(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A274 901

DTIC
ELECTE
JAN 25 1994
S B D

THESIS

IMPLEMENTING
A MULTIPLE CRITERIA MODEL BASE
IN CO-OP
WITH A GRAPHICAL USER INTERFACE GENERATOR

by

Hsin-Yen Tsai

September 1993

Thesis Advisor Tung X Bui

Approved for public release, distribution is unlimited

94-02103

94 1 24 036

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503

1 AGENCY USE ONLY *(Leave blank)*	2 REPORT DATE 23 Sep 1993	3 REPORT TYPE AND DATES COVERED Master's Thesis

4 IMPLEMENTING A MULTIPLE CRITERIA MODEL BASE IN CO-OP WITH A GRAPHICAL USER INTERFACE GENERATOR

5 FUNDING NUMBERS

6 AUTHOR(S) Tsai, Hsin-Yen

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8 PERFORMING ORGANIZATION REPORT NUMBER

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10 SPONSORING/MONITORING AGENCY REPORT NUMBER

11 SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U S Government

12a DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited

12b DISTRIBUTION CODE
A

13 ABSTRACT

This thesis designs and implements prototype Multiple Criteria Decision Making (MCDM) modules to add into the Cooperative Multiple Criteria Group Decision Support System (Co-op) for Windows. The algorithms and the graphical user interfaces for these modules are implemented using Microsoft Visual Basic under the Windows based environment operating in a IBM compatible microcomputer. Design of the MCDM programs interface is based on general interface design principles of user control, screen design, and layout.

14 SUBJECT TERMS
Multiple Criteria Decision Making, Group Decision Support System, Graphical User Interface, and Co-op

15 NUMBER OF PAGES
87

16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	18 SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	20 LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

IMPLEMENTING A MULTIPLE CRITERIA MODEL BASE IN CO-OP WITH A GRAPHICAL USER INTERFACE GENERATOR

by

Hsin-Yen Tsai
Lieutenant, Taiwan Navy
B.S., Chinese Naval Academy, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1993

Author _____
Hsin-Yen Tsai

Approved by _____
Tung X. Bui, Thesis Advisor

_____
Myung M. Suh, Associate Advisor

_____
David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

This thesis designs and implements prototype Multiple Criteria Decision Making (MCDM) modules to add into the Cooperative Multiple Criteria Group Decision Support System (Co-op) for Windows. The algorithms and the graphical user interfaces for these modules are implemented using Microsoft Visual Basic under the Windows based environment operating in a IBM compatible microcomputer. Design of the MCDM programs interface is based on general interface design principles of user control, screen design, and layout.

DTIC QUALITY INSPECTED 8

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. OBJECTIVES 2

C. SCOPE 2

D. ORGANIZATION 5

II. PROTOTYPE 1: ELECTRE I 6

A. THE ALGORITHM 6

B. NUMERIC EXAMPLE 8

1. Problem Outline 8

2. Walk Through 9

C. INDIVIDUAL SCREEN DESIGN OF ELECTRE I 11

1. ELECTRE I Main Screen 11

2. The Scaling Method Screen 14

a. The Standard Scale Screen 14

b. The Weighted Standard Scale Screen 16

c. The Free Scale Screen 18

3. The Alternatives Evaluation Screen 20

4. The Threshold Input Screen 20

5. The Outranking Matrix Screen 22

a. The Concordance Matrix Screen 24

b. The Discordance Matrix Screen 24

III. PROTOTYPE 2: ELECTRE III 27

A. THE ALGORITHM 27

B. SCREEN DESIGN OF ELECTRE III 29

1. The Threshold Input Screen 29

2. The Result Screen 31

3. The Credit Matrix Screen 31

IV. PROTOTYPE 3: ELECTRE IV 35

A. THE ALGORITHM 35

B. NUMERIC EXAMPLE 37

C. SCREEN DESIGNS OF ELECTRE IV 39

V. PROTOTYPE 3: PROMETHEE 44

A. THE ALGORITHMS 44

1. Basic Algorithm of PROMETHEE I and PROMETHEE II 45

a. Use of the Algorithm in PROMETHEE I 49

b. Use of the Algorithm in PROMETHEE II 50

2. Algorithm of PROMETHEE V 50

B. SCREEN DESIGNS OF PROMETHEE 51

1. PROMETHEE I and PROMETHEE II 52

a. The Main Screen 52

b. The Criteria Identification Screen 52

c. The Generalized Criteria Introduction Screen 55

d. The Alternatives Evaluation Screen 57

e. The Result Screen 57

f. The Result Matrix Screen 60

2. PROMETHEE V 60

a. The Main Screen 60

b. The Objective Function and Constraints Definition Screen . 64

c. The Coefficient Matrix Screen 64

d. The Result Screen 67

VI. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH . . . 69

A. SUMMARY 69

B. RECOMMENDATIONS FOR FUTURE RESEARCH 69

APPENDIX A 71

A. WHAT IS VISUAL BASIC? 71

B. CONTROLS USED IN THE DESIGN OF CO-OP 72

1. Text box 72

2. Label 74

3. Picture Box 74

4. Frame 74

5. Command Button 74

6. Check Box 75

7. Option Button 75

8. Spin Button 75

9. Grid 75

LIST OF REFERENCES 77

INITIAL DISTRIBUTION LIST 78

I. INTRODUCTION

A. BACKGROUND

In the last decade, research on Multiple Criteria Decision Making (MCDM) in the field of management science has evolved from its infancy to a high level of maturity. Unlike optimization models, MCDM techniques are suggestive rather than prescriptive. They focus on helping decision makers to structure complex decision problems using operation research techniques. However, in order to facilitate the use of MCDM techniques, MCDM researchers should acknowledge the importance of providing user-friendly tools.

There are a wide variety of MCDM problems that a manager has to solve everyday. We always choose the best alternative by comparing either via ranking or ordering of the objects of interest with respect to given criteria of choice. (Zeleny, 1982)

Multiple and conflicting objectives, for example, "better performance" and "lower cost", are the daily issues faced by decision-makers and managers. To achieve an objective sometimes requires the compromise of another. The MCDM is a process of resolving and balancing these conflicting objectives in an optimized way.

To solve a multiple criteria decision problem, the evaluation criteria must first be identified. Weights are commonly allocated to the various criteria to identify their relative importance.

The crux of the problem is to identify the various issues or criteria that compose the problem. The weight of each criterion will, of course, depend on how the decision makers feel about the impact of each criterion on the outcome of the decision. The MCDM programs implemented in this thesis will help decision-makers to structure an otherwise unstructured problem and to provide them with suggestive decision.

B. OBJECTIVES

Co-op is a Cooperative Multiple Criteria Group Decision Support System (MCGDSS) developed by Tung X Bui. This thesis involves the design and implementation of algorithms and the graphical user interfaces (GUI) for MCDM modules to add into the existing Co-op for Windows program which was developed by Tung X Bui and Ralph Sabene (1992).

The MCDM modules are implemented using Microsoft Visual Basic exploiting its GUI generating capability. It is expected that the MCDM modules help users to effectively solve the MCDM problems in a user-friendly and interactive environment.

C. SCOPE

The scope of this thesis involves the design and implementation of MCDM modules for the Windows-based Co-op system. The prototype MCDM modules to be implemented in this thesis are ELECTRE I, ELECTRE III, ELECTRE IV, and PROMETHEE.

To help readers understand where these MCDM modules are to be used in the Co-op, a brief description of the Co-op is provided below.

Figure 1 shows the main screen of the Co-op system. The main screen gives the user an overview of the six decision making steps for solving a MCDM problem. Each labeled command button the user sees on the main screen identifies a step in the decision making process. When clicked on any of the command buttons, it opens that particular sub-module. The design itself represents a flow chart of how a problem could be solved. The first step defines all the available alternatives along with the criteria of measurement, and the group norm which includes identifying members, members' weights, and communication parameters. This step must be completed before proceeding further.

The model then allow two courses of action. The first is to utilize the various model components, i.e., ELECTRE I, ELECTRE III, ELECTRE IV, and PROMETHEE, to evaluate alternatives. If the user chooses this course, the criteria of the particular problem must first be defined. To define criteria, the user must use the ***Criteria Prioritization*** button to start the criteria definition module. The user will be able to prioritize criteria by assigning proper weight on each criterion either using a direct input or pairwise comparison method. Once the criteria prioritization is complete, the user will be able to choose a specific MCDM model to evaluate alternatives by clicking on the ***Alternative Evaluation*** command button on the Co-op main screen. This command button opens the ranking method screen (see Figure 2). The user can choose a ranking method to solve a particular problem on this screen.

The alternate course, if chosen, will allow user to rank alternatives directly without going through formal alternative evaluations.

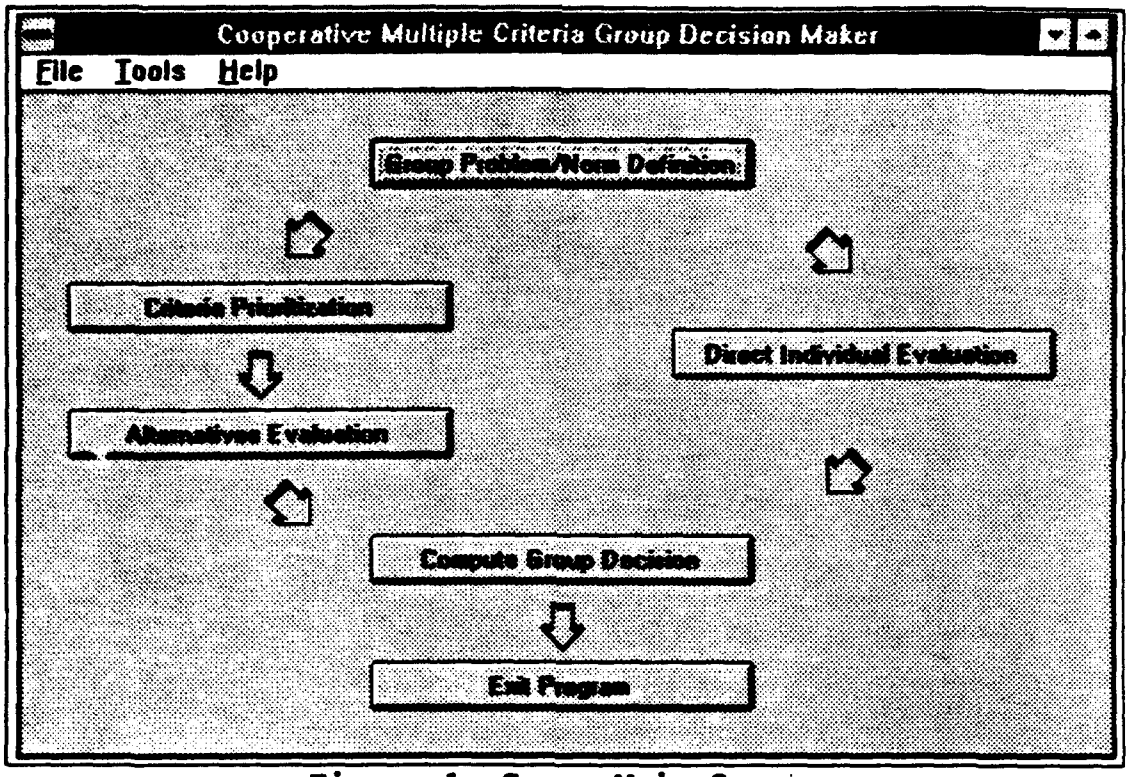


Figure 1. Co-op Main Screen

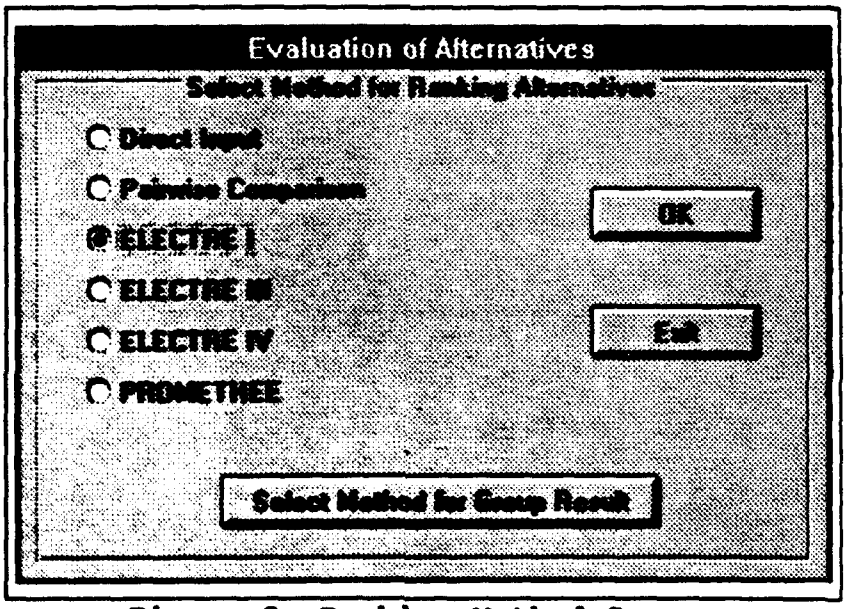


Figure 2. Ranking Method Screen

Both courses lead to the group decision button which will compute and display group decision results according to the defined norm. The final command button exits the program.

To sum up, this thesis designs and implements the four MCDM modules that are to be invoked from the ranking method screen as shown in Figure 2

D. ORGANIZATION

This thesis is divided into seven chapters. Chapter I gives a brief introduction and states the objective and scope of the thesis. Chapters II, III, IV, and V briefly describe the algorithms of ELECTRE I, ELECTRE III, ELECTRE IV, and PROMETHEE, respectively. Analysis of the algorithms is beyond the scope of this thesis and hence will not be discussed. Instead the design specifications for these algorithms are presented in these three chapters. Chapter VI contains the conclusions and recommendations for further research. Appendix A provides a description of basic constructs of the Microsoft Visual Basic.

II. PROTOTYPE 1: ELECTRE I

A. THE ALGORITHM

ELECTRE I (ELimination Et Choix Traduisant la REalite) is a multiple criteria decision-aid model. It is intended to structure the set of alternatives using constructs known as concordance and discordance. The idea is to provide a partial order of alternatives. Then through the use of concordance threshold and discordance threshold as two filters to generate a set of outranking or "best" solutions.

> This model was developed by Professor Bernard Roy, director of the LAMSADE (Laboratoire d'Analyze it de Modelisation des Systemes pour l'Aide a la DEcision) of the University of Paris at Dauphine (1968) in France and was introduced to the United States by Bui (1982). ELECTRE I is known by its simplicity in modeling the decision-maker's preferences. More importantly, it seeks to avoid forcing the decision-maker to arbitrarily eliminate good decision alternatives. The ELECTRE I model tries to add structure to the evaluation process of decision making by helping the decision maker to analyze preferences with objectivity and confidence (Price, 1992).

The principles of ELECTRE I are based on the rule developed by the French philosopher, the Marquis de Condorset (1750).

