivity Design

The sensitivity analysis designed for the SAC warning worth assessment software addresses all the alert postures below the proposed alert status. For example, if the proposed alert status was 3, the sensitivity analysis would include the changes necessary to make alert status 1 or alert status 2 the proposed alert status. The first step in the sensitivity analysis finds the changes needed in the aircraft escaping attribute level along with the changes needed in the weights for both the aircraft escaping and the alert cost attributes. The attribute level for alert cost is not derived because the meaning would be difficult to interpret. After the attribute level and weights are determined, the aircraft escaping attribute is then translated into the changes needed in spacing time and the decision time/alert time (reaction time) combination. This translation can be accomplished by using Equation 2.1 and setting the number of aircraft escaping to the level needed

to make a given alert status the proposed alert status. Four reference points for the spacing time and reaction time are provided. The two end points represent the changes in only spacing time or reaction time, while the two midpoints are combinations of the two times.

The scenario sensitivity analysis software is presented in Appendix A, and sample output can be seen in Appendix B.

V. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

With the advent of computers, man has been given the capability to do things much faster. The computer is an extension of man and does what it is programmed to do. One aspect of computer use is that of Management Information Systems (MIS). With the development of MIS, the decision maker has access to large amounts of data -- in some cases too much data, to solve problems. It is becoming more and more evident that man is the weakest link (in terms of time) in any decision making process. Current efforts by Morlan (1979), Lee (1981), and Stimpson (1981) have provided a capability for decision function design in a real-time, on-line, interactive computer program; however, one more step can be taken in terms of decision making with respect to the computer. The primary purpose of this thesis is to develop an initial step into providing the capability to use the computer to make "automated" decisions. This initial step is in the form of a proposed decision and is not, and should not, be construed as validated. With decision analysis still in its infancy when it comes to entrusting a computer with responsibility, automated decision making has to be monitored until confidence can be developed in the concept

of a machine making a decision from an "inputted" decision function. After all, it is not a computer making the decision -- it is a decision maker that has given the computer instructions on how to make the decision.

The most important observation made in this thesis effort concerns the problems that will be encountered in implementing an automated decision function. The first problem encountered with this thesis was the attitude people have about putting their fate into the "hands" of a computer. When implementing a system of this nature, it is important to remember that the computer is doing what it is programmed to do. Even though there is no capability for a computer to be rational, a computer does not succumb to stress. In a quick reaction situation it will address all the variables it is programmed to consider -- unlike man who has a tendency to make decisions on a few major variables instead of all salient variables. The solution to this problem is to involve the decision maker as much as possible in the algorithm design process. With decision maker participation, the negative attitude towards computerization is eased because the decision maker has an opportunity to understand that the decisions are being made using his preference function.

The second problem involved with implementing a decision analysis algorithm, especially in an operational environment, is judging the system on the scenario developed (Lucas & Ruff, 1977: 59). While the scenario is important,

attention should be directed to the system. If the scenario is not accurate but does not invalidate the system, the system should not be rejected, instead the scenario should be corrected.

Recommendations

An immediate application of this thesis includes implementation on the SAC Warning and Control System. This implementation should be a two step process. The first step would be to modify the hypothetical scenario by substituting actual performance parameters, and operating the software in an "off line" mode. In an "off line" mode, the software could be run in a nonthreatening environment to familiarize the decision maker and his personnel with the worth assessment procedures, while also developing operating procedures.

The second step would be to develop the decision maker's preference function -- utilizing the decision maker's attitudes and performance measures (attributes). At this point in time, it would be easier and more beneficial because the decision maker would have a basic understanding of how worth assessment works in the SAC environment, and also have the basic operating procedures developed. After the decision maker's preference function has been validated (through "off line" exercises) it should be implemented "on line."

One short-term development that would enhance the worth assessment software would be to address the functional form

of the decision maker's preferences. Once the worth assessment concept has been accepted, other functional forms, such as the multiplicative form, should be implemented to better fit the decision maker's preference function.

Finally, a software improvement that is dependent upon SAC warning enhancements would be to implement utility functions -- allowing for the capability of risk -- into the decision analysis function. Currently, the warning system is deterministic, making it difficult to effectively use a risk function; however, as the software developed incorporates probability, a utility function will be useful.

Bibliography

- Allen, J., D. Buede, and M. O'Connor. The Use of Multi-Attribute Value Assessment Techniques in the Development of a Marine Corps Combat Readiness Evaluation System (MCCRES). McLean, Va: Decisions and Designs, Inc., 1977.
- Bowman, E. H. "Consistency and Optimality in Managerial Decision Making," Management Science, 9: 310-321, 1963.
- Chinnis, T. O., C. W. Kelley, R. D. Minckler, and M. F. O'Connor. Single Channel Ground and Airborne Radio System (SINCGARS) Evaluation Model. McLean, Va: Decisions and Designs, Inc., 1975.
- DeWispelare, A. R., Captain, USAF. Lectures in ST6.31, Weapons Effectiveness/Tradeoff Studies, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, 1980 & 1982.
- DeWispelare, A. R. and A. P. Sage. "On Combined Multiple Objective Optimization Theory and Multiple Attribute Utility Theory for Evaluation and Choicemaking" Journal of Large Scale Systems, 2: 1-19, 1981.
- Farris, D. R. and A. P. Sage. "On Decision Making and Worth Assessment," International Journal of System Sciences, 6: 12: 1135-78, December 1975.
- Fischer, G. W., W. Edwards, and C. W. Kelly, III. Decision Theoretic Aids for Inference, Evaluation, and Decision Making: Review of Research and Experience. McLean, Va: Decision and Designs, Inc., Feb 1978. (AD053962).
- Goldberg, L. R. "Man Versus Model of Man: A Rationale, Plus Some Evidence for a Method of Improving on Clinical References." Psych Bulletin 73: 422-432, 1970.
- Keeny, Ralph L. and Howard Raiffa. Decisions with Multiple Objectives: Preference and Value Tradeoff. New York: John Wiley and Sons, 1976.
- Lee, David B. Enhanced Decision Analysis Support System. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1981. (AFIT/GST/OS/81M-8).
- Lee, S. M. Goal Programming for Decision Analysis. Philadelphia: Auerbach Publishers, 1972.

Lucas, Gary L. and Jan Ruff. An Investigation of Operational Decision Aids. Arlington, Va: System Planning Corp, July 1977. (AD-A047 147).

MacCrimmon, K. R. "An Overview of Multiple Objective Decision Making" in Multiple Criteria Decision Making, J. L. Cochrane and M. Zeleny (ed.), Columbia, S. C.: University of South Carolina Press, 1973.

Miller, James R. Professional Decision Making. New York: Praeger Publishers, 1970.

Morlan, Bruce W. A Computer-Based Decision Analysis Support System. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, Dec 1979. (AFIT/GOR/SM/79D-6, AD-A083706).

Sage, Andrew P. Methodology for Large-Scale Systems. New York: McGraw-Hill, 1977.

Stimpson, Wayne A. MADAM: Multiple-Attribute Decision Analysis Model. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, Dec 1981. (AFIT/GOR/AA/81D-1).

APPENDIX A
USER'S/PROGRAMMER'S MANUAL
FOR
WORTH ASSESSMENT IMPLEMENTATION
ON THE
SAC WARNING AND CONTROL SYSTEM

0

WORTH ASSESSMENT
FOR THE
SAC WARNING AND CONTROL SYSTEM

USER'S/PROGRAMMER'S GUIDE

Douglas E. Lee
Captain USAF

This user's/programmer's guide was written by Captain Douglas E. Lee as documentation for implementing a worth assessment algorithm on the SAC Warning and Control System. This documentation was written as part of a Master's thesis for the Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Table of Contents

	Page
List of Figures	iii
I. Introduction	1
II. Worth Assessment	2
Worth Assessment Definition	2
Assumptions	2
Worth Assessment Procedure	3
III. Algorithm Implementation for SWCS	13
The Environment	13
The Bases	13
The Aircraft	14
The Alert Status	16
The Value Functions	17
The Algorithm	18
The Inputs to SWCS	19
IV. Software Implementation	21
Overview	21
Implementation	21
V. Program Source Code	28
VI. Sensitivity Analysis	44
Weight Sensitivity Analysis	45
Level Sensitivity Analysis	46
User Implementation	47
Sensitivity Analysis Flowchart	48
Program Source Code	51
Bibliography	72

List of Figures

Figure		Page
2.1	Car Buying Worth Assessment Hierarchy . . .	4
2.2	Car Hierarchy with Defined Characteristics.	5
2.3	Weights for the Car Buying Scenario	9
3.1	Value Function for Aircraft Escaping	17
3.2	Value Function for Alert Status.	18
4.1	Flowchart for Worth Assessment	22
6.1	Flowchart for Sensitivity Analysis	49

I. INTRODUCTION

The major reason for this user's/programmer's manual and the thesis in which it is a part was to provide the Commander-in-Chief of SAC with a tool to aid in his decision making process concerning the safety of his aircraft. In a quick reaction environment, such as that in SAC, there is a tendency for decision makers to make judgements on courses of action using only one or two of the major factors affecting the outcome instead of all the salient variables. This manual was designed to aid the decision maker in developing a decision making tool and implementing it on a computer.

The purpose of this manual is twofold. First, it describes worth assessment and sensitivity analysis, and second, it illustrates how a worth assessment model with sensitivity analysis is implemented. Worth assessment is described step by step using a very simple example. To delineate how it can be implemented on the SAC Warning and Control System (SWCS), a scenario used in generating the decision algorithm is given. It should be noted that the scenario is hypothetical, and in no way is intended to represent actual input -- the format of the scenario is intended to be similar to aid in implementation.

II. WORTH ASSESSMENT

Worth Assessment Definition

Before worth assessment can be defined, the concept of worth must first be addressed. James R. Miller defines worth in this way:

"The worth of any object or activity inheres in the degree to which it or its consequences are perceived by a given individual in a given situation as satisfying his preferences." (Miller, 1970: 14)

Worth assessment, then, is a decision analysis procedure that finds a worth or value for each possible course of action (alternative) in a problem. This procedure is accomplished by first assigning a worth score to the level of each factor or attribute associated with an alternative, and then, combining these worth scores to form an overall worth score for that particular alternative. This measure of worth indicates preferences among the alternatives (Farris & Sage, 1975: 1160).

Assumptions

Independence. To use the worth assessment function addressed in this user's/programmer's manual, value independence is assumed. Value independence means that the value of one of the variables or attributes in the problem is not affected by the levels of the other attributes (Morlan, 1979: 19). For example, if we were looking at hamburgers and milkshakes, value independence exists if the

value or importance of hamburgers does not rely on the number of milkshakes.

Constant Tradeoff. Another assumption that is closely related to independence is constant tradeoff. For constant tradeoff to exist, the value of all attributes must remain constant over their respective ranges (Morlan, 1979: 20). Again with the hamburger and milkshake example, constant tradeoff exists if the values for both the hamburgers and milkshakes remain constant given any combination.

These assumptions are not presented to justify the model, they are presented to validate the model. The importance of the model that will be designed is not how well it adheres to the assumptions, but rather, how well it models the decision maker's value function (DeWispelare, 1982).

Worth Assessment Procedure

Listed below are the necessary steps in worth assessment. To assist in understanding, a car buying example presented in Methodology for Large-Scale Systems by Andrew P. Sage will be developed with the procedure.

1 List Overall Performance Objectives or Attributes (Sage, 1977: 356). In this step, all factors important to the problem should be listed. These factors should

- a) be of the highest degree of importance,
- b) include all relevant objectives, and,
- c) be mutually exclusive.

In the car buying example, the important factors that

will be addressed are cost, aesthetics, and safety.

2 Construct a Hierarchy of Performance Criteria (Sage, 1977: 356). The next step in the procedure is to subdivide the higher level objectives into lower level objectives. This process continues until all subobjectives are measurable attributes. This step results in a tree diagram with the top representing the high level objectives and the lower levels representing the subobjectives. At this point, one of the most important steps has been accomplished in that the problem has been translated into a well-defined, easily understood form.

The objectives in the car example with their subobjectives are presented in a tree diagram below (Fig 2.1).

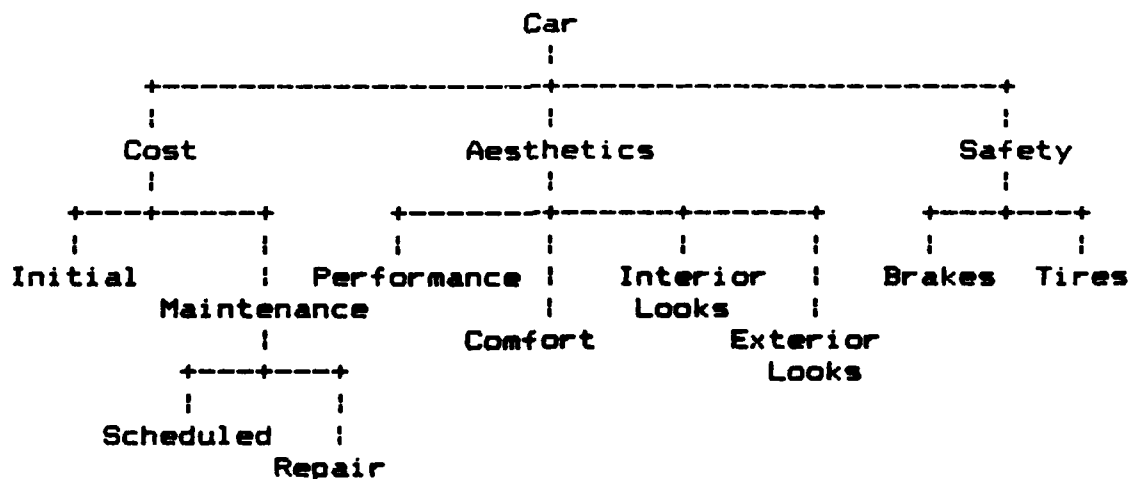


Figure 2.1. Car Buying Worth Assessment Hierarchy

3 Select Appropriate Physical Performance Measures (Sage, 1977: 356). After the hierarchy is established, the

next step is to define physical characteristics of performance that can measure the degree of satisfaction that the attribute or subobjective has attained. There may be some instances in which there may not be any physical characteristics that can define the attribute. In this case, a Direct Worth Estimate (DWE) has to be made for that attribute. A DWE is the decision maker's worth for that attribute's level in each alternative.