> When an action A is better than another action B in the majority of decision criteria, and, in addition, there is no criterion by which A is clearly worse than B, we can say without risk that A is better than B, or, in other words, A outranks B.

The constructs of concordance and discordance are presented by two separated matrices. Each entry in both matrices, which is named concordance coefficient and discordance coefficient respectively, is a result of computation through pairwise comparison. The concordance coefficient is defined by the formula

$$C_{AB} = \frac{\text{Sum of the weights of the criteria by which A outranks B}}{\text{Sum of the weights of all the criteria of the model}}$$

It indicates to what extent an alternative, A, is better than another, B.

The discordance coefficient is calculated from evaluation scores instead of weights and is defined by the formula:

$$D_{AB} = \frac{\text{The greatest negative variation (i.e., B outranks A) between the evaluation scores for a single criterion}}{\text{The maximum range between the highest possible score and the lowest possible score}}$$

This coefficient indicates to what extent an alternative, A, contains discordant elements that might make the alternative unsatisfactory. Both of these factors vary from 0 to 1. A perfect value for the concordance coefficient is 1, a "fatal" score for the discordance coefficient is 1. (Bui, 1982)

These two factors will then be compared with the concordance threshold and discordance threshold that are chosen arbitrarily by user. The concordance threshold, P, varies from 0.5 to 1, and is more severe as it approaches 1; the discordance threshold, Q, is more severe as it approaches 0. The rules of ELECTRE method can be summarized as follows:

IF	THEN
$C_{A,B} >= P$ and $D_{A,B} <= Q$	A outranks B
$C_{B,A} >= P$ and $D_{B,A} <= Q$	B outranks A
A outranks B and B outranks A	The options are equivalent
Otherwise	The comparison is characterized by indetermination or incomparability

Source: (Bui, 1982)

B. NUMERIC EXAMPLE

1. Problem Outline

The example used to walk through the discussion of screen designs of ELECTRE I is the case of the Caisse Maladie Avenir (CMA) (Price, 1991). The problem is briefly outlined below:

1. The CMA must equip itself with computer hardware, because the current application (liaison with the RESO firm, which does all the calculations) does not allow it to carry out analyses.

2. The alternatives:
   a. RESO = status quo
   b. RESO-FUTURE = keep RESO for calculations and buy a computer for the analyses.
   c. S38CMV = buy an IBM S38 from Caisse Maladie Vaudoise (CMV), and the software to go with it.
   d. AS400CMV = buy a new IBM AS400.
   e. UNISYS = buy the complete UNISYS application.
   f. UNI-FR = use the Fribourg University computers and make the students program the applications.

3. The criteria and weights:

- the cost of	
1. investment	1.0
2. exploitation	0.5
- the performance of the application	
3. reliability	1.5
4. operation speed	1.5
5. flexibility of application as far as changes are concerned	1.0
- satisfaction of needs in	
6. operational treatment	0.5
- implementation of application	
7. rapidity of utilization	1.0
8. complexity of changes to make	1.0
9. resistance manifested by the personnel following changes to work procedures	0.5
10. independence the future can offer	1.5

This example will also be used when discussing ELECTRE III, and PROMETHEE techniques.

2. Walk Through

This problem has to be defined using the problem definition module provided by clicking the ***Group Problem/Norm Definition*** button on the Co-op main screen. The clicking will open the group problem/norm identification screen, as shown in Figure 3. Since this is a new problem, the user should select the ***Define NEW File*** option from the screen. If the user already has a problem file in the system, he can choose the ***Open PREVIOUS File Option***.

When the user decides to define a new problem and clicks on the ***OK*** button, the problem/norm definition screen is opened, as shown in Figure 4. The user can input the name of the problem file and perform one task at a time. When a task is completed,

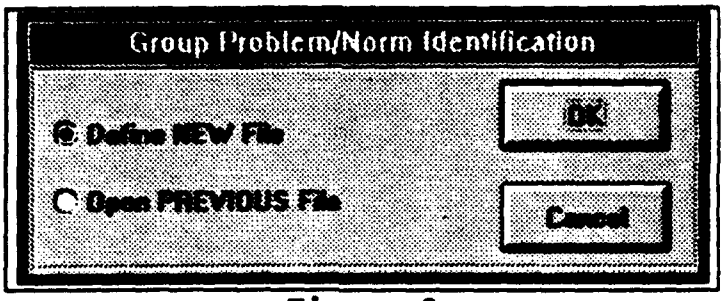


Figure 3
Problem/Norm Identification Screen

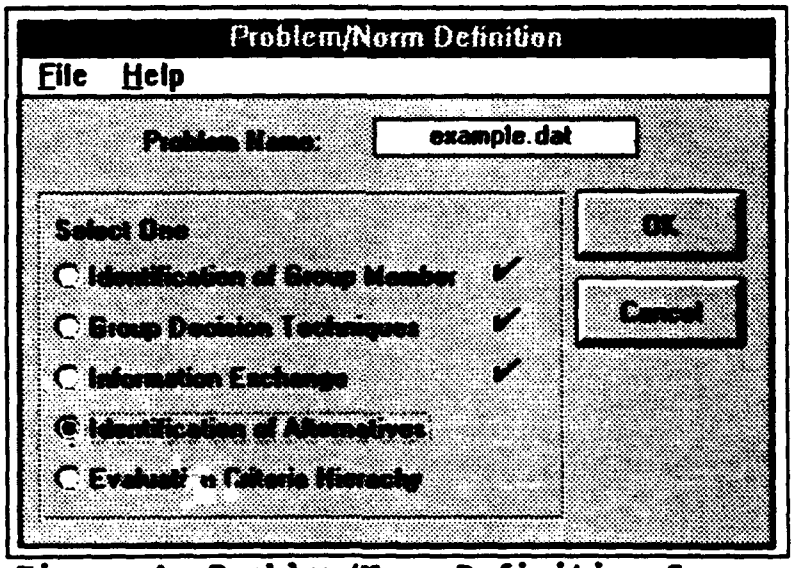


Figure 4. Problem/Norm Definition Screen

it will be marked by a check mark. From this screen, the user can identify the alternatives and evaluation criteria (up to three levels).

When the user select the ***Identification of Alternatives*** option, the alternatives identification screen is opened, as shown in Figure 5. On this screen, the user can input the alternatives, add an alternative, or delete an alternative. When the ***Evaluation Criteria Hierarchy*** option is chosen, the user opens the evaluation criteria identification screen, as shown in Figure 6. On this screen, the user can input the criteria, define the next level criteria, add a criterion, or delete a criterion.

After problem definition process, the user may want to prioritize the criteria. He can click on the ***Criteria Prioritization*** button on the co-op main screen to begin the prioritization. When the button is clicked, the user will be asked to select a problem file from the system database. The open file screen, as shown in Figure 7, is displayed to allow the user to perform this process. After a file been selected, the prioritization method screen, as shown in Figure 8, is opened to allow the user to select desired method.

C. INDIVIDUAL SCREEN DESIGN OF ELECTRE I

1. ELECTRE I Main Screen

ELECTRE I, ELECTRE III, and ELECTRE IV share the same main screen. The steps to structure and solve a problem by ELECTRE I, ELECTRE III, or ELECTRE IV methods are captured in a flow chart and is presented on the ELECTRE I main screen, as shown in Figure 9. The check marks by four main steps of the ELECTRE I main

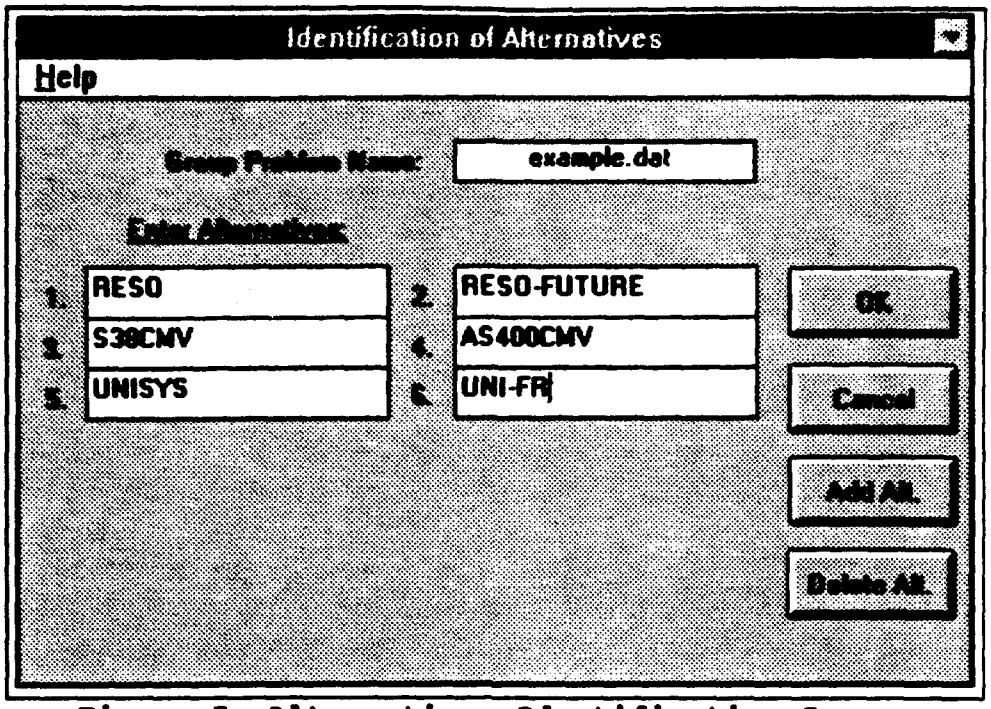


Figure 5. Alternatives Identification Screen

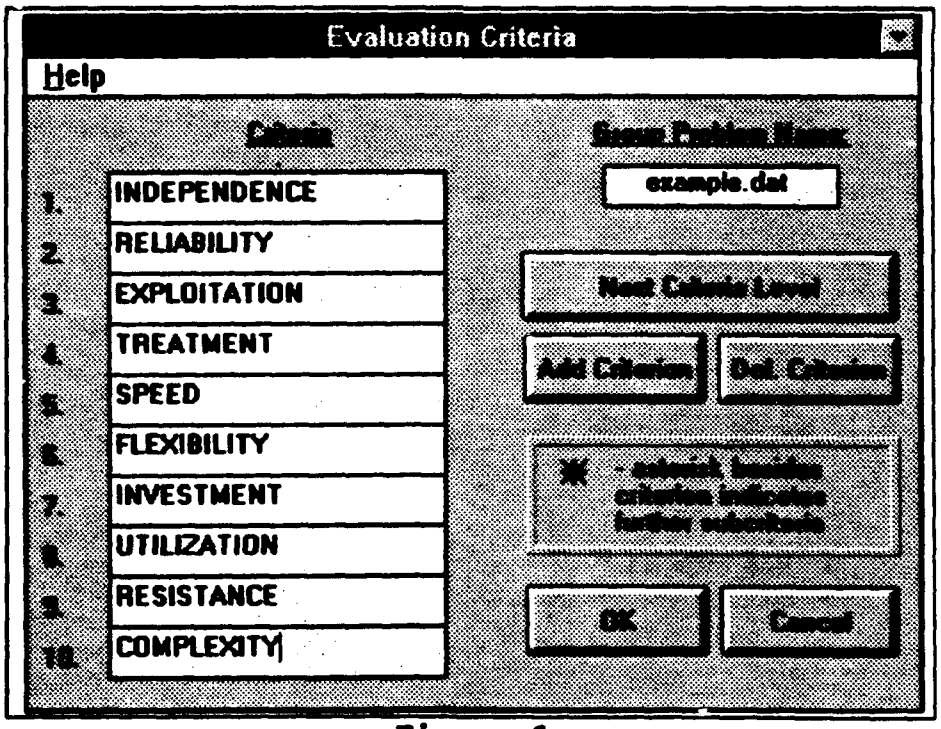


Figure 6
Evaluation Criteria Identification Screen

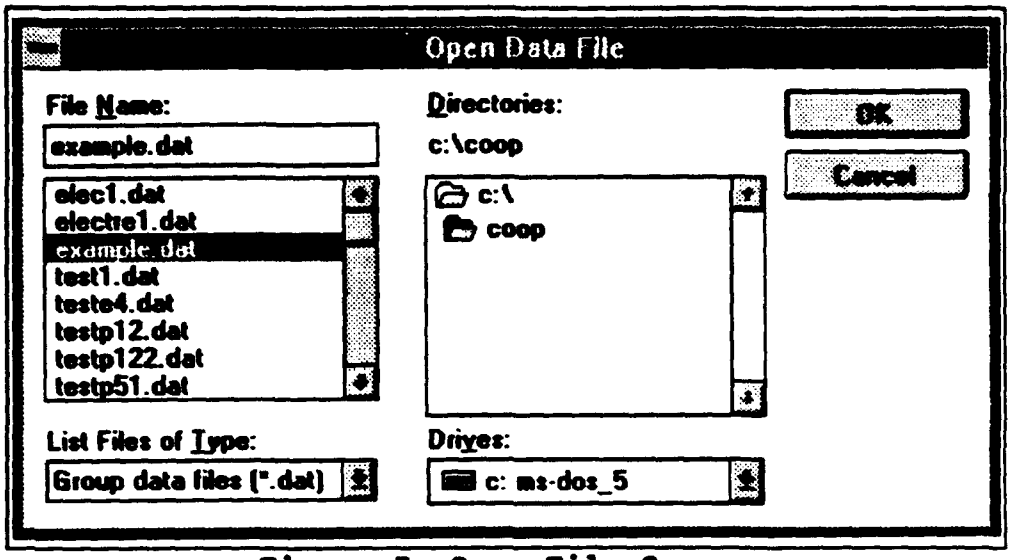


Figure 7. Open File Screen

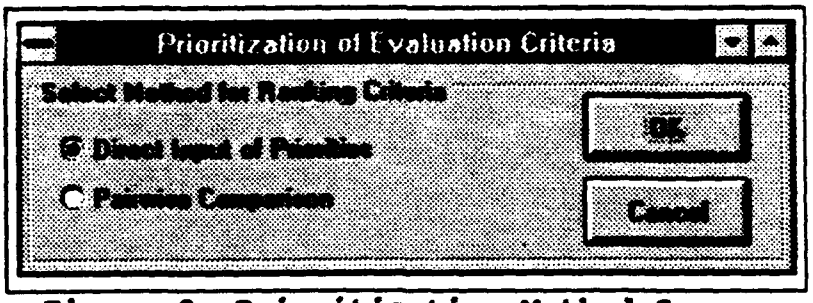


Figure 8. Prioritization Method Screen

screen are invisible at the beginning of a problem. When each step is finished, the check mark beside this step command button will show to indicate its completion. Controls used in this screen are picture boxes and command buttons.

2. The Scaling Method Screen

Figure 10 shows the screen the user sees when he clicks on ***Choose a Scaling Method*** button on the ELECTRE I main screen. There are three available scaling methods which are used to define ordinal scales for each criterion for comparing ordinal to cardinal values. This simple dialogue box allows user to select one of the available scaling methods via option buttons. A scale with five levels (i.e., excellent, good, average, fair, and weak) is then used to evaluate each alternative according to each criterion specified. However, the points associated with each of the five levels vary with the criterion concerned.