In the example, the physical characteristics are shown in Figure 2.2. Note the Direct Worth Estimates for the Aesthetic subobjectives.

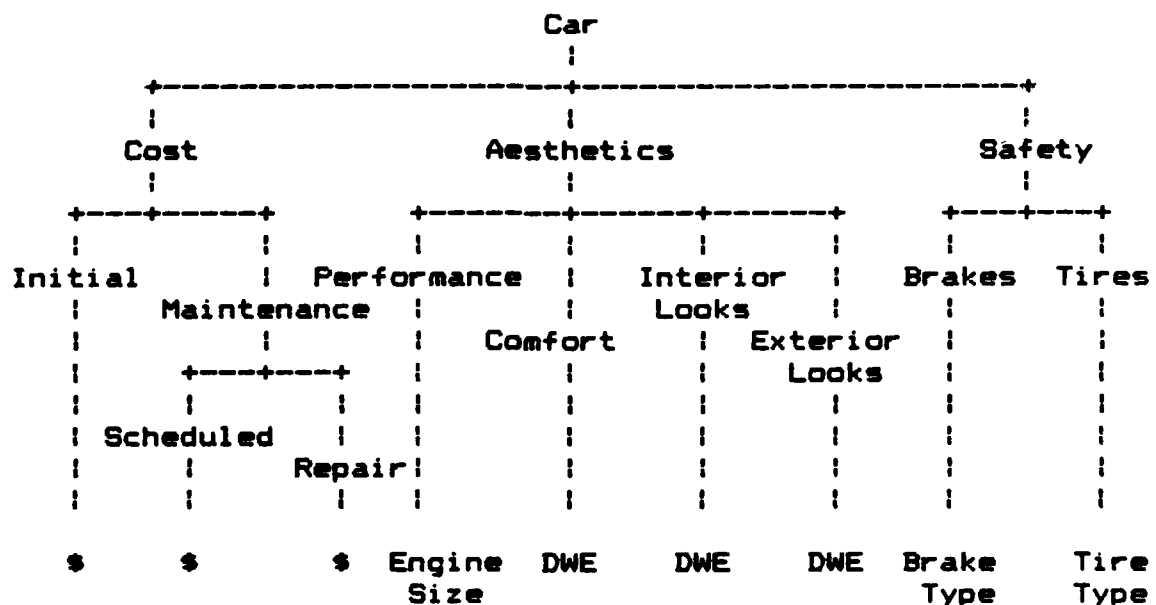


Figure 2.2. Car Hierarchy with Defined Characteristics

4 Define Worth Functions for Each Attribute. The next step in the worth assessment procedure is to develop worth or value functions for each of the attributes or

subobjectives. The worth function is used to measure (usually on a scale of 0 to 1) the degree of attainment for a given alternative and attribute (Sage, 1977: 357). The methods for calculating the worth function include discretely, linearly, and with the midvalue splitting technique (for nonlinear value functions).

The discrete method of defining a value function looks at each of the possible outcomes that an attribute can assume and assigns a worth to each outcome with respect to the other outcomes. An excellent discrete example is the worth function for the type of brakes in the car buying example. In that example, there are only three possible outcomes for types of brakes. These outcomes are disc brakes which were assigned a worth of 1.00, drum brakes which were assigned a .70, or, a combination of the two which was valued at .80.

The linear method calculates worth values by deriving a linear function using the attribute levels defined by the decision maker for values of 0 and 1. An example of the linear method can be seen in the worth function development of the scheduled maintenance costs for the car buying example. In this example, the buyer felt that a scheduled cost of \$228 was worth 0 in value while a scheduled cost of \$141 had a value of 1. To find the worth of the scheduled cost attribute for a given alternative, the buyer would use the equation below (Eq 2.1).

$$MWORTH = \frac{MAXATT - ATTLVL}{MAXATT - MINATT} \quad (2.1)$$

where,

MWORTH = worth associated with given attribute level
 MAXATT = attribute level associated with the maximum value (\$228)
 MINATT = attribute level associated with the minimum value (\$141)
 ATTLVL = Attribute level associated with an alternative

The worth (MWORTH) for a given scheduled cost of \$202 would be .70.

The midvalue splitting technique can be used when the attribute worth function is not linear. Listed below are the steps necessary to develop a function using the midvalue splitting technique.

- 1) Find the attribute levels associated with the minimum and maximum value -- 0 and 1, respectively.
- 2) Find the attribute level associated with a .5 value.
- 3) Next find the attribute levels associated with the new midpoint values -- .75 and .25 (Keeny and Raiffa, 1976: 94-96).

This process can continue by finding the midpoints between the newly set values (the next iteration would include attribute levels associated with .875, .625, .375, and .125), however, five points are usually enough to define a value function (DeWispelare, 1982).

5 Establish Relative Importance Between the Subobjectives (Sage, 1977: 357-358). The next step in the worth assessment procedure is to assign weights to the

subobjectives or attributes. The weight assignment procedure can be accomplished by following the steps below.

- 1) For a given level and set of objectives, the most important objective will be assigned a weight of 1.0.
- 2) Next, the remaining objectives are assigned weights relative to the most important objective.
- 3) After all the objectives in a given level and set are rated, the weights of the objectives are scaled to sum to one. Equation 2.2 shows how the objectives are scaled.

$$\text{Weight}_i = \frac{B_i}{B_1 + B_2 + B_3 + \dots + B_j} \quad (2.2)$$

where,

Weight_i = scaled weight for the ith objective
 B_i = raw weight for the ith objective
 B₁ + B₂ + ... = sum of all the raw weights of the j objectives for a given level and set

For the car buying example, one of the levels and sets would be cost, aesthetics, and safety. If the raw weights assigned to this level and set were 1.0, .6, and .4 for cost, aesthetics, and safety, respectively, the scaled weights would be .5, .3, .2. The next set would consist of the initial and maintenance costs. This process would continue until all the objectives and subobjectives are weighted.

After all of the objectives and subobjectives are weighted the overall weight associated with a given attribute can be calculated by multiplying the weights of

the objective and subobjectives that lead (via the tree diagram) to that attribute. For example, if the scaled weights for cost, maintenance cost, and repair cost are .5, .4, and .75, the overall weight for the subobjective of repair cost would be $(.5)*(.4)*(.75)$ or .15.

Figure 2.3 represents the car buying hierarchy with the scaled weights for the objectives and subobjectives, and the overall weights for the attributes. It should be noted that the sum of the overall weights must always equal one.

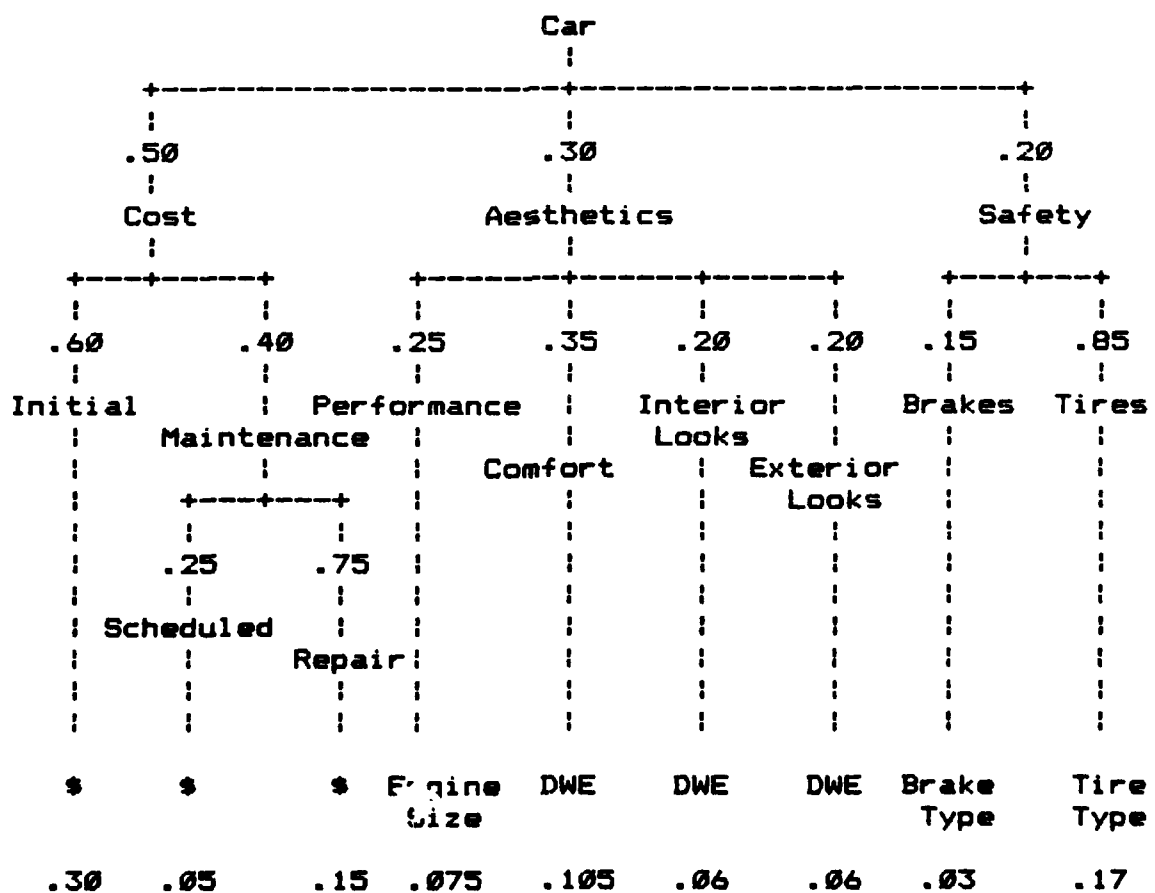


Figure 2.3 Weights for the Car Buying Scenario

6 Adjust the Weights to Reflect Confidence in the Measures (Sage, 1977: 358). After the weights for the objectives and subobjectives have been scaled and the overall weights have been defined for the attributes, the overall weights can be adjusted to reflect the confidence that the decision maker has in the attributes. For example, in the car buying model the attributes used in describing safety (brakes and tires) may not completely define safety, so that the decision maker's confidence measure would be lower for safety than for cost, which is not as abstract a concept and can be more easily defined. Caution should be used in applying a confidence measure because there is a possibility of "double counting" with the decision maker already accounting for his lack of confidence when he initially places weights in the model.

Adjusting the weights to reflect the decision maker's confidence is similar to the initial weighting scheme. The decision maker's interest is in the physical characteristics that are performance measures in the model. In the car model the performance measures include such things as initial cost, scheduled cost, performance, comfort, and brakes. The decision maker looks at all of the performance measures and assigns a confidence weight to each. If the decision maker feels that the performance measure is an accurate measure, he would assign a confidence weight of 1.0 to it, while he would assign a confidence weight of 0 to a performance

measure that is totally inaccurate. Any performance measure that falls between these two extremes would receive an appropriate confidence weight.

Once the confidence weights are assigned, the weights on the performance measures are rescaled using Equation 2.3.

$$PMWGHT_i = \frac{C_i W_i}{C_1 W_1 + C_2 W_2 + C_3 W_3 + \dots + C_j W_j} \quad (2.3)$$

where,

PMWGHT_i = Adjusted weight for the ith performance measure
 $C_i W_i$ = (Confidence Weight * Overall Weight) for the ith performance measure
 $C_1 W_1 + \dots$ = Sum of $C_i W_i$ s

7 Worth Assessment. After the performance measure weights have been calculated, the final step involves finding the "best" alternative (the alternative with the highest worth) by substituting the values of the attribute levels associated with a given alternative into the worth assessment equation (Eq 2.4) to find the overall worth of that alternative.

$$ALTWTH = \sum_i ATTWGT_i * VALLVLi \quad (2.4)$$

where,

ALTWTH = Alternative worth
 ATTWGT_i = Weight associated with ith performance measure (Weight_i or PMWGHT_i)
 VALLVLi = Value for the level of the ith performance measure associated the alternative

In the car buying example, the first step includes

finding the levels for each performance measure (cost, tires, etc.) for each car of interest. Next, the value associated with each of these levels is found, and finally, the worth of each car is found substituting the value and weight associated with each performance level into Equation 2.4. The "best" car will be the car with the highest worth score.