The controls used on this screen are frame, command buttons, option buttons and labels. The ***OK*** button accepts whatever choice the user makes and open the corresponding screen for the user. The ***Cancel*** button returns the user to the ELECTRE I main screen.

a. The Standard Scale Screen

Figure 11 shows the information screen display when the user selects the ***Use standard scale*** option on the scaling method screen. The scales for excellent, good, average, fair, and weak performance are 10, 8, 5, 3, and 0 respectively. The ***OK*** button

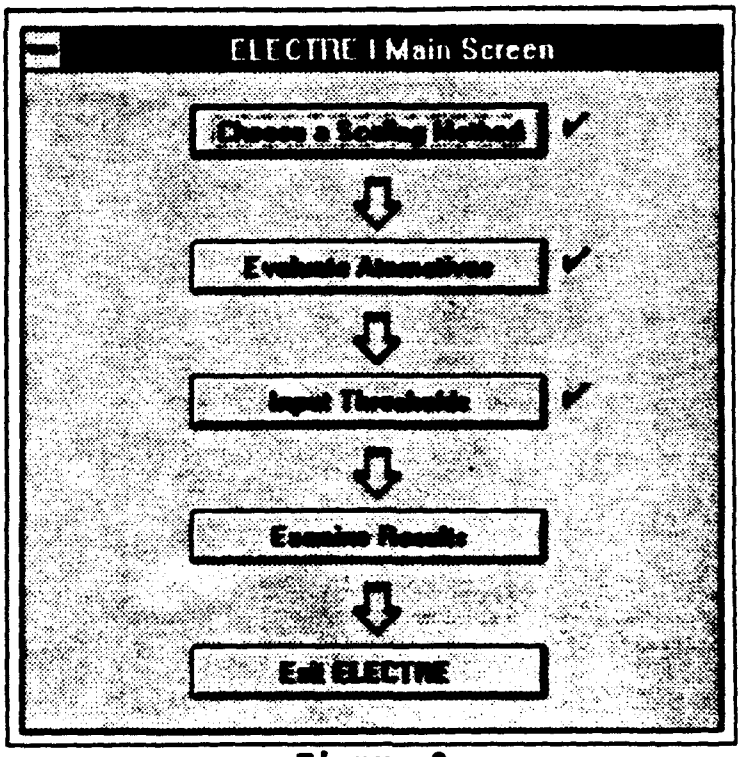


Figure 9
ELECTRE I (III/IV) Main Screen

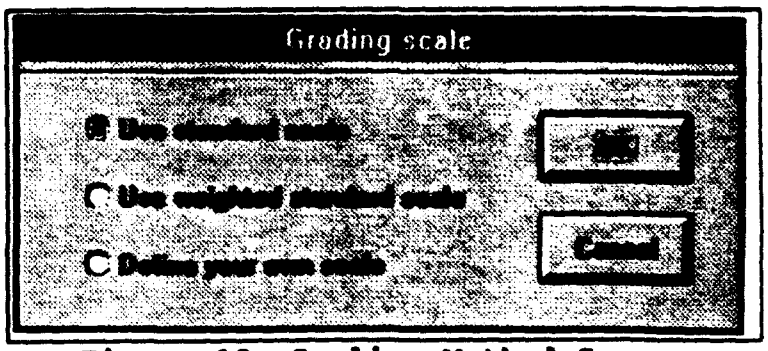


Figure 10. Scaling Method Screen

will close this screen and return the user to the ELECTRE I main screen. The controls used on this screen are command button and labels.

b. The Weighted Standard Scale Screen

Figure 12 shows the screen the user sees when selecting the ***Use weighted standard scale*** option on the scaling method screen. This screen is basically an information screen which displays the weighted standard scales for each criterion. Controls used on this screen are frames, command buttons, and labels.

When the user clicks on a specific criterion label, i.e., C1, C2, . . , C29, C30, the criterion information box on the upper right corner of the screen will show the name of the particular criterion, while the background color of the label itself will change to green. It will also do the same thing when the user clicks on the ***Prior*** or ***Next*** button to see the name of a certain criterion.

For this example, as shown in Figure 12, the criterion information box is displaying the name of the first criterion which is independence. Label C1 is colored green. The reason that the ***Prior*** button is not enabled is because the criterion information box is displaying the name of the first criterion. When clicking on another label or the ***Next*** button, the ***Prior*** button will become enabled for the user to get the name of the prior criterion.

The scales of each criterion are decided using the following rules:

- The scale for excellent performance is given the value of criterion weight.
- The scale for weak performance is 0

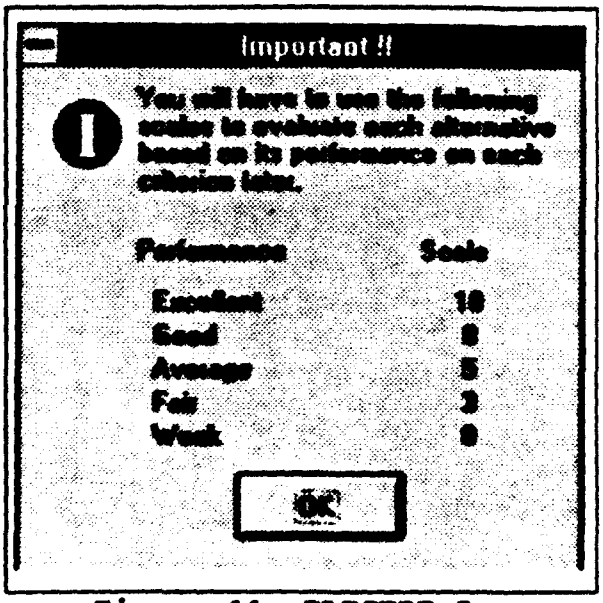


Figure 11. ELECTRE I - Standard Scaling Screen

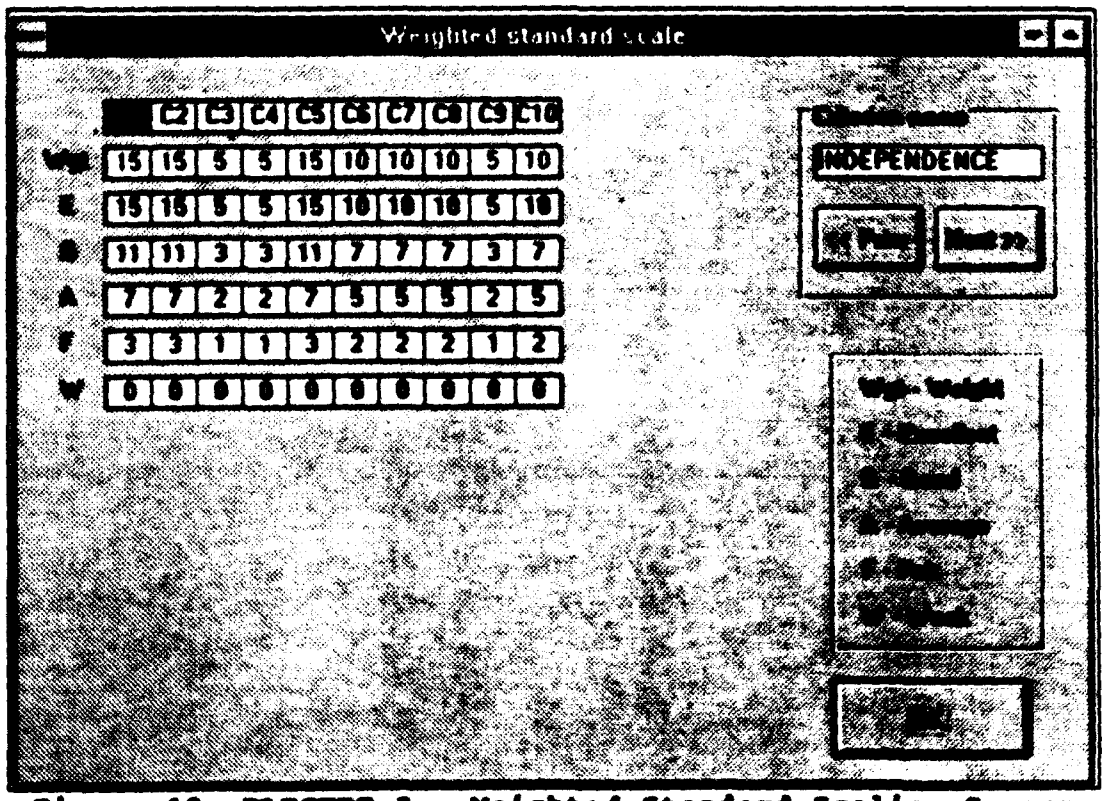


Figure 12. ELECTRE I - Weighted Standard Scaling Screen

- The difference, D, is equal to the value of weight divided by four, i.e., D = weight/4.

- If the scales for good, average, and fair performance are denoted as S_{Good}, S_{Ave}, and S_{Fair} respectively, then

$$S_{Good} = \text{weight} - D$$

$$S_{Ave} = \text{weight} - 2 * D$$

$$S_{Fair} = \text{weight} - 3 * D$$

The user will not be able to change the scales since each entry in this scale table is a label which does not accept any input. The ***OK*** button will close this screen and return the user to the ELECTRE I main screen.

c. *The Free Scale Screen*

Figure 13 shows the screen display when the user selects the ***Define your own scale*** option. This option provides the convenience for users who would like to define his own scales for the criteria considered. Controls used on this screen are labels, spin buttons, and command buttons. The user can define the desired scale by clicking the spin-up or spin-down button. Clicking the spin-up button will increase the scale by one, while clicking the spin-down button will decrease the scale by one. Since each screen can contain only fifteen criteria, a problem with more than fifteen evaluation criteria will be displayed on two screens. The ***Prior Criterion Page*** and the ***Next Criterion Page*** buttons allow the user to go to previous or next screen. The ***OK*** button and the ***Cancel*** both will close this screen and return the user to ELECTRE I main screen, but the user will be ready to evaluate alternatives when the ***OK*** button is clicked.

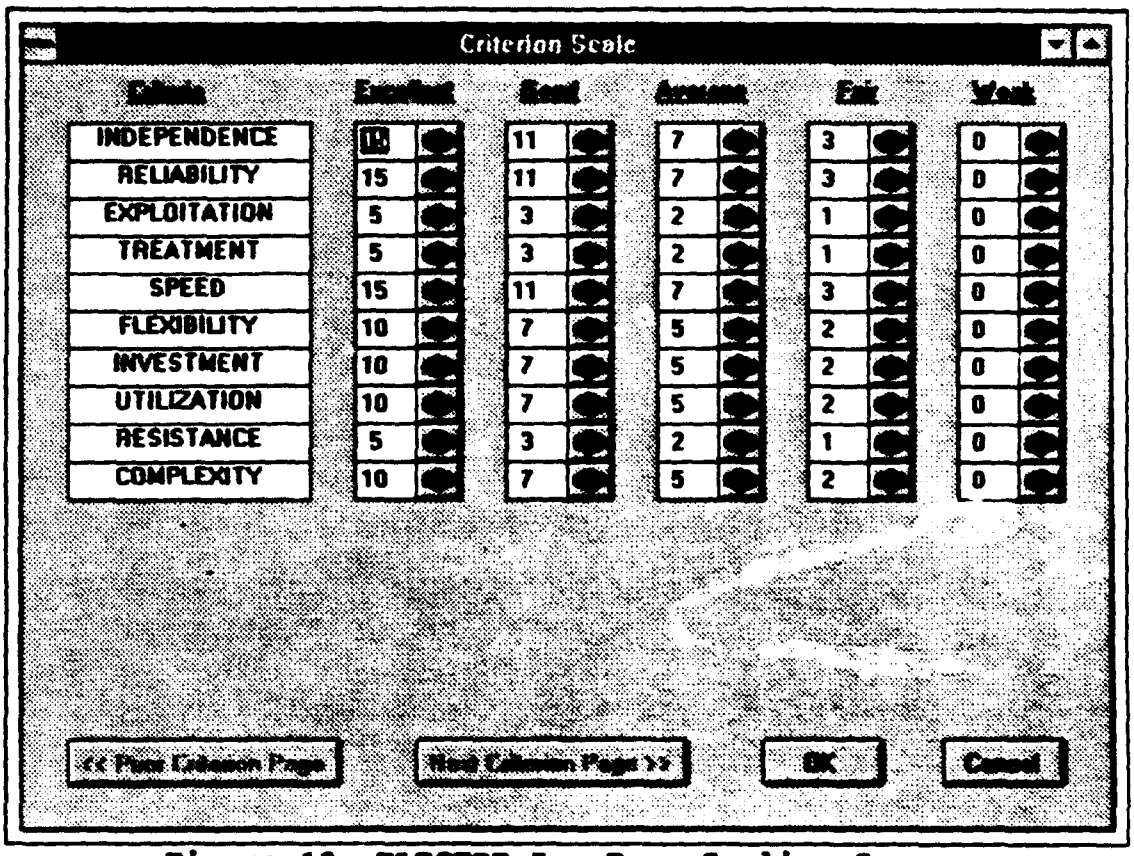


Figure 13. ELECTRE I - Free Scaling Screen

For this example, as shown in Figure 13, there are only ten criteria. Since only one criterion page is needed, the ***Prior Criterion page*** and ***Next Criterion Page*** are not enabled. The user can easily specify a scale for each criterion by clicking the spin button. For the further discussion of this example, this scale method is selected.

3. The Alternatives Evaluation Screen

Figure 14 shows the screen the user sees when he clicks on the ***Evaluate Alternatives*** button on the ELECTRE I main screen. Controls used on this screen are frame, labels, text boxes, and command buttons.

The user can evaluate alternatives only after the criterion scales have been defined. When the user is evaluating alternatives based on certain criterion, the criterion information is provided on the top right of the screen to help the user. The ***Prior Criterion Page*** and the ***Next Criterion Page*** buttons allow the user to go one page backward or forward at a time. The ***OK*** button and the ***Cancel*** both will close this screen and return the user to ELECTRE I main screen, but the user will be ready to go to next step when the ***OK*** button is clicked.

For this example, as shown in Figure 14, the user has input the evaluation of each alternative with the reference of criterion information provided.