III. ALGORITHM IMPLEMENTATION FOR SWCS

The Environment

The scenario used in developing this algorithm consisted of a database file (representing the SAC base file) for ten hypothetical Air Force bases that were under threat from a missile -- either an ICBM or SLBM. Each of the bases could be in one of four alert statuses. The purpose of the algorithm developed was to maximize the number of aircraft that could escape safely while minimizing the cost involved with maintaining an alert status. For this scenario, the hypothetical decision maker considered the aircraft escaping attribute three times as important as the cost of an alert status attribute, and had confidence weights of 1.0 in the performance measures for these attributes.

The Bases

The bases used in this scenario were assigned impact times that were dependent upon one uniformly distributed randomly generated impact time. These impact times represented the flight time for a missile to reach the base (ground impact) from its indicated location. The impact times used for this scenario ranged from 300 seconds to 1800 seconds, reflecting the possible minimum flight times for SLBMs and maximum flight times for ICBMs. An impact time was randomly generated and assigned to the first base in the database file. Each succeeding base was then assigned an

impact time that was fifteen seconds greater than the preceeding base. For example, if an impact time of 1000 seconds was randomly generated, it would be assigned to the first base. The second base would have an impact time of 1015 seconds, and the third base would have an impact time of 1030 seconds. This impact time assignment would continue to the final base which would have an impact time of 1135 seconds. The impact times were generated to substitute for impact times that SWCS would produce. This algorithm was not concerned with improving impact time generation in SWCS -- only the decision making process associated with the time.

The Aircraft

Each base in this scenario had a differing amount of maximum aircraft. A formula using impact time, decision time, alert time, and spacing time was used to determine the number of aircraft that could escape safely. For this scenario, an aircraft was considered to have escaped safely if it could takeoff before a missile impact. Listed below are the definitions for each of these variables.

Impact time - The time (including warning time) it takes for a missile to travel from its launch location to a specific base.

Decision time - The time it takes for the decision maker to release the aircraft. For example, if the decision maker were at HQ SAC, the time it would take to release the aircraft would be less than if he were at another base because of the time involved with contacting him and receiving an answer.

Alert time - The time it takes the first aircraft to

prepare for takeoff because of the alert status. This time represents the "start up" time that is associated with a particular alert status.

Spacing time - The time interval between aircraft as they takeoff. This time reflects factors such as weather at the base or maintenance on runways.

Again, the impact time could vary between 300 seconds and 1800 seconds. The decision time could vary from 10 seconds to 300 seconds. The lower bound on this interval was used to represent a decision maker making an instantaneous decision (being readily available when the launch decision was needed), while the upper bound represented a decision maker not readily available (for example, a TDY). The alert times used were 600 seconds for the lowest alert posture to 10 seconds for the highest alert status. Finally, the bounds on the spacing time were 15 seconds for the minimum spacing distance and 60 seconds for the worst possible conditions.

The number of aircraft that could escape was derived by subtracting the decision time and alert time from the impact time, then dividing by the spacing time (Eq 3.1).

$$\text{Esc A/C} = \text{INT} \left[1 + \frac{\text{Impct tme} - \text{Dcsn tme} - \text{Alrt tme}}{\text{Spcing tme}} \right] \quad (3.1)$$

If the Escaping Aircraft was less than zero, Escaping Aircraft was set to zero. If Escaping Aircraft was greater than the maximum number associated with a given base,

Escaping Aircraft was set to the maximum number of aircraft associated with that base. Once the number of aircraft that could escape was established, the percentage of aircraft that could escape with respect to the maximum aircraft at a given base was calculated. This percentage was then used to find the value of the number of aircraft escaping.

The Alert Status

In this scenario, four alert postures were used. The first posture or status was the least costly and lowest in terms of readiness, and the fourth status was the most expensive and the highest alert status. Listed below are descriptions of each alert status and the time (alert time) in seconds for the first aircraft to prepare for takeoff because of that status.

Alert Status	Description	Alert Time
1	Lowest status, normal duty, crew not confined to alert facility.	600
2	Crew confined to alert facility.	200
3	Crew in aircraft with engines running.	60
4	Aircraft prepared to takeoff.	10

The cost for each alert status is directly associated with the status. For example, the lowest alert status would be the least expensive with the cost increasing as the alert status increases. The alert status for a given base is used to determine the value of the cost for the alert status.

The Value Functions

The value functions used in this scenario were a function of the percentage of the maximum aircraft that could safely escape, and the cost of a given alert status or posture.

The value function (Fig 3.1) for the percentage of aircraft safely escaping a given base was a linear relationship with 100% safely escaping receiving a value of 1.0, and no aircraft escaping receiving a value of 0.0.

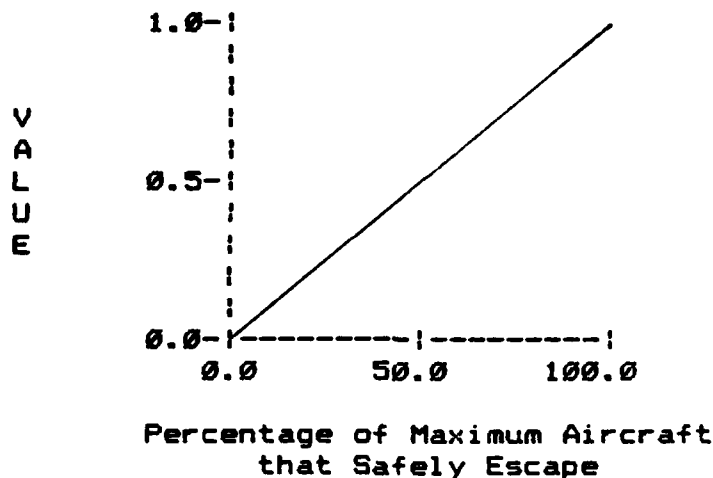


Figure 3.1. Value Function for Aircraft Escaping

The value function for the cost of the alert postures (Fig 3.2) was determined with respect to the lowest alert status. The lowest alert status had a value of 1.0, and the second lowest alert status was assigned a .8 because of the relatively small increase in cost. The next alert status, however, was given a value of .4 because of the greater usage of resources. Finally, the fourth and highest alert

status had a value of .1 because of all the preparation involved.

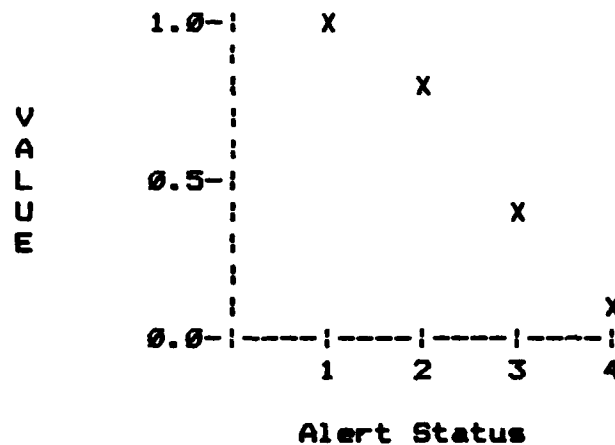


Figure 3.2. Value Function for Alert Status

For this scenario, these cost values were independent of the number of aircraft located at a given base because the cost of every alert status is proportional to the number of aircraft. If one base had ten aircraft and another base had twenty aircraft, the cost to maintain a given alert status was assumed to be half as much for the base with ten aircraft because only half as much of the resources would be used. The value of the cost between alert postures, however, would remain the same for the two bases (e.g. .8 for the second alert status) because the relative resource usage between alert statuses at each base is the same.

The Algorithm

The algorithm used in this scenario proposed an alert status for each base in the database file. The alert status

was determined by calculating a value (between 0.0 and 1.0) for each possible alert status for that base (Eq 3.2).

$$\text{Alert Worth} = \sum \text{Attribute weight} * \text{Attribute value} \quad (3.2)$$

For this scenario, the alert worth combined a weighted worth of the percentage of aircraft escaping with a weighted worth of the alert cost. The alert status that corresponded with the highest value was the proposed alert status.

The Inputs to SWCS

This scenario was designed as an example. The user should develop a scenario to fit his value function, however, if this algorithm were to be used in SWCS, the following changes would have to be made. First, the database file would have to be changed to reflect the actual bases and the number of aircraft located at each base. Next, the alert status would have to show all possible postures and their associated times for the first aircraft to takeoff. Value functions would also have to be designed to model the decision maker's values for the percentage of aircraft that can escape safely and the cost for each alert status. One method for deriving value functions will be presented in the software implementation portion of this manual. Finally, the decision maker's preference between aircraft escaping and alert cost would have to be measured.

The inputs used in this algorithm include the impact time of the missile, the decision time, the alert time, and the spacing time. With the impact time of the missile currently provided, the only modification to the current software would be to allow for input of decision time, alert time and spacing time. This modification would be relatively simple because the alert times could be set to reflect the mean time to takeoff of the first aircraft for each alert posture. The decision time could be an input that is prompted anytime a change is necessary, and the same decision time would apply to each base. The spacing time would have to be initialized for each base and thereafter be changed daily or when the need arises on a base by base case.

IV. SOFTWARE IMPLEMENTATION

Overview

The worth assessment program presented assumes that the user is concerned with identifying an optimal alert status for each base in the database file. The model designed in this thesis addresses one base at a time looking at all the possible alert postures for that base. The values associated with the percentage of aircraft that can escape from that base and the cost for each alert posture are derived. Once these values are found, they are then substituted in the decision maker's overall value function to find the value for that given base-alert status combination. This value is then checked to see if the value calculated for the alert status is the highest score for that particular base. After all the alert postures are addressed for the base, the proposed alert status is that alert status with the highest value. The output for this algorithm would be the name of the base, the projected impact time of the missile, and the proposed alert status.

Implementation

Listed below are the necessary steps needed to implement the worth assessment algorithm that has been presented. Figure 4.1 is a flowchart to assist in the implementation process.

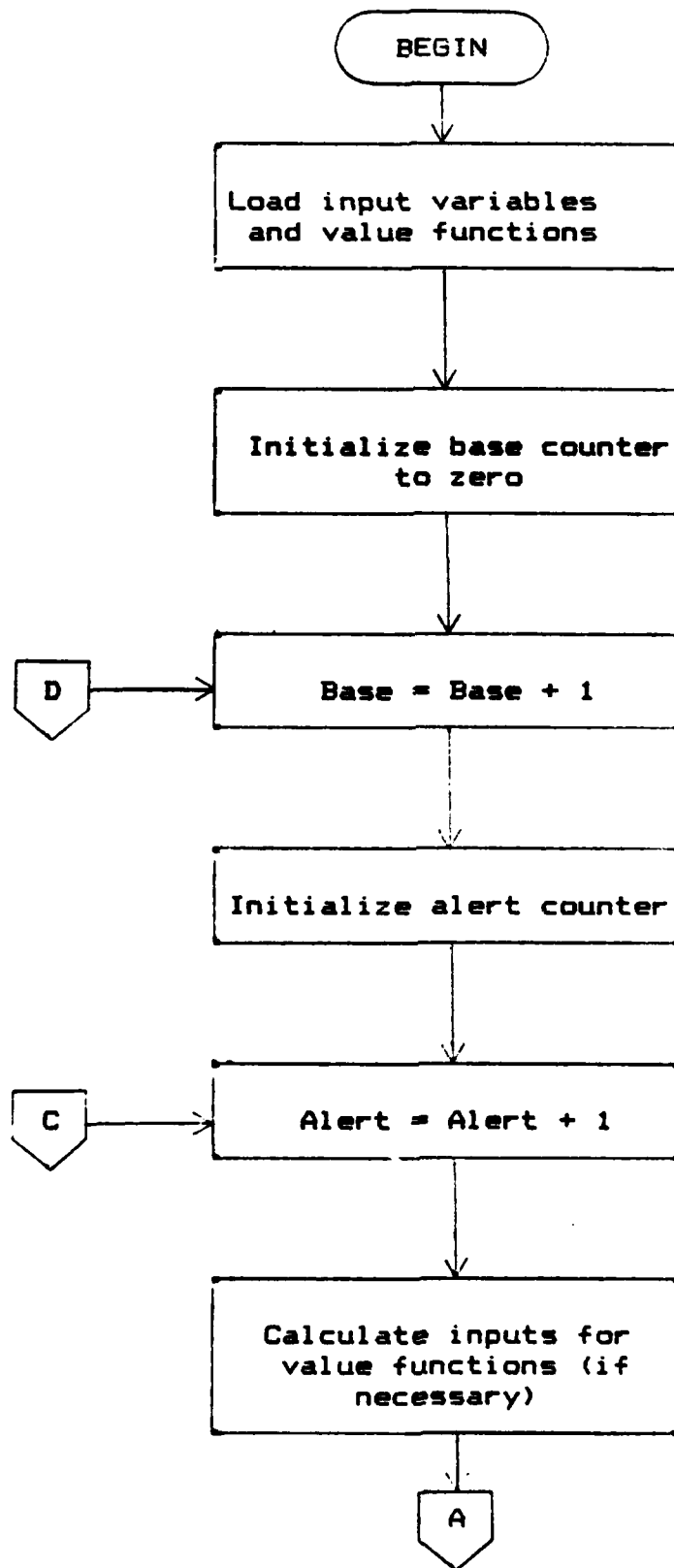


Figure 4.1. Flowchart for Worth Assessment

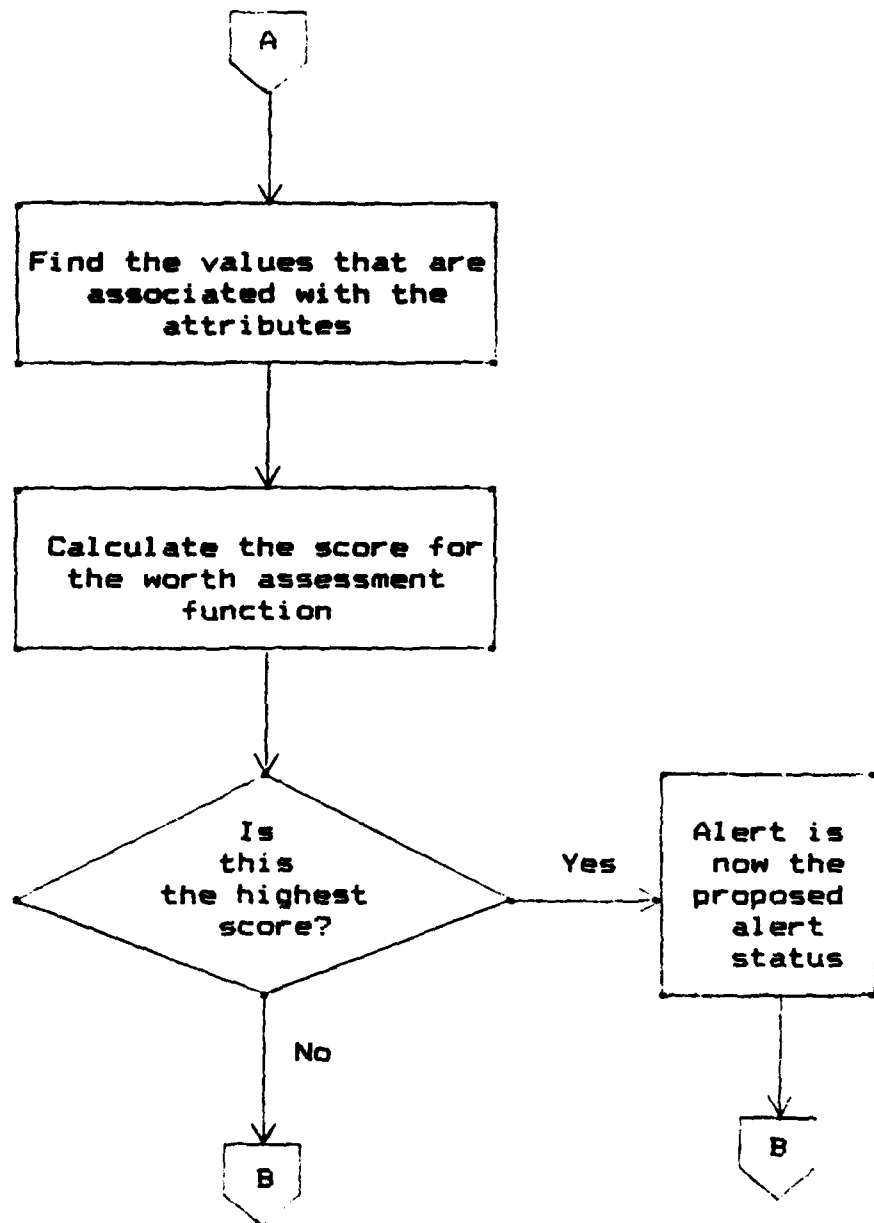


Figure 4.1. Flowchart for Worth Assessment (Cont)

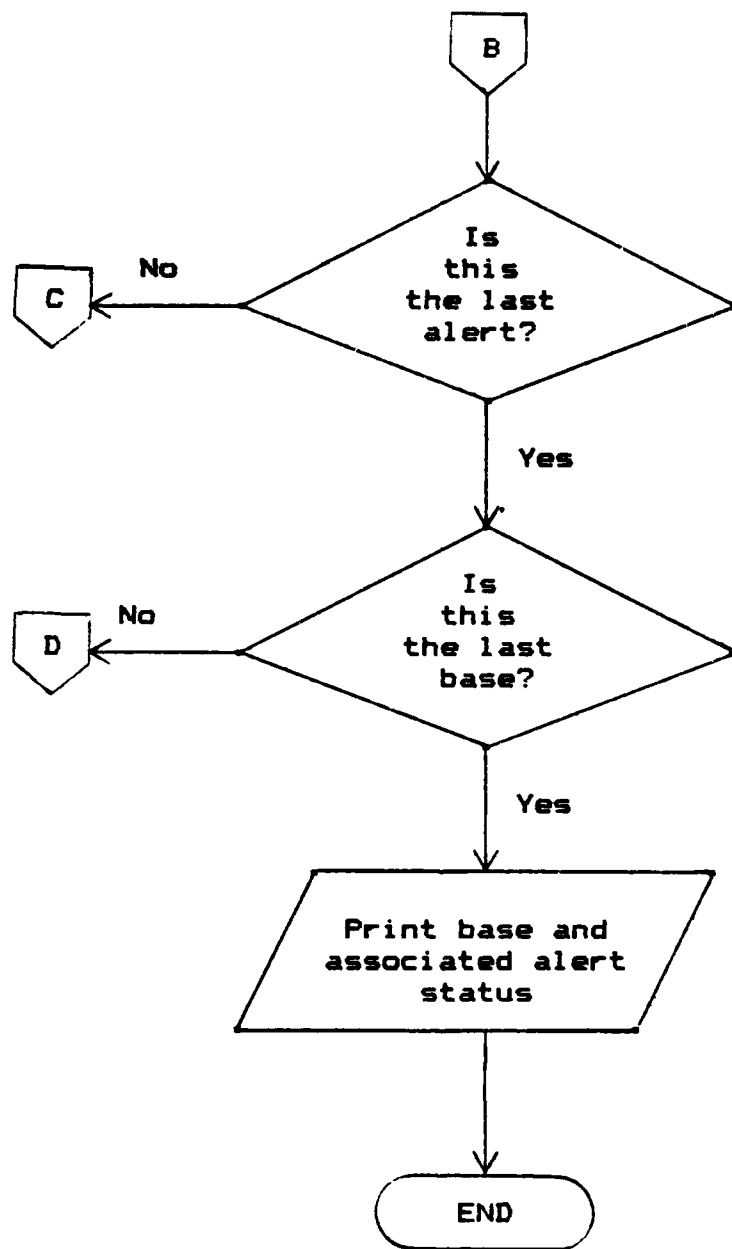


Figure 4.1. Flowchart for Worth Assessment (Cont)

1) The first step involved with implementing the worth assessment algorithm is to find the decision maker's overall value function in assessing alert postures for his aircraft. This can be accomplished by following the worth assessment procedure in Section II of this manual.

2) Next, the value functions for the individual attributes should be set up in subroutines. In the scenario, these attributes were the percentage of aircraft escaping and the cost of the alert. Ways to design these functions are addressed in Chapter II.

3) After the subroutines are created, the next step is to build the main program. The first step in the main program consists of loading in the necessary data required by the decision maker's overall value function. In the scenario that was presented, the reaction times due to the alert postures, the values associated with alert costs, the number of aircraft stationed at each base, and the value function for percentage of aircraft escaping were the data loaded at this point in the program.

4) Once the data is loaded, a nested DO-LOOP is constructed. The outer loop is the counter for the bases, and cycles through the base database file. The inner loop is the alert counter used to calculate the decision maker's worth for each alert status for a given base.

5) The first block inside of the nested DO-LOOP is for making the calculations needed for any of the attributes. Finding the percentage of aircraft that could escape within a missile's flight time is an example of a calculation that would be in this block.

6) The next block finds the values associated with the given levels of each of the attributes. In the scenario, it was in this block that the values for percentage of aircraft escaping and for alert cost were found.

7) After the values for each attribute are found, they are then used in the decision maker's worth function to find the value for the given alert posture and base.

8) The next step involves checking the value of the worth function for the alert posture calculated in the last step against that of the alert posture currently having the highest score. If the value of the alert status just calculated has a value that is greater than the current high score, it is then made the proposed alert status and its score becomes the current high score. If there is a tie, the lowest alert posture will be the proposed alert status.

9) The nested DO-LOOP is now closed. The results will be a proposed alert status (the alert status with the

highest worth value) for each of the bases in the database file.

10) The final step in this program involves passing the data that has been calculated. In the scenario developed this data was printed out; however, in SWCS, this data can be passed to other programs for further computations.

V. PROGRAM SOURCE CODE

On the following pages (31-43) is the source code used to implement the scenario presented on an IBM Personal Computer. This software was programmed using IBM PC Advanced BASIC Version 1.00. Listed below is a brief description of what each section of the program does with that section's corresponding lines.

Line Numbers	Description
1950 - 1970	Dimension arrays
2040	Go to subroutine to load values for the variables used in the worth assessment function
2090	Go to subroutine to load in the databases that will be used in the algorithm
2140	DO-LOOP that cycles through the bases of interest
2200	DO-LOOP that cycles through the possible alert postures
2250 - 2350	Calculate the number of aircraft that can

escape from a given base and alert status

2410 Find the value associated with the percentage
of aircraft that can escape

2460 Find the value associated with the cost of
the given alert posture

2520 Find the value of the alert posture

2570 - 2590 Check to see if the value calculated is the
highest value for an alert posture thus far

2710 Print out the results showing the base and
its proposed alert status

2860 - 2890 Initialize the reaction time for the aircraft
due to each alert status

2970 - 3000 Initialize the values associated with the
cost of each alert status

3060 - 3150 Initialize the maximum number of aircraft
located at each base

3200 - 3220 Initialize the maximum score of the alert
postures for base

3320	Function that calculates a value for the percentage of aircraft that can escape
3400	Function that calculates a value for the cost of a given alert status
3500	Function that calculates the overall worth for an aircraft escaping -- alert cost combination
3610	Set the seed for the random number generator
3680 - 3710	Impact time random number generator
3760 - 3780	Spacing time generator
3850	Decision time generator
3960 - 4100	Print subroutine

```

1000 REM *****
1010 REM *
1020 REM Program: WADIM *
1030 REM *
1040 REM Programmer: Doug Lee *
1050 REM Date: 7 Nov 82 *
1060 REM Language: IBM PC Advanced BASIC Version 1.00 *
1070 REM *
1080 REM *
1090 REM This algorithm is a worth assessment model *
1100 REM designed for use with the SAC Warning and Control *
1110 REM System (SWCS). This worth assessment model is *
1120 REM concerned with maximizing the percentage of *
1130 REM aircraft that can escape in a threatening *
1140 REM environment from a given base while minimizing *
1150 REM the cost of the alert status. The scenario uses *
1160 REM the impact time of the missile (threat), the time *
1170 REM to make the decision to launch the aircraft, the *
1180 REM time for the first aircraft to prepare for takeoff*
1190 REM from a given alert status, and the time interval *
1200 REM between takeoffs for calculating the number of *
1210 REM aircraft that can escape. The costs of the *
1220 REM alert statuses were determined relative to the *
1230 REM lowest alert status. Also, in this scenario, *
1240 REM aircraft escaping was considered three times as *
1250 REM important as the cost of the alert status. *

```

```

1260 REM Finally, ten bases and four alert statuses were *
1270 REM used. *
1280 REM *
1290 REM *
1300 REM *
1310 REM *
1320 REM Listed below are the variables, arrays, and *
1330 REM functions used in this program with their *
1340 REM associated meanings. *
1350 REM *
1360 REM Variables: *
1370 REM *
1380 REM Name          Meaning                      Unit *
1390 REM *
1400 REM ALERT        Counter for alert status      n/a *
1410 REM BASE         Counter for bases             n/a *
1420 REM DECTME       Amount of time to decide to   *
1430 REM              launch aircraft              sec *
1440 REM NAC          Temporary storage for number of *
1450 REM              aircraft that escape safely   a/c *
1460 REM VACESC       Worth value for number of aircraft *
1470 REM              escaping                      n/a *
1480 REM VALTCT       Worth value for the cost       *
1490 REM              associated with an alert status n/a *
1500 REM *
1510 REM *
1520 REM *

```

1530	REM Arrays:		*
1540	REM		*
1550	REM Name	Meaning	Unit *
1560	REM		*
1570	REM ACESC(*)	Number of aircraft that escape	*
1580	REM	safely	a/c *
1590	REM ALTPAD(*)	Amount of time before first	*
1600	REM	aircraft can prepare for takeoff	*
1610	REM	due to alert status	n/a *
1620	REM ALTSCR(*)	Score or value of the combined	*
1630	REM	alert status and number of	*
1640	REM	aircraft that can escape	n/a *
1650	REM ALTVAL(*)	Value associated with each alert	*
1660	REM	status	n/a *
1670	REM MAXAC(*)	Maximum number of aircraft that	*
1680	REM	are located at a given base	a/c *
1690	REM MAXALT(*)	Alert status associated with the	*
1700	REM	current maximum score	n/a *
1710	REM MAXSCR(*)	Current maximum score	n/a *
1720	REM SPCPAD(*)	Spacing interval in time between	*
1730	REM	aircraft taking off	sec *
1740	REM TMEIMP(*)	Time to impact for missile	sec *
1750	REM		*
1760	REM		*
1770	REM		*
1780	REM Functions:		*
1790	REM		*