4. The Threshold Input Screen

For alternative A to outrank alternative B, A's outranking relation with respect to B must satisfy both concordance and discordance requirements (thresholds), i.e., the concordance coefficient must be greater than or equal to the concordance threshold, and

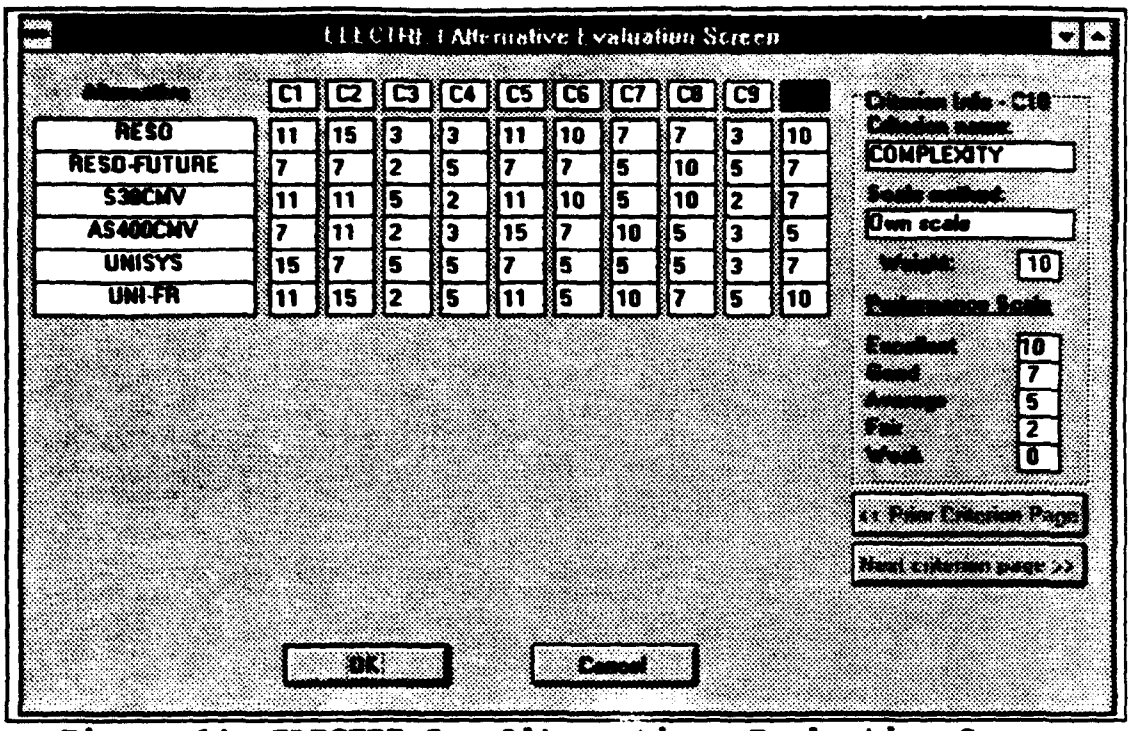


Figure 14. ELECTRE I - Alternatives Evaluation Screen

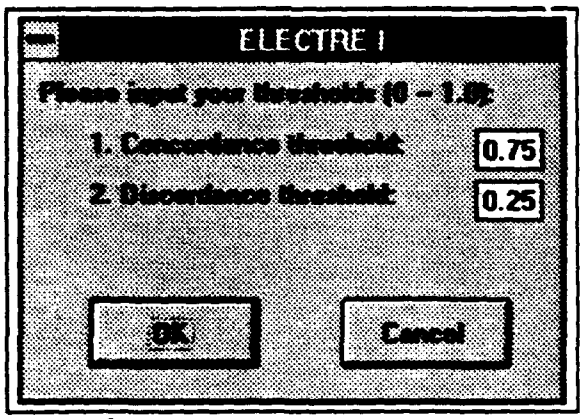


Figure 15. ELECTRE I - Threshold Input Screen

the discordance coefficient must be less than or equal to the discordance threshold. The concordance and the discordance threshold should be defined before analyzing the results. The default value is 0.75 for concordance threshold and 0.25 for discordance threshold. The default thresholds are used for this example.

Figure 15 shows the screen display when the user clicks on the ***Input Thresholds*** button on the ELECTRE I main screen. The controls used on this screen are labels, text boxes, and command buttons. The input threshold must be between 0 and 1. The ***OK*** button and the ***Cancel*** both will close this screen and return the user to ELECTRE I main screen, but the user will be ready to examines the results when the ***OK*** button is clicked.

5. The Outranking Matrix Screen

Figure 16 shows the screen the user sees when he clicks on the ***Examine Results*** button. It is the result of the example solved by ELECTRE I technique. Controls used on this screen are frames, labels, text boxes, and command buttons.

Each entry in this matrix represents an outranking relation. An outranking relation 1 is the one that satisfies both concordance and discordance requirements (thresholds), while 0 is the one that does not satisfy both or any of the concordance and discordance requirements. The labels on the top of the matrix only display first three characters of the alternatives. They are arranged in the same order as of the alternatives on the left of the matrix.

The frame labeled ***Outranking analysis*** that can be found at the lower right corner of the screen allows the user to change the concordance or the discordance

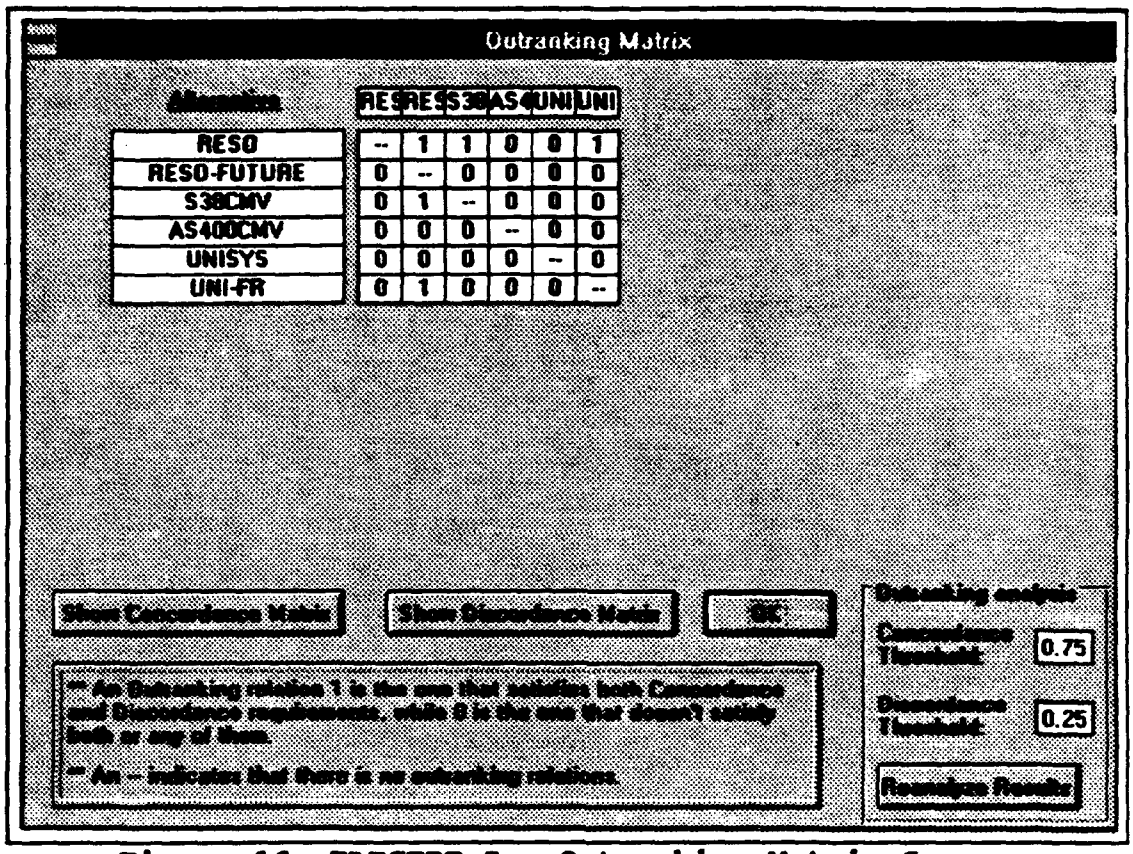


Figure 16. ELECTRE I - Outranking Matrix Screen

thresholds or both so that recomputation can be carried out based on the change. The results will be recalculated according to the user's change when he clicks on the ***Reanalyze Results*** button. The ***Show Concordance Matrix*** and ***Show Discordance Matrix*** buttons allow the user to review the concordance matrix or the discordance matrix. The ***OK*** button will close this screen and return the user to ELECTRE I main screen.

a. The Concordance Matrix Screen

Figure 17 shows the information screen display when the user clicks on the ***Show Concordance Matrix*** button on the outranking matrix screen. Controls used on this screen are frames, labels, and command button. Notice that there is no text box on this screen design, since the user is not expected to change the matrix. Each entry in this matrix contains the concordance coefficient. The concordance threshold shown at the lower right corner of this screen can not be changed by the user. However, the user can change it on the outranking matrix screen. The ***OK*** button on this screen will close the screen and return the user to the outranking matrix screen.

b. The Discordance Matrix Screen

Figure 18 shows the information screen the user sees when he clicks on the ***Show Discordance Matrix*** button on the outranking matrix screen. This screen basically shares the same design with the concordance matrix screen. Each entry in this matrix contains the discordance coefficient.

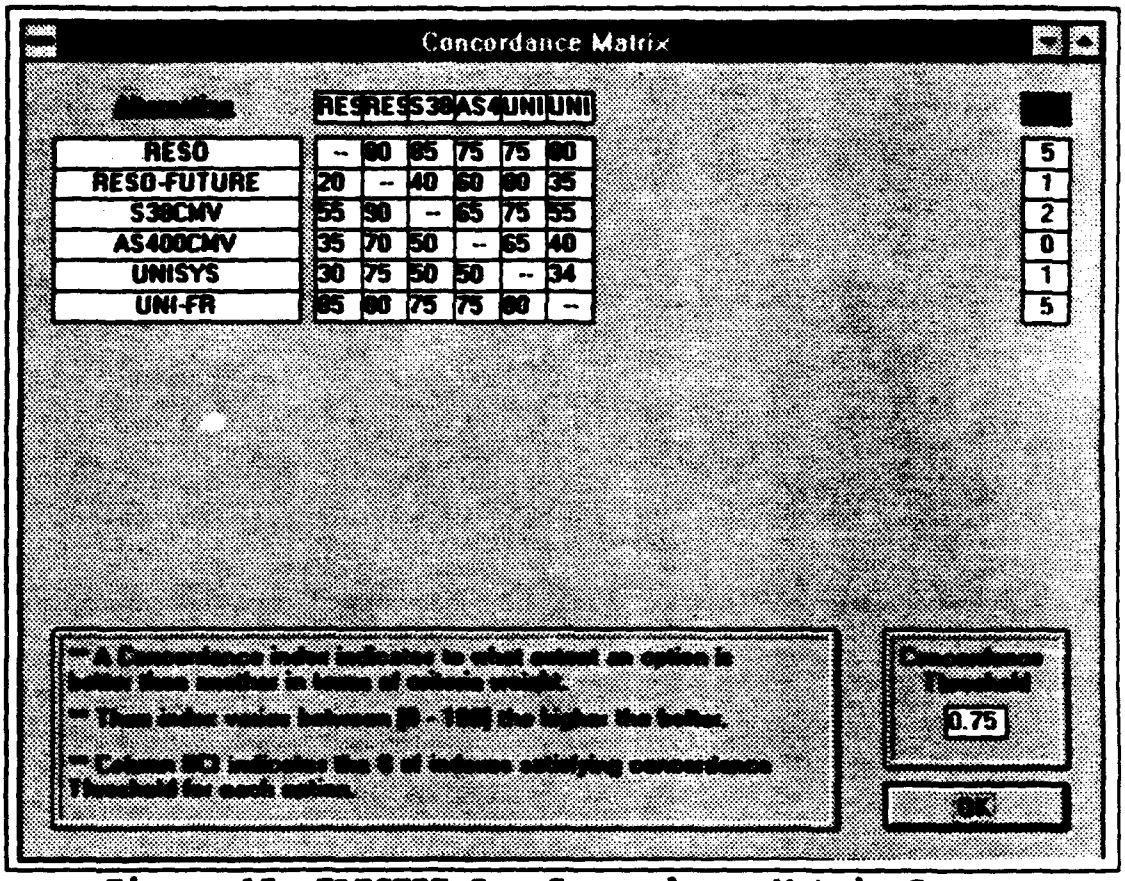


Figure 17. ELECTRE I - Concordance Matrix Screen

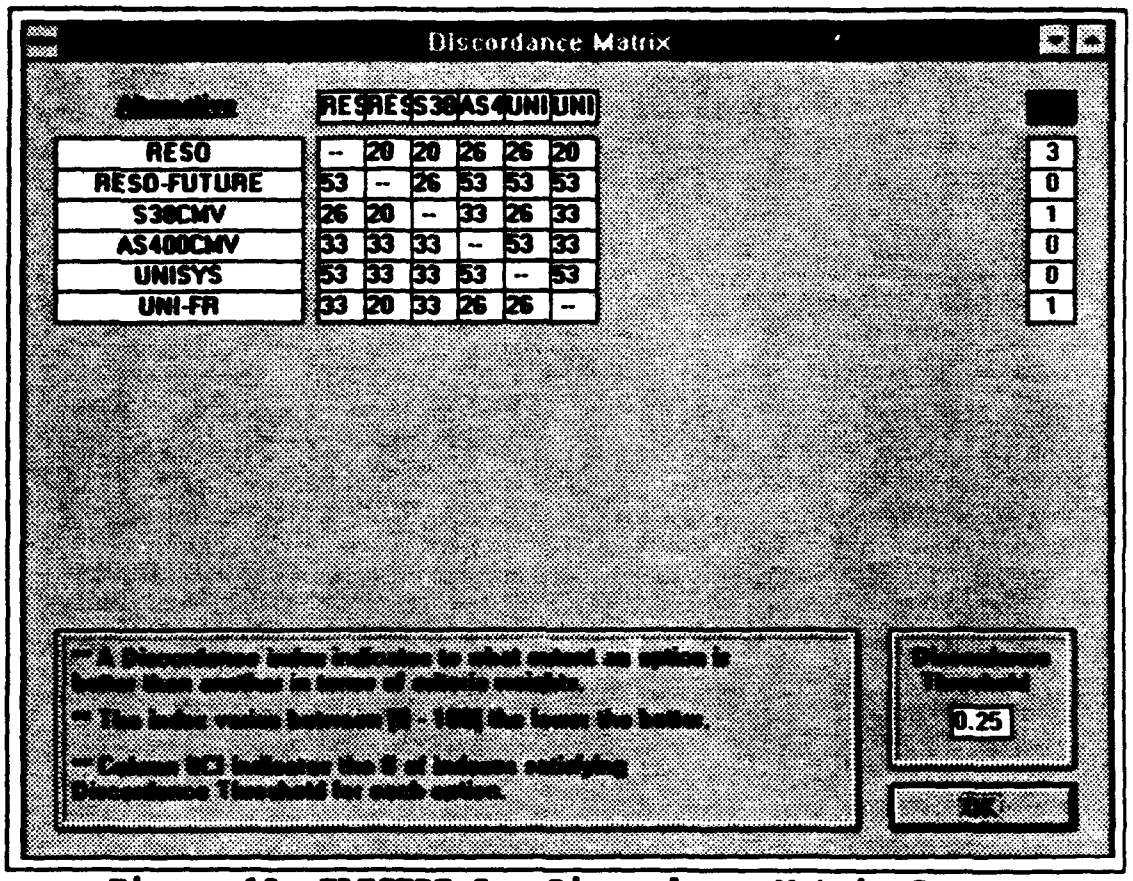


Figure 18. ELECTRE I - Discordance Matrix Screen

III. PROTOTYPE 2: ELECTRE III

A. THE ALGORITHM

Basically, ELECTRE I and ELECTRE III share the same underlying concepts, i.e., the notions of concordance and discordance. They are both developed by Bernard Roy. ELECTRE I, discussed in the previous chapter, is designed to help decision makers choose the appropriate alternatives among a set of alternatives. The outranking principle of ELECTRE I does not lend itself well to a problem that seeks to rank alternatives. ELECTRE III is an algorithm that is suited to cover multiple criteria analysis of problem of arrangement and classification. (Roy, 1978)

Another difference between ELECTRE I and ELECTRE III is the definition and use of thresholds. In ELECTRE I, there are two thresholds which are the concordance threshold and discordance threshold, while there are three thresholds for each evaluation criterion and a discrimination threshold in ELECTRE III.