1800 REM Name	Meaning	Unit	*
1810 REM			*
1820 REM FNVAC	Calculates the value associated		*
1830 REM	with percentage of aircraft that		*
1840 REM	that can takeoff	n/a	*
1850 REM FNVALT	Calculates the value associated		*
1860 REM	with cost of a given alert status	n/a	*
1870 REM FNACEC	Calculates the value associated		*
1880 REM	with overall worth of a given		*
1890 REM	number of aircraft escaping and		*
1900 REM	alert cost	n/a	*
1910 REM			*
1920 REM			*
1930 REM	*****		
1940 REM			
1950	DIM ACESC(10),ALTPAD(10),ALTSCR(10),ALTVAL(10)		
1960	DIM MAXAC(10),MAXALT(10),MAXSCR(10),SPCPAD(10)		
1970	DIM TMEIMP(10)		
1980 REM	*****		
1990 REM	* This portion of the program is used to *		
2000 REM	* initialize the variables that would normally be *		
2010 REM	* passed to this program from SWCS *		
2020 REM	*****		
2030 REM			
2040	GOSUB 3520		
2050 REM			
2060 REM	Load in the data bases that will be used with this		

```

2070 REM scenario
2080 REM
2090     GOSUB 2730
2100 REM
2110 REM Cycle through the base file to determine the
2120 REM proposed alert status for each base
2130 REM
2140     FOR BASE = 1 TO 10
2150         REM
2160         REM Check each alert status for the highest value,
2170         REM making the alert status with the highest score
2180         REM the proposed alert status
2190         REM
2200             FOR ALERT = 1 TO 4
2210                 REM
2220                 REM Calculate the number of aircraft that can
2230                 REM escape within the impact time
2240                 REM
2250                     NAC = TMEIMP(BASE) - DECTME - ALTPAD(ALERT)
2260                     NAC = NAC / SPCPAD(BASE)
2270                     NAC = INT(1 + NAC)
2280                 REM
2290                 REM Check to see if the number of aircraft that
2300                 REM can escape is less than or exceeds the number
2310                 REM allowed
2320                 REM
2330                 IF NAC < 0 THEN NAC = 0

```

```

2340      IF NAC > MAXAC(BASE) THEN NAC = MAXAC(BASE)
2350      ACESC(BASE) = NAC
2360      REM
2370      REM Find the value for the number of aircraft
2380      REM that can escape for the associated alert
2390      REM status
2400      REM
2410      VACESC = FNVAC(ACESC(BASE),BASE)
2420      REM
2430      REM Find the value for the cost of the associated
2440      REM alert status
2450      REM
2460      VALTCT = FNVALT(ALERT)
2470      REM
2480      REM Find the total value of the number of
2490      REM aircraft escaping with the associated alert
2500      REM cost
2510      REM
2520      ALTSCR(BASE) = FNACEC(VACESC,VALTCT)
2530      REM
2540      REM Check to see if this score is the current
2550      REM maximum score
2560      REM
2570      IF ALTSCR(BASE) <= MAXSCR(BASE) THEN GOTO 2600
2580      MAXSCR(BASE) = ALTSCR(BASE)
2590      MAXALT(BASE) = ALERT
2600      NEXT ALERT

```

```

2610      NEXT BASE
2620 REM
2630 REM * * * * *
2640 REM * This portion of the program prints out the *
2650 REM * impact time, decision time, alert reaction time,*
2660 REM * spacing interval, aircraft escaping, maximum *
2670 REM * aircraft, and proposed alert status for each *
2680 REM * base *
2690 REM * * * * *
2700 REM
2710      GOSUB 3870
2720      END

```

```

2730 REM ***** DATA LOADING SUBROUTINE *****
2740 REM This subroutine loads in the data that will be used
2750 REM in this algorithm
2760 REM
2770 REM
2780 REM ALTPAD(Alert status) represents the time ,in
2790 REM seconds, for an aircraft to prepare for takeoff
2800 REM due to the delay in readying the aircraft from a
2810 REM specific alert status. The lowest alert status
2820 REM (1) will take the longest amount of time while
2830 REM the highest alert status (4) will take the
2840 REM shortest amount of time
2850 REM
2860 ALTPAD(1) = 600
2870 ALTPAD(2) = 200
2880 ALTPAD(3) = 60
2890 ALTPAD(4) = 10
2900 REM
2910 REM ALTVAL(Alert status) is the value of the cost
2920 REM for each alert status with the lowest alert
2930 REM status (1) being the most valuable (1) and the
2940 REM highest alert status (4) being the most
2950 REM expensive and lowest value (.1)
2960 REM
2970 ALTVAL(1) = 1
2980 ALTVAL(2) = .8
2990 ALTVAL(3) = .4

```

```

3000      ALTVAL(4) = .1
3010      REM
3020      REM MAXAC(Base) represents the maximum number of
3030      REM aircraft allocated to each of the bases in the
3040      REM base file
3050      REM
3060      MAXAC(1) = 20
3070      MAXAC(2) = 30
3080      MAXAC(3) = 25
3090      MAXAC(4) = 35
3100      MAXAC(5) = 20
3110      MAXAC(6) = 23
3120      MAXAC(7) = 32
3130      MAXAC(8) = 37
3140      MAXAC(9) = 28
3150      MAXAC(10) = 22
3160      REM
3170      REM Initialize the maximum score MAXSCR(Base)
3180      REM associated with each base
3190      REM
3200      FOR BASE = 1 TO 10
3210          MAXSCR(BASE) = -1
3220      NEXT BASE
3230      REM
3240      REM FNVAC(Aircraft,Base) is a function that
3250      REM calculates the value of the percentage of the
3260      REM maximum aircraft that can escape from the given

```

```

3270 REM base. For this scenario, the values for
3280 REM escaping aircraft were assigned using a linear
3290 REM function with 100% getting a value of 1 and 0%
3300 REM escaping getting a value of 0
3310 REM
3320 DEF FNVAC(AC,B) = AC / MAXAC(B)
3330 REM
3340 REM FNVALT(Alert status) is a function that
3350 REM determines the value of the alert cost for each
3360 REM status. In this scenario, the values have been
3370 REM assigned with respect to the lowest alert status
3380 REM (giving the lowest alert status a value of 1)
3390 REM
3400 DEF FNVALT(A) = ALTVAL(A)
3410 REM
3420 REM FNACEC(Value of aircraft escaping, Value of
3430 REM alert status cost) is a function that determines
3440 REM the value of the overall worth function. In
3450 REM this scenario, the value for aircraft escaping
3460 REM is considered three times as important as the
3470 REM value of the alert cost. The worth function can
3480 REM range from 0 to 1.0
3490 REM
3500 DEF FNACEC(VAC,VCT) = .75 * VAC + .25 * VCT
3510 RETURN

```

```

3520 REM
3530 REM * * * * *
3540 REM * This portion of the program is used to *
3550 REM * initialize the variables that would normally be *
3560 REM * be passed to this program from SWCS *
3570 REM * * * * *
3580 REM
3590 REM Set the random number seed
3600 REM
3610     RANDOMIZE
3620 REM
3630 REM Generate an impact time (TMEIMP) between 5 and 30
3640 REM minutes (300 - 1800 seconds) for the first base,
3650 REM then add 15 seconds to the impact time of each
3660 REM succeeding base
3670 REM
3680     TMEIMP(1) = RND * 1500 + 300
3690     FOR BASE = 2 TO 10
3700         TMEIMP(BASE) = TMEIMP(BASE-1) + 15
3710     NEXT BASE
3720 REM
3730 REM Generate a spacing time (SPCPAD) between 15 and 60
3740 REM seconds for the aircraft at each base
3750 REM
3760     FOR BASE = 1 TO 10
3770         SPCPAD(BASE) = RND * 45 + 15
3780     NEXT BASE

```

3790 REM

3800 REM Generate a decision time (DECTME) between 10 and
3810 REM 300 seconds. This decision time will be the same
3820 REM for all bases because the decision to launch will
3830 REM be given to all bases simultaneously

3840 REM

3850 DECTME = RND * 290 + 10

3860 RETURN

```

3870 REM
3880 REM * * * * *
3890 REM * This portion of the program prints out the      *
3900 REM * impact time, decision time, alert reaction time,*
3910 REM * spacing interval, aircraft escaping, maximum    *
3920 REM * aircraft, and proposed alert status for each    *
3930 REM * base                                              *
3940 REM * * * * *
3950 REM
3960     LPRINT SPC(60)"Proposed"
3970     LPRINT "      Impact Decision Alert Spacing ";
3980     LPRINT "Aircraft Maximum      Alert"
3990     LPRINT "Base Time      Time      Time      Time ";
4000     LPRINT "Escaping Aircraft      Status"
4010     FOR BASE = 1 TO 10
4020         LPRINT USING " ##      ";BASE;
4030         LPRINT USING "####      ";TMEIMP(BASE);
4040         LPRINT USING "###      ";DECTME;
4050         LPRINT USING "###      ";ALTPAD(MAXALT(BASE));
4060         LPRINT USING "##      ";SPCPAD(BASE);
4070         LPRINT USING "##      ";ACESC(BASE);
4080         LPRINT USING "##      ";MAXAC(BASE);
4090         LPRINT USING "#";MAXALT(BASE)
4100     NEXT BASE
4110     RETURN

```

VI. SENSITIVITY ANALYSIS

The purpose of the sensitivity analysis presented in this chapter is to give the decision maker an opportunity to see how the parameters -- specifically, the weights assigned to the worth assessment function and the attribute level for a given alert posture -- affect the overall problem of finding an optimal or proposed alert status. The parameters addressed are the weights assigned to aircraft escaping and alert cost, and the value of the number of aircraft that can escape for a given alert status. The parameters are addressed independently (changing one parameter at a time and holding the other variables constant) to find the weight or level change in each that is necessary to make a non-optimal alert posture the proposed alert status. The program was designed to perform a sensitivity analysis on all of the alert postures that are below the proposed alert status. For example, if the proposed alert status was 3, a sensitivity analysis would be performed on alert status 2 and alert status 1.

Once the parameters necessary to make a non-optimal alert status the proposed alert status are defined, an indepth analysis is conducted on the value level of aircraft escaping. The first step in this analysis is to convert the aircraft escaping value into the number of aircraft that must escape. After the number is derived, it is then translated into time -- the change needed in both the

spacing time and a decision/alert time combination to attain the number of aircraft escaping.

Weight Sensitivity Analysis

The purpose of the weight sensitivity analysis is to find the needed change in an attribute weight in order for a given alert posture to become the proposed alert status. Equation 6.1 shows the inputs necessary to find an attribute weight that will change a given alert status to the proposed alert status.

$$\begin{aligned} \text{ATTWGT}_j' = & [\text{ALTWTH} - (\text{VALLVL}_j * \text{ATTWGT}_j) \\ & - \text{ALTWTH}' + (\text{ALTWTH}' * \text{ATTWGT}_j)] \quad (\text{Eq 6.1}) \\ & / [\text{ALTWTH} - \text{VALLVL}_j] \end{aligned}$$

where,

ATTWGT_j' = new weight associated with the j th attribute
 ATTWGT_j = old weight associated with the j th attribute
 ALTWTH' = proposed alert worth score
 ALTWTH = alert worth score associated with the current alert being addressed
 VALLVL_j = value for the level of the j th attribute.

In the hypothetical scenario presented, ATTWGT_j represents the weights associated with either aircraft escaping and alert cost while VALLVL_j is the corresponding attribute level. ALTWTH' is the worth score of the proposed alert status for a given base and ALTWTH represents the worth score for each of the alert postures of interest (the statuses that are lower than the proposed alert status). While Equation 6.1 adjusts one of the attribute weights, the remaining weights will also change because the equation retains the normalized weights of the decision tree.

Level Sensitivity Analysis

The level sensitivity analysis is used to determine value level of an attribute needed for a given alert status to become the proposed alert status (Eq 6.2).

$$VALLVL_j' = \frac{ALTWTH' - ALTWTH + (ATTWGT_j * VALLVL_j)}{ATTWGT_j} \quad (\text{Eq 6.2})$$

where,

$VALLVL_j'$ = new value for the jth attribute

$VALLVL_j$ = old value for the jth attribute

$ALTWTH'$ = proposed alert worth score

$ALTWTH$ = worth score associated with the current alert status

$ATTWGT_j$ = weight associated with the jth attribute.

When applying Equation 6.2 to the hypothetical scenario, only the attribute level associated with aircraft escaping is addressed because of the difficulty in interpreting the results of a sensitivity analysis using the alert cost value levels. While $ATTWGT_j$ is the weight associated with the aircraft escaping, $VALLVL_j$ is the value associated with a given alert status' aircraft escaping attribute, and $VALLVL_j'$ is the value needed in the aircraft escaping attribute for the alert status to become the proposed alert posture. As in Equation 6.1, $ALTWTH'$ is the worth score of the current proposed alert status and $ALTWTH$ is the worth score for the alert status being addressed.

Value Level Conversion

After the aircraft value level needed to make a given

alert status the proposed alert status is calculated, the level is then converted into the number of aircraft that must escape. This is accomplished by using the value function defined for aircraft escaping (Figure 3.1, Value Function for Aircraft Escaping). Next, Equation 3.1 is used to determine the spacing time required and then the decision time/alert time combination required to attain the needed number of aircraft to make a given alert status the proposed alert status.

User Implementation

Because the sensitivity analysis developed is dependent upon the hypothetical scenario, it would not be feasible to provide the exact steps used to implement this specific software; however, a flowchart, description of the software code, and listing of the software code are provided to help the decision maker design the software necessary to implement his sensitivity analysis. Figure 6.1 is the flowchart of the sensitivity analysis designed for the hypothetical scenario.

Sensitivity Analysis Flowchart

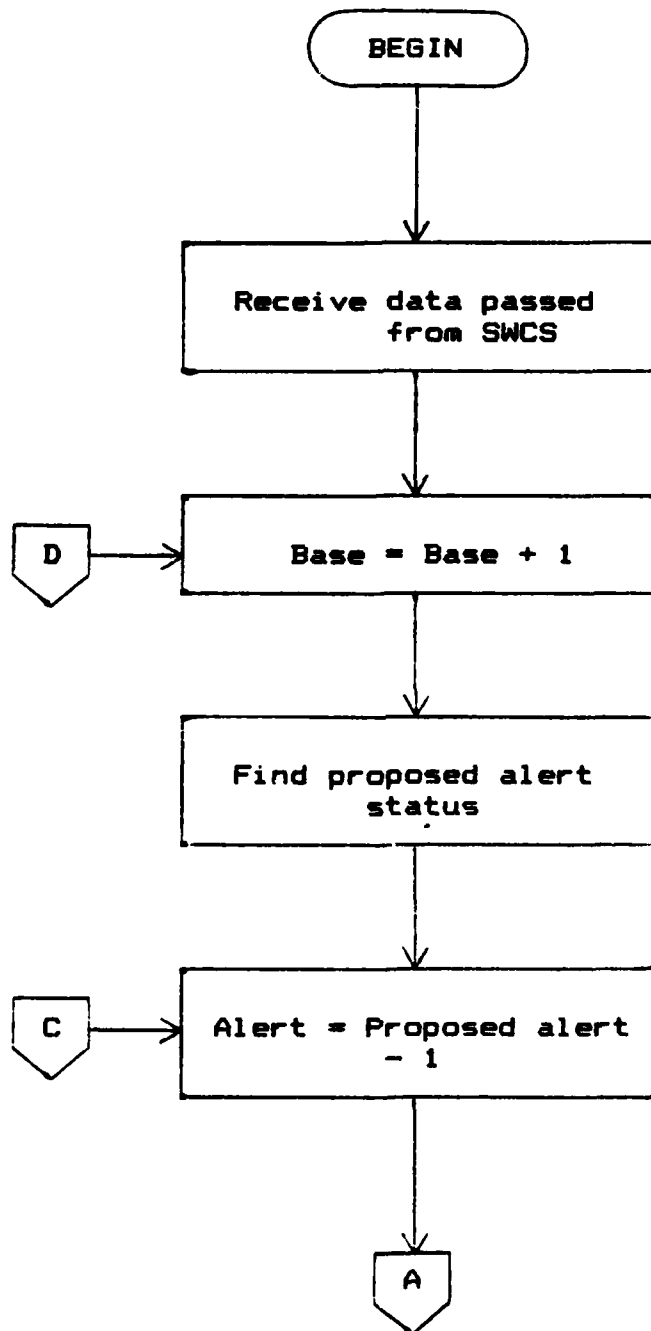


Figure 6.1. Flowchart for Sensitivity Analysis

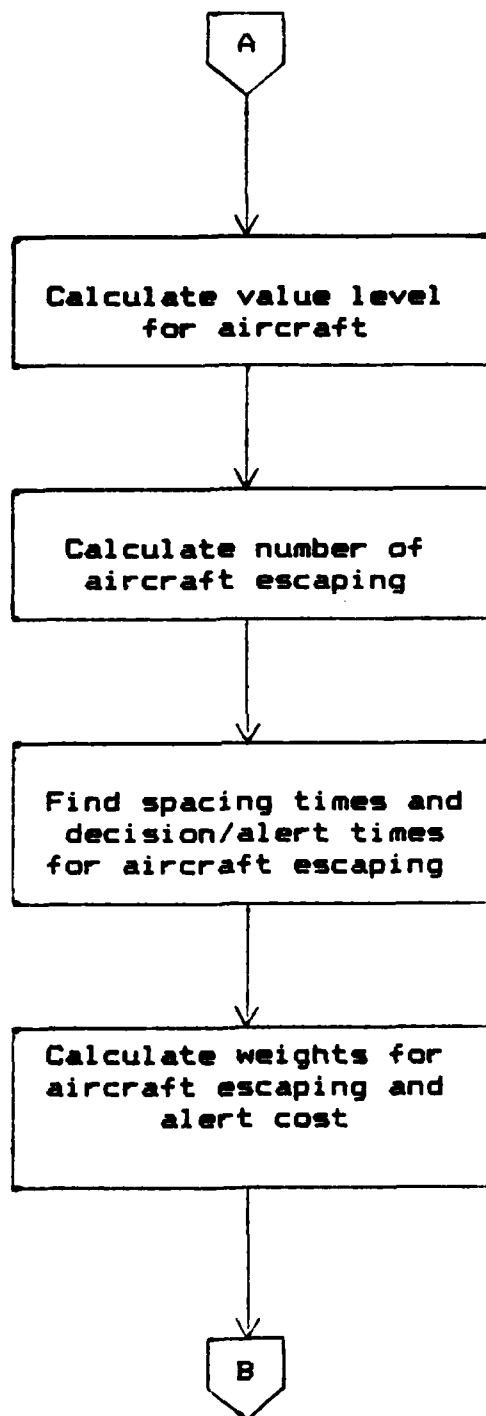


Figure 6.1. Flowchart for Sensitivity Analysis (Cont)

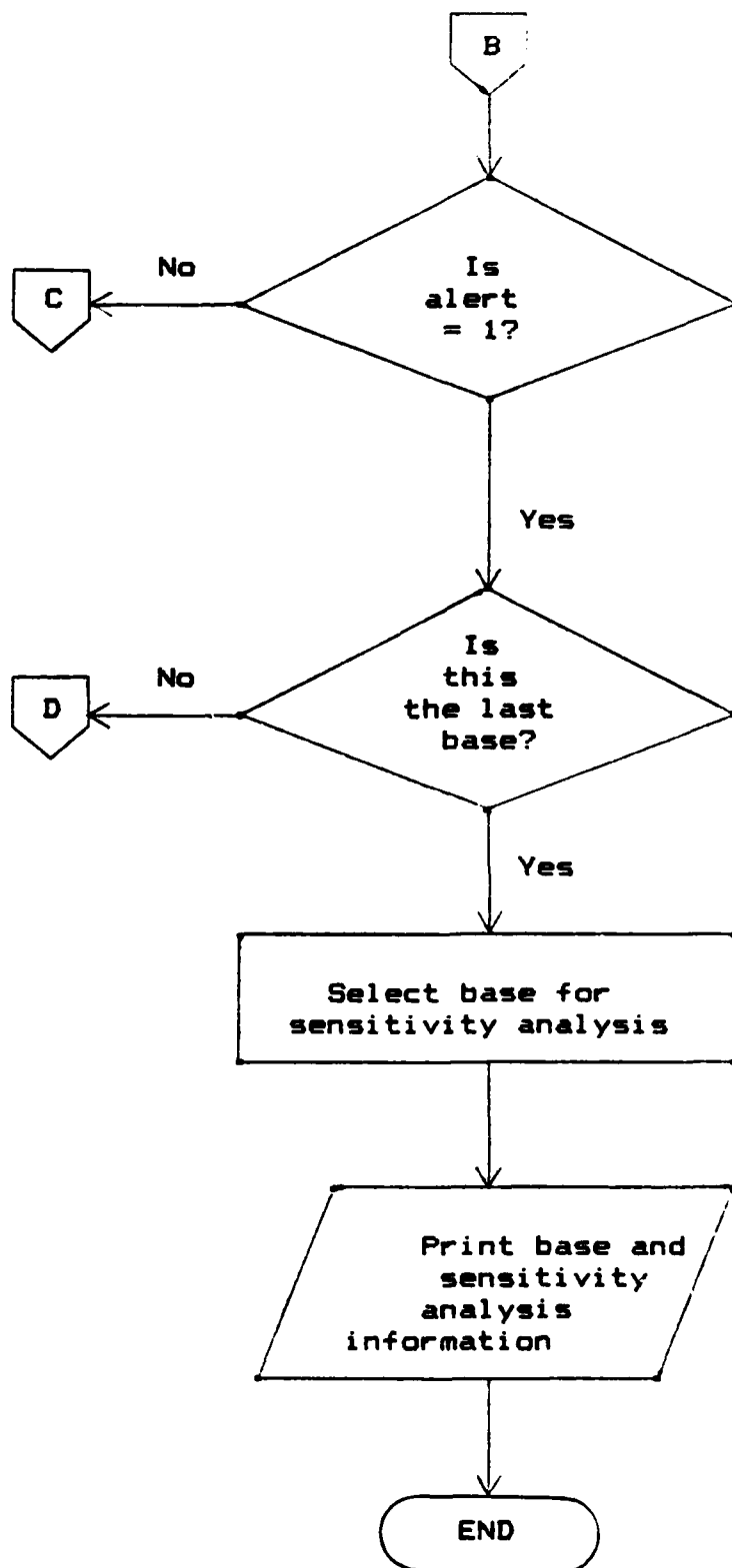


Figure 6.1. Flowchart for Sensitivity Analysis (Cont)

Program Source Code

On the following pages (53-71) is the source code used to implement the sensitivity analysis developed on an IBM Personal Computer. As with the worth assessment software, this sensitivity analysis software was programmed using IBM PC Advanced BASIC Version 1.00. Listed below is a brief description of what each section of the program does with that section's corresponding lines.

Line Numbers	Description
2220 - 2240	Dimension arrays
2310	Go to subroutine to receive data passed from SWCS
2360	Go to subroutine to find the proposed alert status for each base
2430	Go to subroutine to find value level for aircraft escaping for each alert status of interest
2480	Go to subroutine to convert value level into number of aircraft escaping
2540	Go to subroutine to calculate spacing time

AD-A124 730

A DECISION ANALYSIS ALGORITHM FOR THE SAC WARNING AND
CONTROL SYSTEM(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING D E LEE
DEC 82 AFIT/GOR/OS/82D-9

2/2

UNCLASSIFIED

F/G 12/1

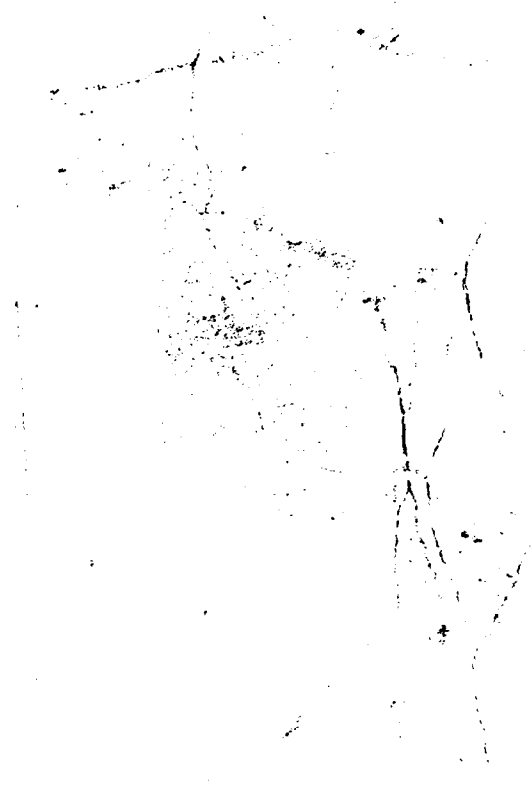
NL

END

FORMED

4

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

and decision/alert times needed to attain the
number of aircraft escaping

2600 Go to subroutine to find the weights
 necessary for each attribute in order for the
 alert status to become the proposed alert
 status

2680 Go to print subroutine

2700 - 2910 Receive variables from SWCS

2920 - 3160 Find the proposed alert status for each base

3170 - 3400 Find value level for aircraft escaping

3410 - 3680 Convert value level into number of aircraft
 escaping

3690 - 4240 Find spacing time and decision/alert times

4250 - 4770 Calculate weights for aircraft escaping and
 alert cost attributes

4780 -5430 Print sensitivity analysis output

```

1000 REM *****
1010 REM *
1020 REM Program: SNSTVTYA *
1030 REM *
1040 REM Programmer: Doug Lee *
1050 REM Date: 30 Nov 82 *
1060 REM Language: IBM PC Advanced BASIC Version 1.00 *
1070 REM *
1080 REM *
1090 REM This sensitivity analysis is designed for use *
1100 REM with the SAC worth assessment software. The *
1110 REM sensitivity analysis provides the decision maker *
1120 REM with the changes needed in number of aircraft *
1130 REM escaping, and the changes in weights for aircraft *
1140 REM escaping and alert cost in order for a given alert*
1150 REM status (lower than the proposed alert status) to *
1160 REM exceed the proposed alert status. Each of these *
1170 REM changes is calculated independently with all *
1180 REM other variables held constant. The number of *
1190 REM aircraft escaping is then translated into the *
1200 REM spacing time and a combination of decision time *
1210 REM time and alert time needed in order for all the *
1220 REM aircraft to escape. The differences between the *
1230 REM current states and needed change for each area of *
1240 REM interest (aircraft escaping, weights, etc) is *
1250 REM then calculated for each alert status lower than *
1260 REM the proposed alert status. *