Three thresholds for each criterion are indifference threshold, preference threshold, and veto threshold. They are denoted as SI, SP, and SV, respectively. Let the difference between the user's evaluation scores of any two alternatives be simply mentioned as difference. If this difference is less than or equal to the indifference threshold, the difference is considered insignificant. This means that these two alternatives are no different from each other in terms of preference. If the difference is greater than indifference threshold and is less than or equal to the preference threshold, the alternative

with higher evaluation score is considered weakly preferential. If the difference is greater than the preference threshold and is less than or equal to the veto threshold, the alternative with higher evaluation score is strongly preferred. If the difference is greater than the veto threshold, the alternative with higher evaluation score is exactly better. Eventually, the discrimination threshold plays the role of final indifference threshold to filter out negligible difference between any pair of alternatives.

The algorithm is briefly presented below. (Roy, 1978)

(i) the index of credibility $d_i(A/B)$

$$d_i(A/B) = \frac{SP_i - \min\{(B_i - A_i)\ ,\ SP_i\}}{SP_i - \min\{(B_i - A_i)\ ,\ SI_i\}}$$

(ii) the index of concordance C(A/B)

$$C(A/B) = \sum_{i=1}^{n} p_i * d_i(A/B) \quad \text{where } p_i = \text{weight of criterion } i$$

(iii) the index of discordance $D_i(A/B)$

$$D_i(A/B) = \min\ [1,\ \max\{0,\ \frac{(B - A) - SP}{SV - SP}\}] \quad \text{where } SV \geq SP$$

(iv) the global index of credibility d(A/B)

$$\text{if } D_i(A/B) > C(A/B) \text{ then} \quad d(A/B) = C(A/B) * \frac{1 - D_i(A/B)}{1 - C(A/B)}$$

$$\text{if } C(A/B) > D_i(A/B) \text{ then} \quad d(A/B) = C(A/B) * \frac{1 - C(A/B)}{1 - D_i(A/B)}$$

where A, B are alternatives to be compared, subscript i is the criterion considered, SI is the indifference threshold, SP is the preference threshold, and SV is the veto threshold.

When the global indices of credibility are decided, the final process is to rank the alternatives. There are three ways to rank the alternatives. The first one is descending distillation which is to first find the best alternative and finish with the worst alternative. The second one is ascending distillation which is to first find the worst alternative and finish with the best alternative. The last one is actually the combination of the results from the previous two methods. The detailed analysis of the ELECTRE III algorithm is not the intent of this thesis, so it would not be discussed further in this thesis.

B. SCREEN DESIGN OF ELECTRE III

In ELECTRE III, it is not necessary to define the criterion scales for criteria. With the exception of the first problem solving step using in ELECTRE I, the basic concepts and the problem solving methodology of ELECTRE III are similar to that of ELECTRE I. Two screen designs of ELECTRE III are inherited from ELECTRE I, i.e., the screen designs of the ELECTRE III main screen and the alternative evaluation screen (Figure 9 and Figure 14). When the ELECTRE III main screen is opened, the ***Choose a Scaling Method*** button is not enabled since the criterion scales are not needed. The rest of the designs will be discussed in this section. The example used in this chapter is same as in the previous chapter.

1. The Threshold Input Screen

Figure 19 shows the screen display when the user clicks on the ***Input Thresholds*** button on the ELECTRE III main screen (See Figure 4). This screen requests the necessary input from the user to solve a problem using ELECTRE III algorithm.

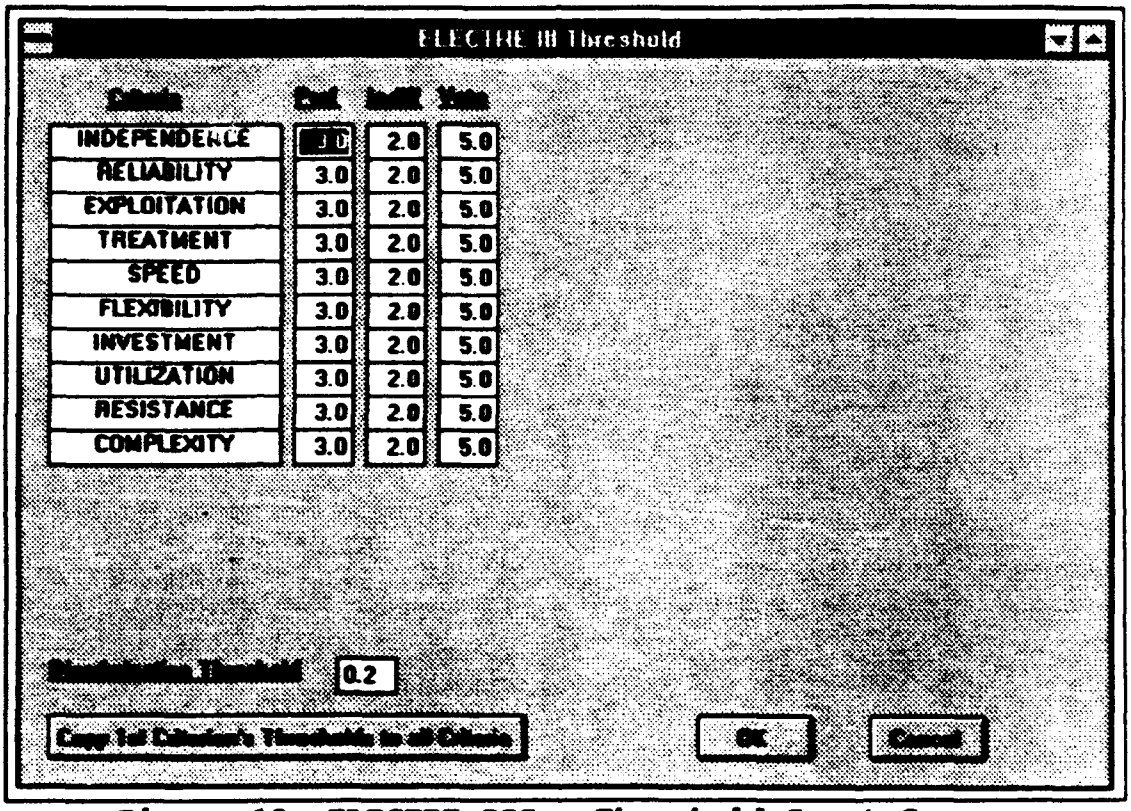


Figure 19. ELECTRE III - Threshold Input Screen

Controls used on this screen are labels, text boxes, and command buttons. If the user prefers to use the same preference, indifference, and veto thresholds, he can click on the ***Copy 1st Criterion's Thresholds to all Criteria*** button. The text box which is right on the top of this button will allow user to input the discrimination threshold. The default value of the discrimination threshold is 0.2. Both the ***OK*** and the ***Cancel*** buttons will close this screen and return to ELECTRE III main screen, but the user will be ready to go to next step when the ***OK*** button is pressed.

2. The Result Screen

Figure 20 shows the screen the user sees when he clicks on the ***Examine Results*** button in the ELECTRE III main screen. This screen displays the final result of the problem solving using ELECTRE III technique. The cardinal ranking is used to show the computation results of the problem, while the three ordinal rankings is to show the ranking of the alternatives using ascending distillation method, descending distillation method, and the combination result of the previous two methods. The ***OK*** button is to close this screen and return the user to ELECTRE III main screen. If the user prefers to examine the results, he can simply click on the ***Show Credit Matrix*** button which will open the credit matrix screen.

3. The Credit Matrix Screen

Figure 21 shows the screen display when the user clicks on the ***Show Credit Matrix*** button on the ELECTRE III result screen. Controls used on this screen are frame, labels, and command button.

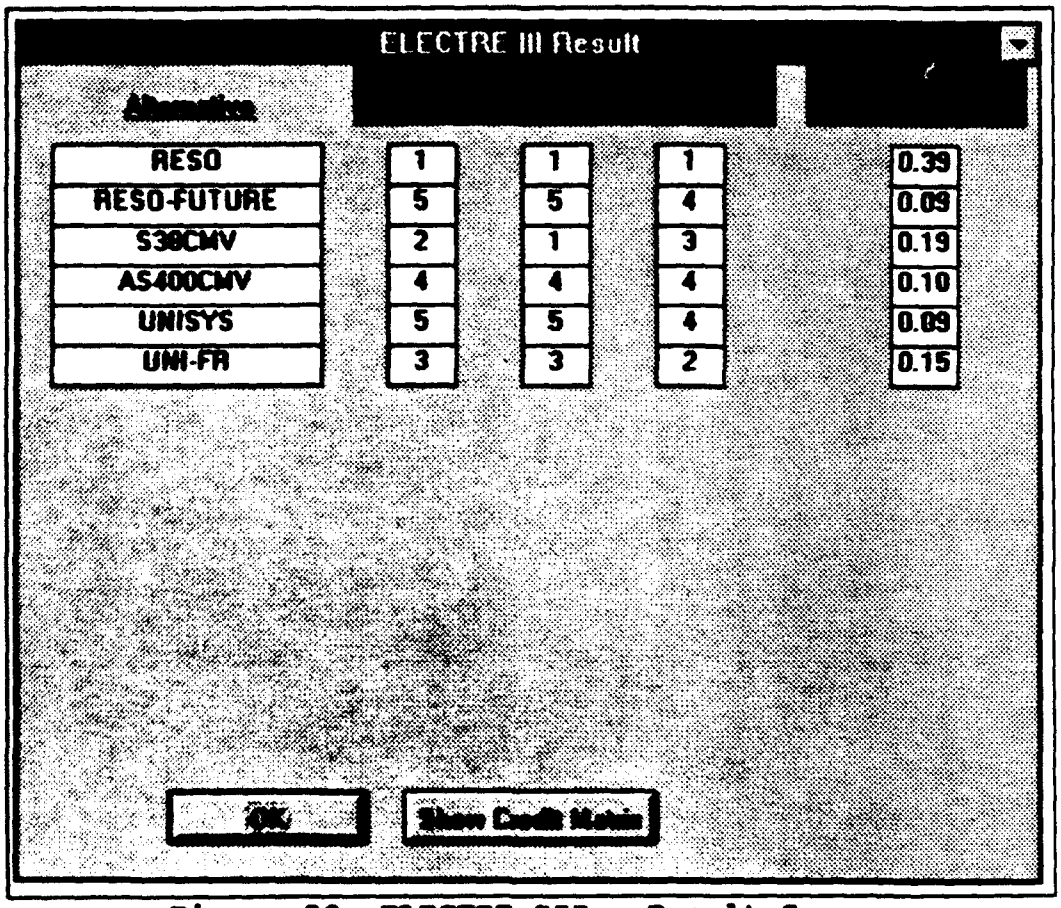
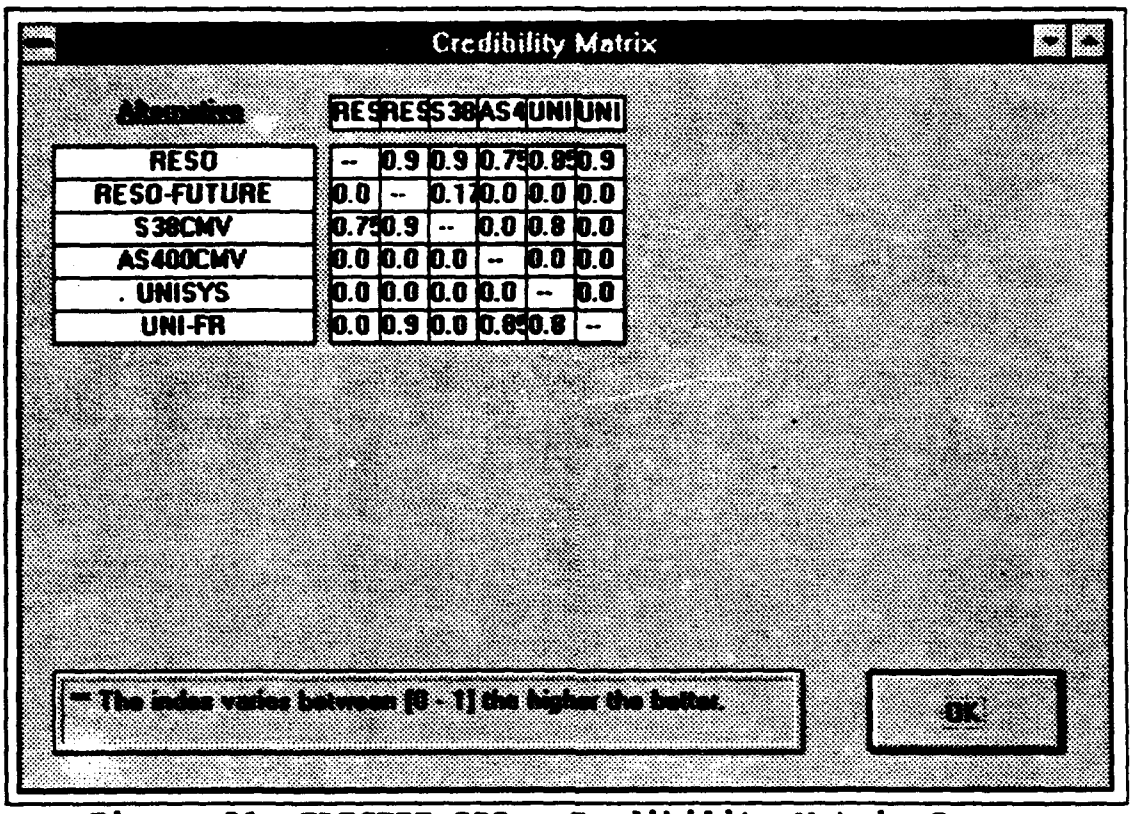


Figure 20. ELECTRE III - Result Screen



Credibility Matrix

Alternative	RES	RES	S38	AS4	UNI	UNI
RESO	--	0.9	0.9	0.75	0.85	0.9
RESO-FUTURE	0.0	--	0.17	0.0	0.0	0.0
S38CMV	0.75	0.9	--	0.0	0.8	0.0
AS400CMV	0.0	0.0	0.0	--	0.0	0.0
. UNISYS	0.0	0.0	0.0	0.0	--	0.0
UNI-FR	0.0	0.9	0.0	0.85	0.8	--

-- The index varies between [0 - 1] the higher the better.