```

```

1270 REM *
1280 REM *
1290 REM Listed below are the variables and arrays used in *
1300 REM this program with their associated meanings. *
1310 REM *
1320 REM *
1330 REM Variables: *
1340 REM *
1350 REM Name      Meaning      Unit *
1360 REM *
1370 REM ACES1      Current number of aircraft that *
1380 REM            can escape      A/C *
1390 REM ACES2      Proposed number of aircraft that *
1400 REM            can escape      A/C *
1410 REM ACWGT      Current weight for aircraft *
1420 REM            escaping        n/a *
1430 REM ACWT       Proposed weight for aircraft *
1440 REM            escaping        n/a *
1450 REM ALERT      Counter for alert status      n/a *
1460 REM ALTSCN     Worth score of the proposed alert *
1470 REM            status          n/a *
1480 REM ALTSCR     Worth score of the current alert *
1490 REM            status          n/a *
1500 REM ALTWGT     Weight associated with current *
1510 REM            alert cost      n/a *
1520 REM ALTWT      Weight associated with proposed *
1530 REM            alert cost      n/a *

```

1540 REM BASE	Counter for bases	n/a	*
1550 REM DSN*	Data set name (for loading data)	n/a	*
1560 REM TMEIMP	Impact time of threat	sec	*
1570 REM VAC	Value for aircraft escaping	n/a	*
1580 REM VACECN	Value for proposed aircraft		*
1590 REM	escaping	n/a	*
1600 REM VACECO	Value for current aircraft		*
1610 REM	escaping	n/a	*
1620 REM VALT	Value of alert cost	n/a	*
1630 REM			*
1640 REM			*
1650 REM			*
1660 REM			*
1670 REM			*
1680 REM			*
1690 REM Arrays:			*
1700 REM			*
1710 REM Name	Meaning	Unit	*
1720 REM			*
1730 REM ACADJ(,,1)	Current aircraft escaping	A/C	*
1740 REM ACADJ(,,2)	Proposed aircraft escaping	A/C	*
1750 REM ACADJ(,,3)	ACADJ(,,1) - ACADJ(,,2)	A/C	*
1760 REM			*
1770 REM DAT(*)	Reaction times (Decision time		*
1780 REM	+ Alert time) associated with		*
1790 REM	given spacing times	sec	*
1800 REM			*

1810 REM MAXAC(*)	Maximum aircraft associated	*
1820 REM	with each base	A/C *
1830 REM		*
1840 REM PROALT(,1)	Proposed alert status	n/a *
1850 REM PROALT(,2)	ALTSCR of proposed status	n/a *
1860 REM PROALT(,3)	VAC of proposed status	n/a *
1870 REM PROALT(,4)	VALT of proposed status	n/a *
1880 REM PROALT(,5)	TMEIMP of proposed status	sec *
1890 REM PROALT(,6)	DECTME of proposed status	sec *
1900 REM PROALT(,7)	ALTPAD of proposed status	sec *
1910 REM PROALT(,8)	SPCPAD of proposed status	sec *
1920 REM PROALT(,9)	ACES1 of proposed status	A/C *
1930 REM		*
1940 REM SNSTVA(,,1)	ALTSCR	n/a *
1950 REM SNSTVA(,,2)	VAC	n/a *
1960 REM SNSTVA(,,3)	VALT	n/a *
1970 REM SNSTVA(,,4)	TMEIMP	sec *
1980 REM SNSTVA(,,5)	DECTME	sec *
1990 REM SNSTVA(,,6)	ALTPAD	sec *
2000 REM SNSTVA(,,7)	SPCPAD	sec *
2010 REM SNSTVA(,,8)	ACES1	sec *
2020 REM		*
2030 REM SPA(*)	Spacing times associated with	*
2040 REM	given reaction times	sec *
2050 REM		*
2060 REM VACADJ(,,1)	Value of current aircraft	*
2070 REM	escaping	n/a *

```

2080 REM VACADJ(,,2)      Value of proposed aircraft      *
2090 REM                  escaping                          n/a *
2100 REM VACADJ(,,3)      | VACADJ(,,1) - VACADJ(,,2) |  n/a *
2110 REM                  *
2120 REM WGT(,,1)         ACWGT                            n/a *
2130 REM WGT(,,2)         ACWT                             n/a *
2140 REM WGT(,,3)         | ACWGT - ACWT |                 n/a *
2150 REM WGT(,,4)         ALTWGT                           n/a *
2160 REM WGT(,,5)         ALTWT                            n/a *
2170 REM WGT(,,6)         | ALTWGT - ALTWT |               n/a *
2180 REM                  *
2190 REM                  *
2200 REM *****
2210 REM
2220     DIM ACADJ(10,3,3),DAT(4),MAXAC(10)
2230     DIM PROALT(10,9),SNSTVA(10,4,8),SPA(4)
2240     DIM TIME(10,3,5,4),VACADJ(10,3),WGT(10,3,6)
2250 REM * * * * *
2260 REM * This portion of the program is used to retrieve *
2270 REM * the variables that would normally be passed to *
2280 REM * this program from SWCS                          *
2290 REM * * * * *
2300 REM
2310     GOSUB 2700
2320 REM
2330 REM Find the proposed alert status (alert status with
2340 REM the highest value)

```

2350 REM
2360 GOSUB 2920
2370 REM
2380 REM Find the value level for the number of aircraft
2390 REM that must escape in order for the given alert
2400 REM status' value to exceed that of the proposed alert
2410 REM status
2420 REM
2430 GOSUB 3170
2440 REM
2450 REM Calculate the number of aircraft escaping for a
2460 REM given aircraft escaping value
2470 REM
2480 GOSUB 3410
2490 REM
2500 REM Find the difference needed in spacing time and
2510 REM the decision time/alert time pair to change alert
2520 REM posture
2530 REM
2540 GOSUB 3690
2550 REM
2560 REM Find the weight for a given attribute level in
2570 REM order for that alert status' value to exceed
2580 REM that of the proposed alert status
2590 REM
2600 GOSUB 4250
2610 REM

2620 REM Print output showing the given base with its
2630 REM proposed alert status. Presented are the changes
2640 REM needed (holding all other levels or weights
2650 REM constant) to make a given alert alert status
2660 REM the proposed alert status
2670 REM
2680 GOSUB 4780
2690 END

```

2700 REM
2710 REM * * * * *
2720 REM * This portion of the program is used to retrieve *
2730 REM * the variables that would normally be passed to *
2740 REM * this program from SWCS *
2750 REM * * * * *
2760 REM
2770 REM Load data
2780 REM
2790     INPUT "Data set name";DSN$
2800     OPEN DSN$ FOR INPUT AS #1
2810     FOR BASE = 1 TO 10
2820         FOR ALERT = 1 TO 4
2830             FOR I = 1 TO 8
2840                 INPUT #1,SNSTVA(BASE,ALERT,I)
2850             NEXT I
2860         NEXT ALERT
2870     NEXT BASE
2880     FOR BASE = 1 TO 10
2890         INPUT #1,MAXAC(BASE)
2900     NEXT BASE
2910     RETURN

```

```

2920 REM
2930 REM * * * * *
2940 REM * This portion of the program finds the proposed *
2950 REM * alert status by finding the status with the *
2960 REM * highest worth score. The alert status values *
2970 REM * associated with the proposed alert are then *
2980 REM * stored into an array (PROALT) for further *
2990 REM * manipulation *
3000 REM * * * * *
3010 REM
3020     FOR BASE = 1 TO 10
3030         ALTSCR = -9999
3040         FOR ALERT = 1 TO 4
3050             IF SNSTVA(BASE,ALERT,1) <= ALTSCR GOTO 3090
3060             ALTSCR = SNSTVA(BASE,ALERT,1)
3070             PROALT(BASE,1) = ALERT
3080         NEXT ALERT
3090     NEXT BASE
3100     FOR BASE = 1 TO 10
3110         ALERT = PROALT(BASE,1)
3120         FOR I = 1 TO 8
3130             PROALT(BASE,I+1) = SNSTVA(BASE,ALERT,I)
3140         NEXT I
3150     NEXT BASE
3160     RETURN

```

```

3170 REM
3180 REM * * * * *
3190 REM * Find the value level for the number of aircraft *
3200 REM * that must escape in order for the given alert *
3210 REM * status' value to exceed that of the proposed *
3220 REM * alert status *
3230 REM * * * * *
3240 REM
3250     FOR BASE = 1 TO 10
3260         IF PROALT(BASE,1) = 1 GOTO 3390
3270         FOR ALERT = PROALT(BASE,1)-1 TO 1 STEP -1
3280             ALTSCR = SNSTVA(BASE,ALERT,1)
3290             ALTSCN = SNSTVA(BASE,PROALT(BASE,1),1)
3300             ACWGT = .75
3310             VACECO = SNSTVA(BASE,ALERT,2)
3320             REM
3330             REM Find the new value for a given alert status
3340             REM
3350             VACECN = ALTSCN - ALTSCR + (ACWGT * VACECO)
3360             VACECN = VACECN / ACWGT
3370             VACADJ(BASE,ALERT) = VACECN
3380             NEXT ALERT
3390         NEXT BASE
3400     RETURN

```

```

3420 REM * * * * *
3430 REM * Calculate the number of aircraft escaping for a *
3440 REM * given aircraft escaping value *
3450 REM * * * * *
3460 REM
3470     FOR BASE = 1 TO 10
3480         IF PROALT(BASE,1) = 1 GOTO 3670
3490         FOR ALERT = PROALT(BASE,1)-1 TO 1 STEP -1
3500             REM
3510             REM Find current number of aircraft escaping
3520             REM
3530             ACES1 = MAXAC(BASE) * SNSTVA(BASE,ALERT,2)
3540             IF ACES1 = 0 GOTO 3570
3550             IF INT(ACES1) / ACES1 = 1 GOTO 3570
3560             ACES1 = INT (ACES1 + 1)
3570             ACADJ(BASE,ALERT,1) = ACES1
3580             REM
3590             REM Find proposed number of aircraft escaping
3600             REM
3610             ACES2 = MAXAC(BASE) * VACADJ(BASE,ALERT)
3620             IF INT(ACES2) / ACES2 = 1 GOTO 3640
3630             ACES2 = INT (ACES2 + 1)
3640             ACADJ(BASE,ALERT,2) = ACES2
3650             ACADJ(BASE,ALERT,3) = ABS(ACES1 - ACES2)
3660             NEXT ALERT
3670         NEXT BASE
3680     RETURN

```

```

3690 REM
3700 REM * * * * *
3710 REM * Find the difference needed in spacing time and *
3720 REM * the decision time/alert time pair to change *
3730 REM * alert posture *
3740 REM * * * * *
3750 REM
3760     FOR BASE = 1 TO 10
3770         IF PROALT(BASE,1) = 1 GOTO 4230
3780         FOR ALERT = PROALT(BASE,1)-1 TO 1 STEP -1
3790             DAT(4) = SNSTVA(BASE,ALERT,6)
3800             DAT(4) = DAT(4) + SNSTVA(BASE,ALERT,5)
3810             SPA(1) = SNSTVA(BASE,ALERT,7)
3820             TMEIMP = SNSTVA(BASE,ALERT,4)
3830             ACES2 = ACADJ(BASE,ALERT,2) - 1
3840             REM
3850             REM Calculate spacing time with reaction time
3860             REM (Decision time + Alert time) held constant
3870             REM
3880             SPA(4) = -999.99
3890             IF ACES2 = 0 GOTO 3910
3900             SPA(4) = (TMEIMP - DAT(4)) / ACES2
3910             REM
3920             REM Calculate reaction time with spacing time
3930             REM held constant
3940             REM
3950             DAT(1) = TMEIMP - (ACES2 * SPA(1))

```

```

3960      REM
3970      REM Calculate reaction time with spacing
3980      REM time 1/3 between the lower and upper
3990      REM spacing times
4000      REM
4010          SPA(2) = (SPA(4) - SPA(1)) / 3
4020          SPA(2) = SPA(1) + SPA(2)
4030          DAT(2) = TMEIMP - (ACES2 * SPA(2))
4040      REM
4050      REM Calculate reaction time with spacing
4060      REM time 2/3 between the lower and upper
4070      REM spacing times
4080      REM
4090          SPA(3) = (SPA(4) - SPA(1)) * (2 / 3)
4100          SPA(3) = SPA(1) + SPA(3)
4110          DAT(3) = TMEIMP - (ACES2 * SPA(3))
4120          TIME(BASE,ALERT,1,1) = SPA(1)
4130          TIME(BASE,ALERT,1,2) = DAT(4)
4140          TIME(BASE,ALERT,1,3) = 0
4150          TIME(BASE,ALERT,1,4) = 0
4160          FOR OP = 2 TO 5
4170              TIME(BASE,ALERT,OP,1) = SPA(OP-1)
4180              TIME(BASE,ALERT,OP,2) = DAT(OP-1)
4190              TIME(BASE,ALERT,OP,3) = SPA(1) - SPA(OP-1)
4200              TIME(BASE,ALERT,OP,4) = DAT(4) - DAT(OP-1)
4210          NEXT OP
4220      NEXT ALERT