OK

Figure 21. ELECTRE III - Credibility Matrix Screen

Each entry in this matrix is an outranking relation. The credibility shown on each entry states the confidence level of that specific outranking relation. Its value varies from 0 to 1, with 1 being the best. The ***OK*** button closes this screen and returns the user to the ELECTRE III result screen.

IV. PROTOTYPE 3: ELECTRE IV

A. THE ALGORITHM

When the user doesn't want or doesn't need to give a weighing function to the criterion family, it become impossible to build a concordance matrix by gathering partial preference. ELECTRE IV uses four outclassing relations (S_q, S_c, S_p, and S_v) between any pair of alternatives to build a blurred outclassing relation over the entire alternative group. This technique is also developed by the LAMSADE laboratory. (Roy, 1992)

The use of these four outclassing relation iay on two main ideas:

- No criteria is more important giving a regrouping of any one half of the criteria.
- No criteria is negligible giving a regrouping of any one half of the criteria.

Actually, the algorithm of ELECTRE IV is very close to that of ELECTRE III. In ELECTRE III, every criterion has three thresholds, namely indifference threshold, preference threshold, and veto threshold. In ELECTRE IV, every criterion also has the same three thresholds, but each of these three thresholds has two coefficients, namely alpha and beta. The user also has to define the mean of threshold calculation which is either direct or indirect for each criterion, and whether the criterion is to be maximized or minimized. Another difference is in the way of calculating the global index of credibility.

The algorithm is presented briefly below. (Roy, 1992)

Notations:

- $m_p(a,b)$: the number of criteria for which alternative a is strictly in favor to alternative b.

- $m_q(a,b)$: the number of criteria for which alternative a is weakly in favor to alternative b.

- $m_i(a,b)$: the number of criteria for which alternative a is indifferent to alternative b.

- $m_o(a,b) = m_o(b,a)$: the number of criteria for which alternative a has the same evaluation as of alternative b.

- d(a,b): the global index of credibility for alternative a outclasses alternative b.

- T(i): indicates whether the criterion i is to be maximized or minimized (Max or Min).

- M(i): indicates the mean of calculation for criterion i (Direct or Indirect).

For any pair of alternatives {a,b}:

$$m = m_p(a,b) + m_q(a,b) + m_i(a,b) + m_o(a,b) + m_p(b,a) + m_q(b,a) + m_i(b,a)$$

where m is the total number of criteria.

Four outclassing relations:

- Quasi-dominant: S_q

  If $m_p(b,a) + m_q(b,a) = 0$ and $m_i(b,a) <= 1 + m_i(a,b) + m_q(a,b) + m_p(a,b)$ then, a S_q b and d(a,b) = 1

- Canonical dominant: S_c

  If $m_p(b,a) = 0$ and $m_q(b,a) <= m_p(a,b)$ and $m_q(b,a) + m_i(b,a) <= 1 + m_i(a,b) + m_q(a,b) + m_p(a,b)$ then, a S_c b and d(a,b)=0.8

- Pseudo-dominant: S_p

  If $m_p(b,a) = 0$ and $m_q(b,a) <= m_q(a,b) + m_p(a,b)$ then, a S_p b and d(a,b)=0.6

- Veto-dominant: S_v

Let $v_i[g_i(a)]$ be the veto threshold for alternative a, if $g_i(b) >= g_i(a) + v_i[g_i(a)]$ then **b** PV_i **a**. If **b** PV_i **a**, then the priority to b is strongly affirmed to give a veto to "a outclass b" for any other preference for a. The veto-dominant S_v is defined by the following:

If (not **b** PV_i **a** for all i and $m_p(b,a) = 0$) or (not **b** PV_i **a** and $m_p(b,a) = 1$ and $m_p(a,b) >= m/2$) then, a S_v b and d(a,b)=0.35

When the global indices of credibility are found, the same ranking methods as in the ELECTRE III will be used to rank the alternatives.

B. NUMERIC EXAMPLE

The example to be used for this algorithm is a case of car purchasing. There are eight cars considered which will be evaluated on four criteria. These criteria are:

- gas consumption at 120km/h
- price ($: in thousand dollars)
- maximum speed
- interior space

The following data is needed to construct the outranking relations using ELECTRE IV:

- the type of criterion (Max or Min)

1. gas consumption: Min
2. price: Min
3. max-speed: Max
4. int. space: Max

- the mean of calculation (Direct or Indirect)

1. gas consumption: Direct
2. price: Indirect
3. max-speed: Direct
4. int. space: Indirect

- the indifference threshold (alpha/beta)

1. gas consumption: 0.05/0.10
2. price: 0.02/0.5
3. max-speed: 0/5

4. int. space: 0.05/0

- the preference threshold (alpha/beta)

1. gas consumption: 0.10/0.10
2. price: 0.10/1
3. max-speed: 0.10/0
4. int. space: 0.10/0

- the veto threshold (alpha/beta)

1. gas consumption: 0.15/0.20
2. price: 0.15/2
3. max-speed: 0.10/10
4. int. space: 0.10/0.4

- the evaluations

	Gas cons.	price	max-speed	space
VW GOLF C	8	41	140	6
R9 GTL	8	45	150	7
GSA X1	7	46	160	8
P305 GR	8	48	153	8
TALBOT	8	49	164	7
AUDI 80CL	7	51	148	7
R18 GTL	8	52	155	6
ALPHA SUD	9	53	170	7

Given these data, the user should be able to define this problem using the problem defining procedures discussed in Chapter III.

C. SCREEN DESIGNS OF ELECTRE IV

Most of the screen designs of ELECTRE IV are the same as of ELECTRE III except the threshold input screen. On the threshold input screen, the controls used are labels, check boxes, text boxes, and command buttons. The selected example is solved by going through alternative evaluation, threshold input, and finally the ranking process. These screens are shown in Figures 22, 23, 24, and 25.

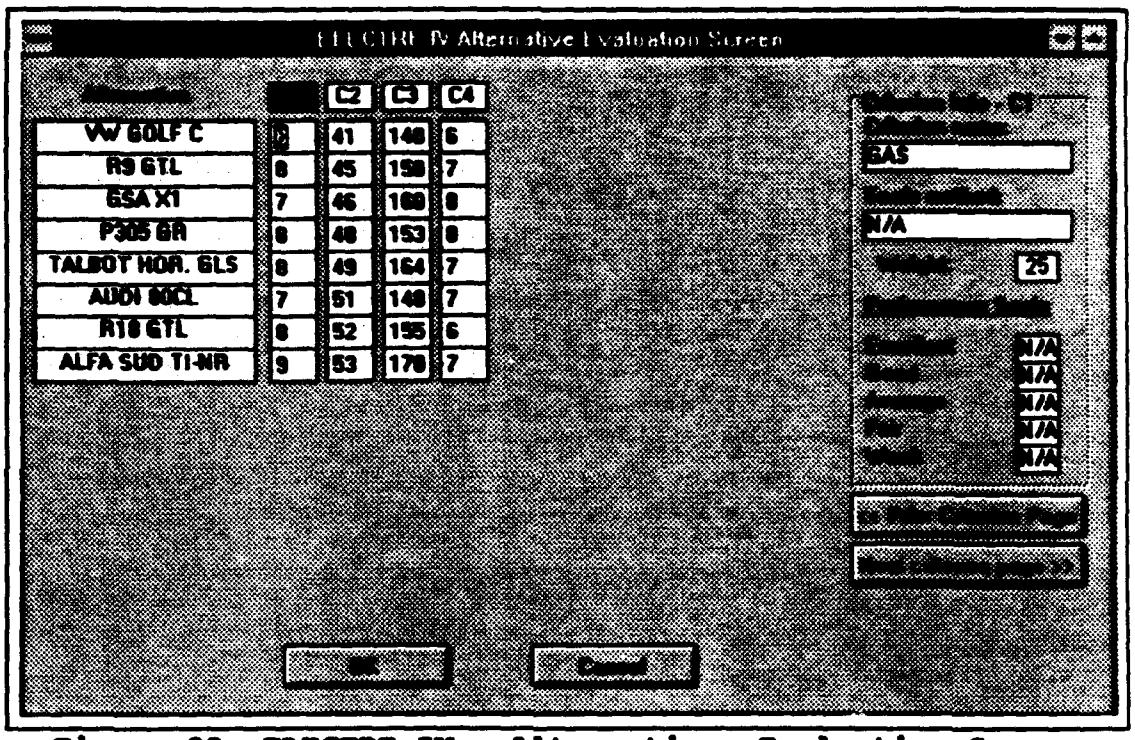


Figure 22. ELECTRE IV - Alternatives Evaluation Screen

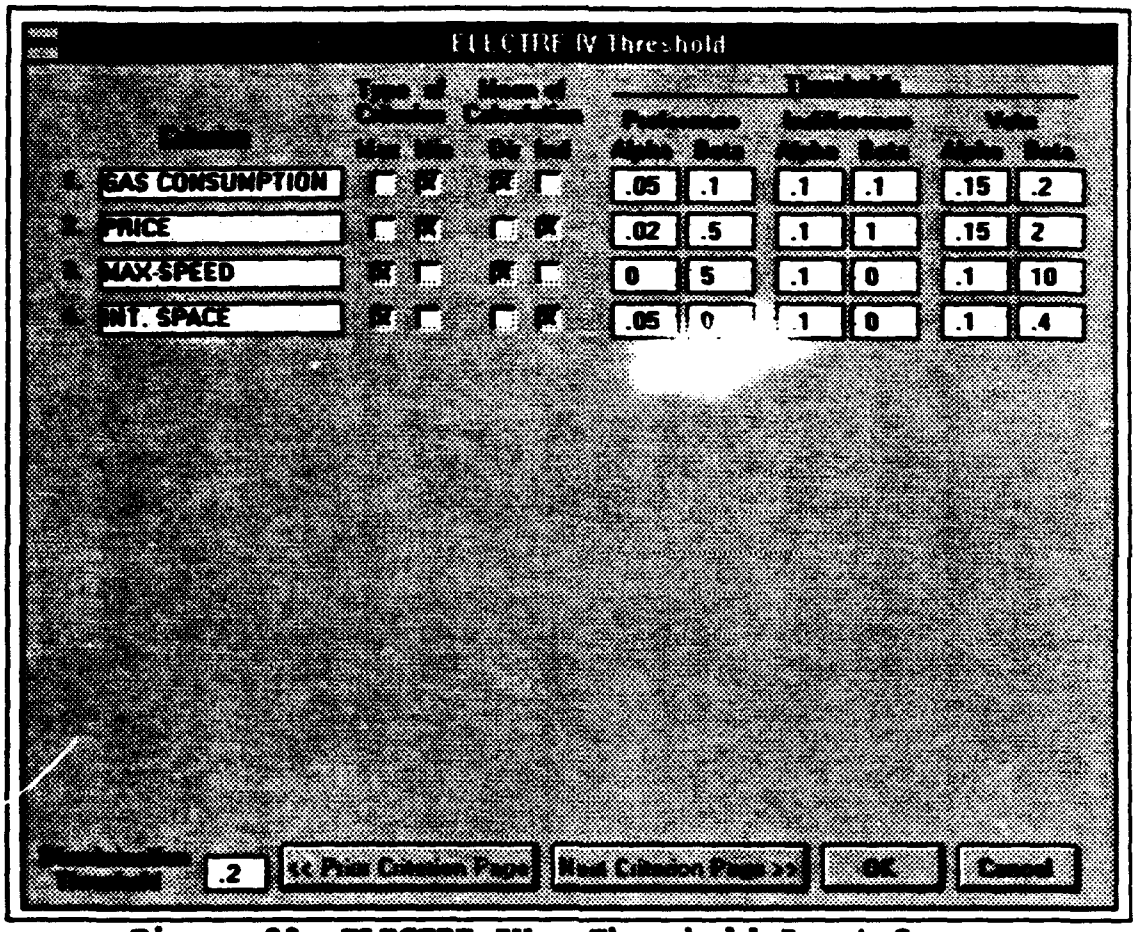


Figure 23. ELECTRE IV - Threshold Input Screen

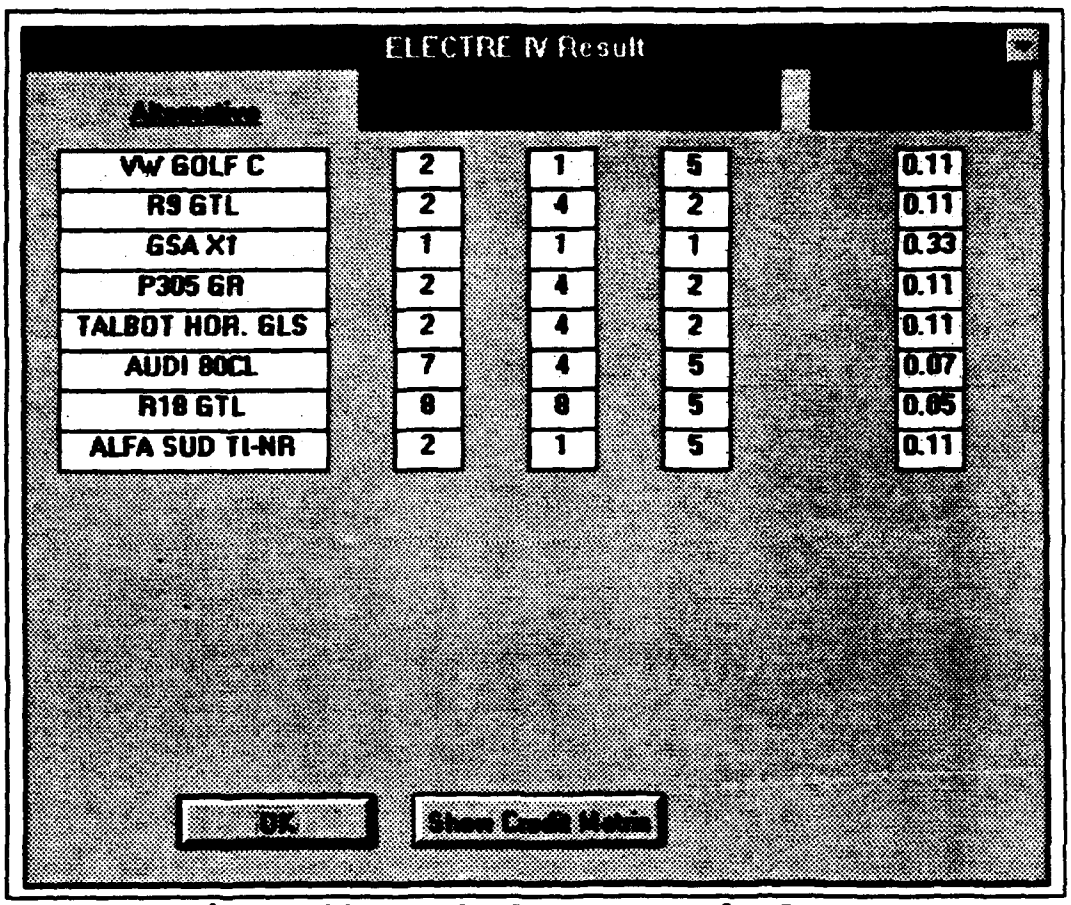
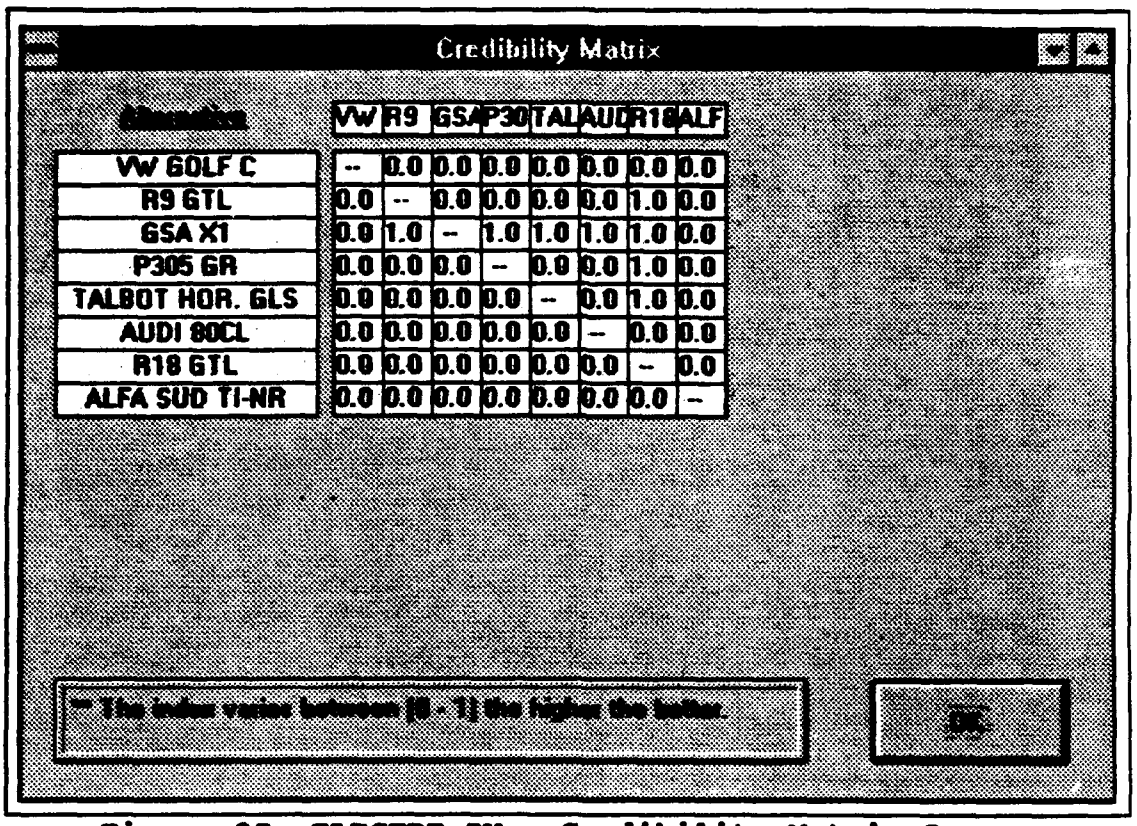


Figure 24. ELECTRE IV - Result Screen



Credibility Matrix

Alternatives	VW	R9	GSA	P30	TAL	AUD	R18	ALF
VW GOLF C	--	0.0	0.0	0.0	0.0	0.0	0.0	0.0
R9 GTL	0.0	--	0.0	0.0	0.0	0.0	1.0	0.0
GSA X1	0.0	1.0	--	1.0	1.0	1.0	1.0	0.0
P305 GR	0.0	0.0	0.0	--	0.0	0.0	1.0	0.0
TALBOT HOR. GLS	0.0	0.0	0.0	0.0	--	0.0	1.0	0.0
AUDI 80CL	0.0	0.0	0.0	0.0	0.0	--	0.0	0.0
R18 GTL	0.0	0.0	0.0	0.0	0.0	0.0	--	0.0
ALFA SUD TI-NR	0.0	0.0	0.0	0.0	0.0	0.0	0.0	--

-- The index varies between [0 - 1] the higher the better.

OK

Figure 25. ELECTRE IV - Credibility Matrix Screen

V. PROTOTYPE 3: PROMETHEE

A. THE ALGORITHMS

PROMETHEE (Preference Ranking Organization METHods for Enrichment Evaluations) is a family of outranking methods in multicriteria analysis based on a generalization of the notion of criterion. This family includes PROMETHEE I, PROMETHEE II, PROMETHEE III, PROMETHEE IV, and PROMETHEE V. To solve a multicriteria problem using these methods, a fuzzy outranking relation is first built. This relation is then used to set up a partial preorder (PROMETHEE I) or a complete preorder (PROMETHEE II) or an interval order (PROMETHEE III) on a finite set of feasible solutions. These results can be easily apprehended by the decision-maker. PROMETHEE IV is developed to solve multicriteria problems with continuous feasible solutions (Brans, 1984). PROMETHEE V is used to solve multicriteria problems with a finite set of possible alternatives grouped in clusters or segments. In other words, PROMETHEE V is used to select a subset of alternatives evaluated by several criteria and submitted to segmentation constraints between and within the clusters. (Brans, 1992)

PROMETHEE I, PROMETHEE II, and PROMETHEE V are the three most commonly used methods among the PROMETHEE family. Hence, there are implemented in the proposed Co-op system.

1. Basic Algorithm of PROMETHEE I and PROMETHEE II

PROMETHEE I and PROMETHEE II actually share the same algorithm. After the alternatives (or the possible actions) and evaluation criteria of a multicriteria problem have been defined, the user has to further identify what type of generalized criterion each criterion is to use, and decide on the value of each parameter of every generalized criterion used.

The key of this algorithm is to calculate the **phi+**, which is a measure of the outranking characteristic of an alternative, and **phi-**, which is a measure of the outranked character of an alternative, for every alternative. Finally, **phi** which is the result of **phi+** minus **phi-** is used to decide the outranking relation.

Consider following multicriteria problem:

Max $\{f_1(a), f_2(a), ..., f_h(a), ..., f_k(a) \mid a \in A\}$ where A is a set of possible actions and f_h, h = 1,2,...,k are k evaluation criteria to be maximized, while $f_h(a)$ is the evaluation for the action a based on criterion f_h.

To solve this problem with PROMETHEE methods, the user has to follow three phases. These three phases are:

1. Construction of generalized criteria.
2. Determination of an outranking relation on A.
3. Evaluation of this relation in order to give an answer to this problem.

In the first phase a generalized criterion is associated to each of these k evaluation criteria by considering a preference function. In the second phase, a multicriteria preference index is defined in order to obtain a valued outranking relation

representing the preferences of the decision maker. The evaluation of the outranking relation is obtained by considering for each action a **phi+** and a **phi-**.

Let f(a) be a criterion to be maximized. For each action $a \in A$, f(a) is an evaluation of this action. When two actions $a \in A$ and $b \in A$ are compared with respect to this criterion the result of the comparison has to be expressed in terms of preferences. Let P(a,b) be the preference function that represents the result of this comparison, giving the intensity of preference of the action a over the action b. If P(b,a) is the case, the converse is true. The function P(a,b) has the following meaning:

- P(a,b) = 0 No preference of a over b, indifference between a and b
- P(a,b) ~ 0 Weak preference of a over b (f(a) > f(b)),
- P(a,b) ~ 1 Strong preference of a over b (f(a) >> f(b)),
- P(a,b) = 0 Strict preference of a over b (f(a) >>> f(b)).

Let d be the difference between f(a) and f(b), i.e., d = f(a) - f(b), and H(d) be the function of the generalized criterion associated to the criterion f. P(a,b) or P(b,a) can be obtained by the following:

(i) If d >= 0 ,which means f(a) >= f(b), then P(a,b) = H(d).

(ii) If d <= 0, which means f(b) >= f(a), then P(b,a) = H(d).

The function H(d) is defined as (also refers to Figure 20 for the shape of each generalized criterion type):

(i) Usual criterion

a. if $d = 0$, then $H(d) = 0$,

b. if $|d| > 0$, then $H(d) = 1$.

(ii) Quasi criterion

a. if $|d| <= q$, then $H(d) = 0$,

b. otherwise, $H(d) = 1$.

(iii) Criterion with linear preference

a. if $|d| <= q$, then $H(d) = |d|/p$,

b. otherwise, $H(d) = 1$.

(iv) Level criterion

a. if $|d| <= q$, then $H(d) = 0$,

b. if $q < |d| <= p$, then $H(d) = 1/2$,

c. otherwise, $H(d) = 1$.

(v) Criterion with linear preference and indifference area

a. if $|d| <= q$, then $H(d) = 0$,

b. if $q < |d| <= p$, then $H(d) = (|d| - q)/(p - q)$,

c. otherwise, $H(d) = 1$.

(vi) Gaussion criterion

$H(d) = 1 - \exp\{-d^2/(2*Ó^2)\}$

where q is an indifference threshold and is the largest value of d below which the decision maker considers there is indifference, while p is a strict preference threshold and is the lowest value of d above which the decision-maker considers there is preference, and Ó is a well known parameter directly connected with the standard deviation of a normal distribution.

Suppose a preference function P_h has been defined for each criterion f_h, h = 1, 2, ..., k. For each couple of actions a, b $\in$ A, let:

$$\pi(a,b) = \frac{1}{k} \sum_{h=1}^{k} P_h(a,b) .$$

π(a,b) is a preference index over all the criteria. It is easy to see that π(a,b) is simply the mean of the values of the k associated preference functions. So it is true that

$0 <= \pi(a,b) <= 1$,

and moreover

- $\pi(a,b) \sim 0$ denotes a weak preference of a over b,
- $\pi(a,b) \sim 1$ denotes a strong preference of a over b.

In order to evaluate the actions of A by using the outranking relation, the following measures should be considered:

(i) ϕ^+ **(phi+)**

$$\phi^+(a) = \sum_{b \in A} \pi(a,b)$$

(ii) ϕ^- **(phi-)**

$$\phi^-(a) = \sum_{b \in A} \pi(b, a)$$

(iii) ϕ **(phi)**

$$\phi(a) = \phi^+(a) - \phi^-(a)$$

a. Use of the Algorithm in PROMETHEE I

The higher the measure **phi+** and the lower the measure **phi-**, the better the action. These two measures, i.e., **phi+** and **phi-**, induce respectively the following preorders on the actions of A:

(i) a $\mathbf{P}^+$ b iff $\phi^+(a) > \phi^+(b)$,

(ii) a $\mathbf{I}^+$ b iff $\phi^+(a) = \phi^+(b)$,

(iii) a $\mathbf{P}^-$ b iff $\phi^-(a) < \phi^-(b)$,

(iv) a $\mathbf{I}^-$ b iff $\phi^-(a) = \phi^-(b)$.

P and I mean respectively preference and indifference.

PROMETHEE I gives a partial preorder ($\mathbf{P}^{(I)}$, $\mathbf{I}^{(I)}$, **R**) on the actions of A obtained by considering the intersection of these two preorders:

(i) if a $\mathbf{P}^+$ b and a $\mathbf{P}^-$ b, or a $\mathbf{P}^+$ b and a $\mathbf{I}^-$ b, or a $\mathbf{I}^+$ b and a $\mathbf{P}^-$ b, then a $\mathbf{P}^{(I)}$ b (a outranks b).

(ii) if a $\mathbf{I}^+$ b and a $\mathbf{I}^-$ b, then a $\mathbf{I}^{(I)}$ b (a is indifferent to b).

(iii) otherwise, a **R** b (a and b are incomparable).

Where the superscript (I) denotes a PROMETHEE I outranking relation.

When proposing the PROMETHEE I partial preorder to the decision-maker, some actions remain incomparable. Usually the PROMETHEE I partial preorder is richer than the dominance order and is therefore providing useful information.

b. Use of the Algorithm in PROMETHEE II

PROMETHEE II gives a complete preorder ($\mathbf{P}^{(II)}$, $\mathbf{I}^{(II)}$) induced by the measure ϕ (**phi**) and is defined as:

(i) $a\ \mathbf{P}^{(II)}\ b$ iff $\phi(a) > \phi(b)$

(ii) $a\ \mathbf{I}^{(II)}\ b$ iff $\phi(a) = \phi(b)$

Where the superscript (II) denotes a PROMETHEE II outranking relation.

It seems easier for the decision-maker to arrive at a decision problem by using the complete preorder PROMETHEE II instead of the partial one given by PROMETHEE I. But, on one hand the partial preorder provides more realistic information by considering only confirmed outrankings with respect to the measure **phi+** and the measure **phi-**. On the other hand incomparabilities can also be very useful.

2. Algorithm of PROMETHEE V

PROMETHEE V is actually used to solve multicriteria problems whose alternatives are subjected to some kind of segmentation constraints. In other words, alternatives are grouped in several clusters or segments. The decision-maker is facing the problem of selecting the best alternatives from each group.

The procedure to solve this kind of multicriteria problem includes two steps. They are:

(1) The multicriteria problem without segmentation constraints is first considered.

(2) The additional segmentation constraints are integrated to construct a (0 - 1) linear program.

The first step simply considers only the evaluation criteria of the problem. Therefore, the decision-maker can analyze the problem by PROMETHEE I and PROMETHEE II methods to see the ranking of the alternatives. In the second step, it should be noted that a (0 - 1) linear program means the value of each decision variable of a linear program's can only be 1 or 0. Each decision variable in the (0 - 1) linear program represents a specific alternative of the problem. PROMETHEE V is to solve the (0 - 1) linear program so that alternatives with value 1 are suggested choices.

B. SCREEN DESIGNS OF PROMETHEE

Since PROMETHEE V is used to solve (0 - 1) linear programs, it is designed separately from PROMETHEE I and PROMETHEE II. The PROMETHEE main screen, shown in Figure 26, is contrived to allow the user to choose either PROMETHEE I and PROMETHEE II, or PROMETHEE V to solve a multicriteria problem. The controls used on this screen are option buttons and command buttons.

The example used for PROMETHEE family is the same as in Chapter II. This example will first be processed by PROMETHEE I and PROMETHEE II, then by the PROMETHEE V.

1. PROMETHEE I and PROMETHEE II

a. The Main Screen

Figure 27 shows the screen the user sees when he choose the ***PROMETHEE I, II*** option on the PROMETHEE main screen. Controls used on this screen are picture boxes and command buttons. The flow chart shown on this screen is an exact presentation of how a multicriteria problem is to be solved using PROMETHEE I and PROMETHEE II. Design of this screen is almost same as that of ELECTRE I (see Figure 9).

b. The Criteria Identification Screen

Figure 28 shows the screen display when the user clicks on the ***Identify Criteria*** button. The design of this screen is to give the user an idea of what is needed to be defined in every criterion. This will inevitably facilitate the work of the user. Controls used on this screen are labels, check boxes, spin buttons, text boxes, and command buttons.