```

4230 NEXT BASE

4240 RETURN

```

4250 REM
4260 REM * * * * *
4270 REM * Find the weight for a given attribute level in *
4280 REM * order for that alert status' value to exceed *
4290 REM * that of the proposed alert status *
4300 REM * * * * *
4310 REM
4320     FOR BASE = 1 TO 10
4330         IF PROALT(BASE,1) = 1 GOTO 4760
4340         FOR ALERT = PROALT(BASE,1)-1 TO 1 STEP -1
4350             ALTSCR = SNSTVA(BASE,ALERT,1)
4360             ALTSCN = SNSTVA(BASE,PROALT(BASE,1),1)
4370             ALTWGT = .25
4380             ACWGT = .75
4390             VAC = SNSTVA(BASE,ALERT,2)
4400             VALT = SNSTVA(BASE,ALERT,3)
4410             WGT(BASE,ALERT,1) = ACWGT
4420             WGT(BASE,ALERT,4) = ALTWGT
4430         REM
4440         REM Find the weight for a given alert status
4450         REM and attribute level (A/C and Alert Cost)
4460         REM
4470         REM
4480         REM Weight for alert cost
4490         REM
4500             ALTWT = ALTSCR - VALT * ALTWGT - ALTSCN
4510             ALTWT = ALTWT + ALTSCN * ALTWGT

```

```

4520      ALTWT = ALTWT / (ALTSCR - VALT)
4530      REM
4540      REM Weight for alert cost
4550      REM
4560      ACWT = ALTSCR - VAC * ACWGT - ALTSCN
4570      ACWT = ACWT + ALTSCN * ACWGT
4580      ACWT = ACWT / (ALTSCR - VAC)
4590      REM
4600      REM Check weights for validity
4610      REM
4620      ALTSCR = ACWT * VAC + ALTWT * VALT
4630      IF ALTSCR >= ALTSCN GOTO 4670
4640      ACWT = -999.99
4650      ALTWT = -999.99
4660      REM
4670      REM Store weights
4680      REM
4690      WGT(BASE,ALERT,2) = ACWT
4700      WGT(BASE,ALERT,5) = ALTWT
4710      WGT(BASE,ALERT,3) = WGT(BASE,ALERT,1) - ACWT
4720      WGT(BASE,ALERT,3) = WGT(BASE,ALERT,3)
4730      WGT(BASE,ALERT,6) = WGT(BASE,ALERT,4) - ALTWT
4740      WGT(BASE,ALERT,6) = WGT(BASE,ALERT,6)
4750      NEXT ALERT
4760      NEXT BASE
4770      RETURN

```

```

4780 REM
4790 REM * * * * *
4800 REM * Print output showing the given base with its *
4810 REM * proposed alert status. Presented are the changes*
4820 REM * needed (holding all other levels or weights *
4830 REM * constant) to make a given alert alert status *
4840 REM * the proposed alert status *
4850 REM * * * * *
4860 INPUT "Desired base";BASE
4870 LPRINT "Base: ";
4880 LPRINT USING "#";BASE;
4890 LPRINT SPC(29)"Proposed Alert Status: ";
4900 LPRINT USING "#";PROALT(BASE,1)
4910 LPRINT
4920 LPRINT
4930 LPRINT "Alert Aircraft* ";
4940 LPRINT " Weight** Weight**"
4950 LPRINT "Status Escaping ";
4960 LPRINT "Aircraft Alert Cost"
4970 LPRINT
4980 FOR ALERT = PROALT(BASE,1) - 1 TO 1 STEP -1
4990 FOR K = 1 TO 3
5000 IF K = 1 THEN LPRINT " Current ";
5010 IF K = 2 THEN LPRINT USING " # ";ALERT;
5020 IF K = 2 THEN LPRINT "Modified ";
5030 IF K = 3 THEN LPRINT " Difference ";
5040 LPRINT USING " ###.## ";ACADJ(BASE,ALERT,K);

```

```

5050      LPRINT "      ";
5060      LPRINT USING "    ####.## ";WGT(BASE,ALERT,K);
5070      LPRINT USING "    ####.##";WGT(BASE,ALERT,K+3)
5080      NEXT K
5090      LPRINT
5100      NEXT ALERT
5110      LPRINT
5120      LPRINT
5130      LPRINT" * - A/C escaping calculated with weights";
5140      LPRINT" held constant"
5150      LPRINT"** - Weights calculated with values";
5160      LPRINT" held constant"
5170      LPRINT
5180      LPRINT
5190      LPRINT
5200      LPRINT "Alert          Current   Option   Option";
5210      LPRINT "    Option   Option";
5220      LPRINT "Status          Inputs    1        2    ";
5230      LPRINT "      3        4    "
5240      LPRINT
5250      FOR A = PROALT(BASE,1) - 1 TO 1 STEP -1
5260          FOR K = 1 TO 4
5270              IF K = 1 THEN LPRINT SPC(8)"Spacing  ";
5280              IF K = 2 THEN LPRINT SPC(8)"Reac Tm  ";
5290              IF K = 3 THEN LPRINT USING "  #";A;
5300              IF K = 3 THEN LPRINT SPC(5)"Spc Dif  ";
5310              IF K = 4 THEN LPRINT SPC(8)"Reac Df  ";

```

```

5320         FOR O = 1 TO 5
5330             LPRINT USING "####.##  ";TIME(BASE,A,O,K);
5340             NEXT O
5350         LPRINT
5360         NEXT K
5370     LPRINT
5380     NEXT A
5390 LPRINT
5400 LPRINT
5410 LPRINT "NOTE:"
5420 LPRINT "Reac Tm = Decision Time + Alert Time"
5430 RETURN

```

Bibliography

- Allen, J., D. Buede, and M. O'Connor. The Use of Multi-Attribute Value Assessment Techniques in the Development of a Marine Corps Combat Readiness Evaluation System (MCCRES). McLean, Va: Decisions and Designs, Inc., 1977.
- Bowman, E. H. "Consistency and Optimality in Managerial Decision Making," Management Science, 9: 310-321, 1963.
- Chinnis, T. O., C. W. Kelley, R. D. Minckler, and M. F. O'Connor. Single Channel Ground and Airborne Radio System (SINCGARS) Evaluation Model. McLean, Va: Decisions and Designs, Inc., 1975.
- DeWispelare, A. R., Captain, USAF. Lectures in ST6.31, Weapons Effectiveness/Tradeoff Studies, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, 1980 & 1982.
- DeWispelare, A. R. and A. P. Sage. "On Combined Multiple Objective Optimization Theory and Multiple Attribute Utility Theory for Evaluation and Choicemaking" Journal of Large Scale Systems, 2: 1-19, 1981.
- Farris, D. R. and A. P. Sage. "On Decision Making and Worth Assessment," International Journal of System Sciences, 6: 12: 1135-78, December 1975.
- Fischer, G. W., W. Edwards, and C. W. Kelly, III. Decision Theoretic Aids for Inference, Evaluation, and Decision Making: Review of Research and Experience. McLean, Va: Decision and Designs, Inc., Feb 1978. (AD053962).
- Goldberg, L. R. "Man Versus Model of Man: A Rationale, Plus Some Evidence for a Method of Improving on Clinical References." Psych Bulletin 73: 422-432, 1970.
- Keeny, Ralph L. and Howard Raiffa. Decisions with Multiple Objectives: Preference and Value Tradeoff. New York: John Wiley and Sons, 1976.
- Lee, David B. Enhanced Decision Analysis Support System. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1981. (AFIT/GST/OS/81M-8).
- Lee, S. M. Goal Programming for Decision Analysis. Philadelphia: Auerbach Publishers, 1972.

Lucas, Gary L. and Jan Ruff. An Investigation of Operational Decision Aids. Arlington, Va: System Planning Corp, July 1977. (AD-A047 147).

MacCrimmon, K. R. "An Overview of Multiple Objective Decision Making" in Multiple Criteria Decision Making, J. L. Cochrane and M. Zeleny (ed.), Columbia, S. C.: University of South Carolina Press, 1973.

Miller, James R. Professional Decision Making. New York: Praeger Publishers, 1970.

Morlan, Bruce W. A Computer-Based Decision Analysis Support System. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, Dec 1979. (AFIT/GOR/SM/79D-6, AD-A083706).

Sage, Andrew P. Methodology for Large-Scale Systems. New York: McGraw-Hill, 1977.

Stimpson, Wayne A. MADAM: Multiple-Attribute Decision Analysis Model. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, Dec 1981. (AFIT/GOR/AA/81D-1).

APPENDIX B

**SAMPLE OUTPUT
FOR THE
WORTH ASSESSMENT ALGORITHM
AND
SENSITIVITY ANALYSIS**

SAMPLE OUTPUT DESCRIPTION

Worth Assessment Output

Figure 1 is a sample of the output produced by the worth assessment algorithm designed for the SAC warning system. The output presented shows the proposed alert status for each base given in the hypothetical scenario. Also in this output are the other variables associated with the proposed alert status.

Sensitivity Analysis Output

Presented in Figure 2 is sample output produced by the sensitivity analysis software designed in conjunction with the worth assessment algorithm. This output was generated from the results of the worth assessment output -- specifically, the output for base 5. The output for the sensitivity analysis can be divided into two areas. The first area is the top half of the output and represents the changes needed in a given alert status in order for that alert status to become the proposed alert status. These changes are calculated for the number of aircraft escaping and the weights associated with the aircraft escaping attribute and alert cost attribute. Each of these changes was derived with all other variables held constant. For example, when the number of aircraft was calculated, the weights for aircraft escaping and alert cost were constant. The output for each alert status was the current state of the variable (Current), the level of the variable needed to

Base	Impact Time	Decision Time	Alert Time	Spacing Time	Aircraft Escaping	Maximum Aircraft	Proposed Alert Status
1	325	101	600	48	0	20	1
2	340	101	600	55	0	30	1
3	355	101	200	48	2	25	2
4	370	101	600	45	0	35	1
5	385	101	10	21	14	20	4
6	400	101	200	56	2	23	2
7	415	101	60	26	10	32	3
8	430	101	200	47	3	37	2
9	445	101	60	38	8	28	3
10	460	101	60	31	10	22	3

Figure 1. Worth Assessment Output

make the alert status the proposed status (Modified), and the difference between the current and modified variables (Difference). For example, the number of aircraft escaping when calculating the original worth score for alert status 4 was 11. The number of aircraft that need to escape in order for alert status 3 to become the proposed alert status is 12, for a difference of 1.

The second half of the sensitivity output translates the number of aircraft that must escape in each alert status into spacing time (Spacing) and a combination of decision time and alert time (Reac Tm). The output consists of the current spacing and reaction time and four options that the decision maker can use to attain the needed number of aircraft to make a given alert status the proposed alert status. Each option shows the decision maker the needed spacing time, needed reaction time, the difference between the current spacing time and the needed spacing time, and

Base: 5

Proposed Alert Status: 4

Alert Status		Aircraft* Escaping	Weight** Aircraft	Weight** Alert Cost
3	Current	11.00	0.75	0.25
	Modified	12.00	1.00	0.00
	Difference	1.00	-0.25	0.25
2	Current	5.00	0.75	0.25
	Modified	10.00	0.45	0.55
	Difference	5.00	0.30	-0.30
1	Current	0.00	0.75	0.25
	Modified	8.00	0.45	0.55
	Difference	8.00	0.30	-0.30

* - A/C escaping calculated with weights held constant

** - Weights calculated with values held constant

Alert Status		Current Inputs	Option 1	Option 2	Option 3	Option 4
3	Spacing	20.91	20.91	20.72	20.53	20.34
	Reac Tm	161.27	155.04	157.11	159.19	161.27
	Spc Dif	0.00	0.00	0.19	0.38	0.57
	Reac Df	0.00	6.23	4.16	2.08	0.00
2	Spacing	20.91	20.91	17.04	13.17	9.30
	Reac Tm	301.27	196.85	231.66	266.46	301.27
	Spc Dif	0.00	0.00	3.87	7.73	11.60
	Reac Df	0.00	104.42	69.61	34.81	0.00
1	Spacing	20.91	20.91	-1.12	-23.15	-45.18
	Reac Tm	701.27	238.66	392.86	547.07	701.27
	Spc Dif	0.00	0.00	22.03	44.06	66.09
	Reac Df	0.00	462.61	308.41	154.20	0.00

NOTE:

Reac Tm = Decision Time + Alert Time

Figure 2. Sensitivity Analysis Output

the difference between the current reaction time and needed reaction time. Option 1 shows the change needed in reaction time when spacing time is not changed from its current state, while option 4 shows the change needed in spacing time when reaction time is held to its current state. The intermediate options are combinations of spacing and reaction times. For example, to make alert status 2 the proposed alert for base 5 using option 3, spacing time would have to be decreased by 8 seconds (from 21 seconds to 13 seconds), and reaction time must decrease 35 seconds (from 301 seconds to 266 seconds).

VITA

Douglas E. Lee was born in Waco, Texas, on June 26, 1955. He received a Bachelor of Science in Management Sciences from the United States Air Force Academy in June 1978. He served as a Trajectory Applications Analyst, in the 544th SIW/Trajectory Division from June 1978 until his entry into the School of Engineering, Air Force Institute of Technology in June 1981.

Permanent Address: 5932 Bramblewood Court
Dayton, Ohio 45424

END