Some criteria are to be maximized, such as the engine performance of a new car, the processing speed of a new computer, etc, while some are to be minimized, for example, the cost of a new house, gas consumption of a new car, etc. For each criterion, it can either be maximized or minimized but not both. The design considers this point. When the user chooses ***Max***, the ***Min*** is automatically disabled, and *vice versa*. Since each evaluation criterion is associated with one of the six generalized criteria, the spin button is used since it is the most convenient way of input to the user. When the

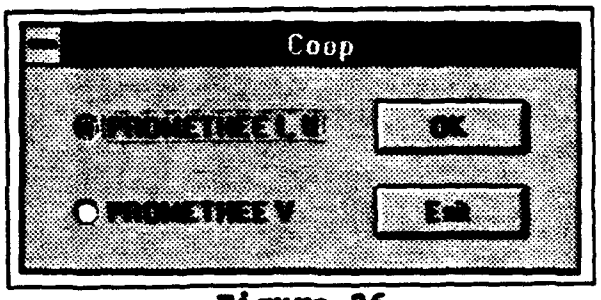


Figure 26
PROMETHEE Main Screen

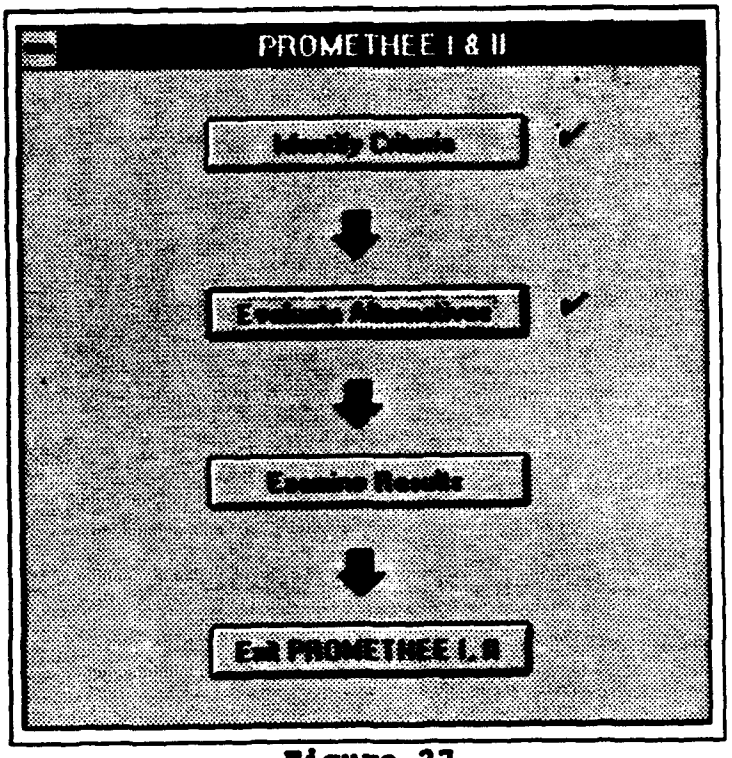


Figure 27
PROMETHEE I,II - Main Screen

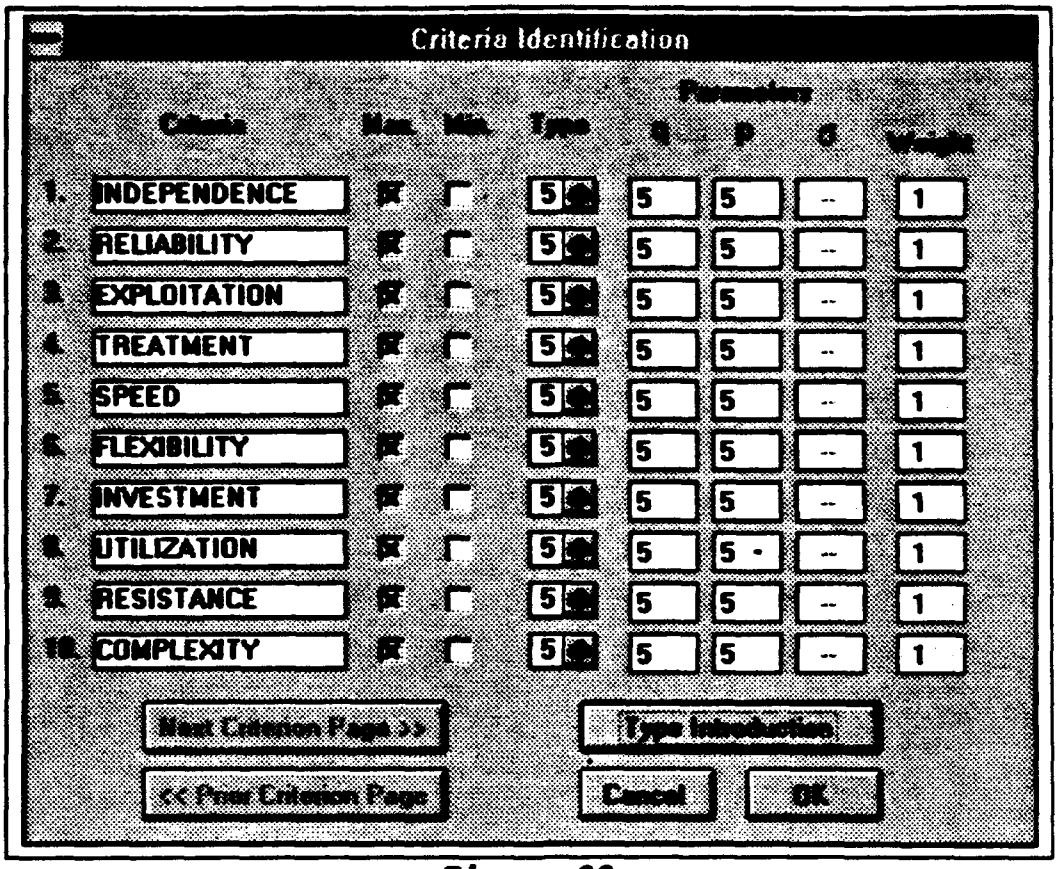


Figure 28
PROMETHEE I,II - Criteria Identification Screen

type of the generalized criterion is changed, the corresponding parameter(s) to be defined are also changed. The text boxes designed for user input are such that only the variables that are required to be defined will be enabled. The default value for the weight of each criterion is 1, but it is up to the user to assign his weight to a criterion.

As can be seen on the screen, the criteria of the selected example are all to be maximized. Every criterion use the generalized criterion type five. Both the parameters p and q of each criterion have the value of five. This example will be processed step by step in the following sections and eventually solved.

This screen can only accommodate ten criteria at a time. A problem with more than ten criteria will so be displayed by more than one screen. The ***Next Criterion Page*** and the ***Prior Criterion Page*** buttons are used to go one page forward or backward at a time. The ***Type Introduction*** button is used to open the generalized criteria introduction screen. The ***OK*** and ***Cancel*** buttons both close this screen and return the user to PROMETHEE I/II main screen, but the user will be ready to go to next step when the ***OK*** button is pressed.

c. *The Generalized Criteria Introduction Screen*

Figure 29 shows the screen the user sees when he clicks on the ***Type Introduction*** button on the PROMETHEE I/II main screen. This screen is used to help the user define each criterion and its parameter(s). The controls used on this screen are picture boxes, labels, and command button. Since these six pictures are all bitmaps, it takes a lot of memory to display this screen. The ***OK*** button will close this screen and return the user to the criteria identification screen.

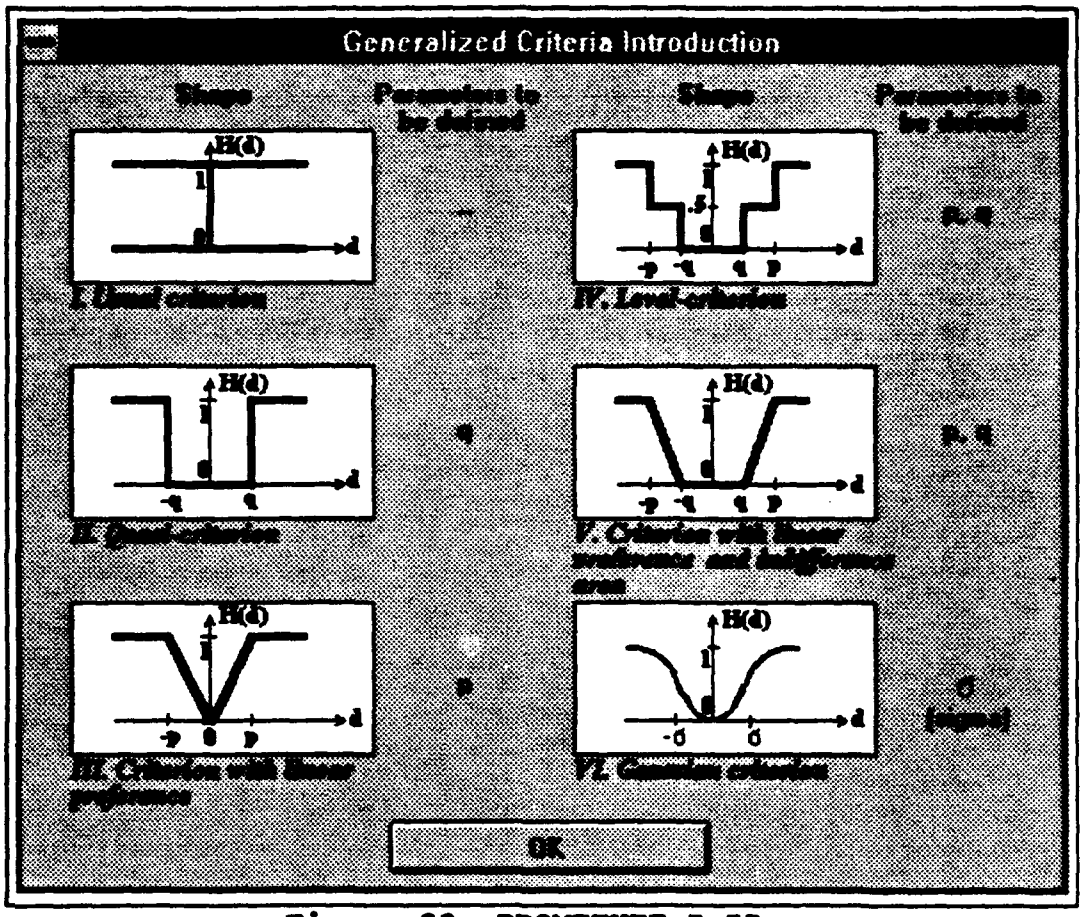


Figure 29. PROMETHEE I,II - Generalized Criteria Introduction Screen

d. The Alternatives Evaluation Screen

Figure 30 shows the screen displayed when the user clicks on the ***Evaluate Alternatives*** button on the PROMETHEE I/II main screen. The design of this screen is exactly the same as the other alternative evaluation screens for previously discussed MCDM techniques. The user can evaluate given alternatives with the help of provided criterion information.

e. The Result Screen

Figure 31 shows the screen the user sees when he clicks on the ***Examine Results*** button on the PROMETHEE I/II main screen. Controls used are frames, labels, list boxes, and command buttons.

The upper frame is to display PROMETHEE I partial preorder, while the lower one is to display PROMETHEE II complete preorder. In the upper frame, the upper pair of lists are used to present the outranking relation of alternatives, while the lower pair to present indifferent relation. To see the partial preorder, the user should use the lists on the upper frame and choose an alternative from one of the left side lists. The content of the right side lists will change according to the changes on the left side lists. As can be seen on the screen, when clicked on the top right list on the screen, the S38CMV outranks RESO, RESO-FUTURE, AS400CMV, UNISYS, and UNI-FR. The user should see the outranking or the indifferent list when he clicks on the corresponding right side list.

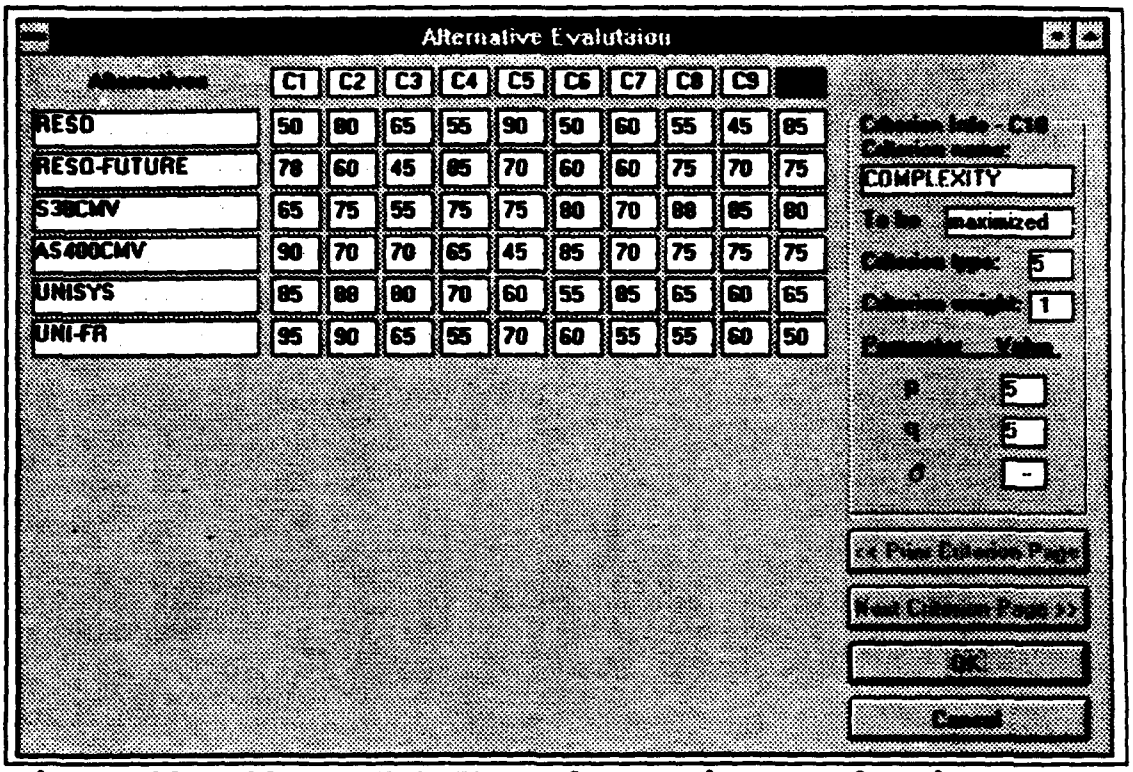


Figure 30. PROMETHEE I,II - Alternatives Evaluation Screen

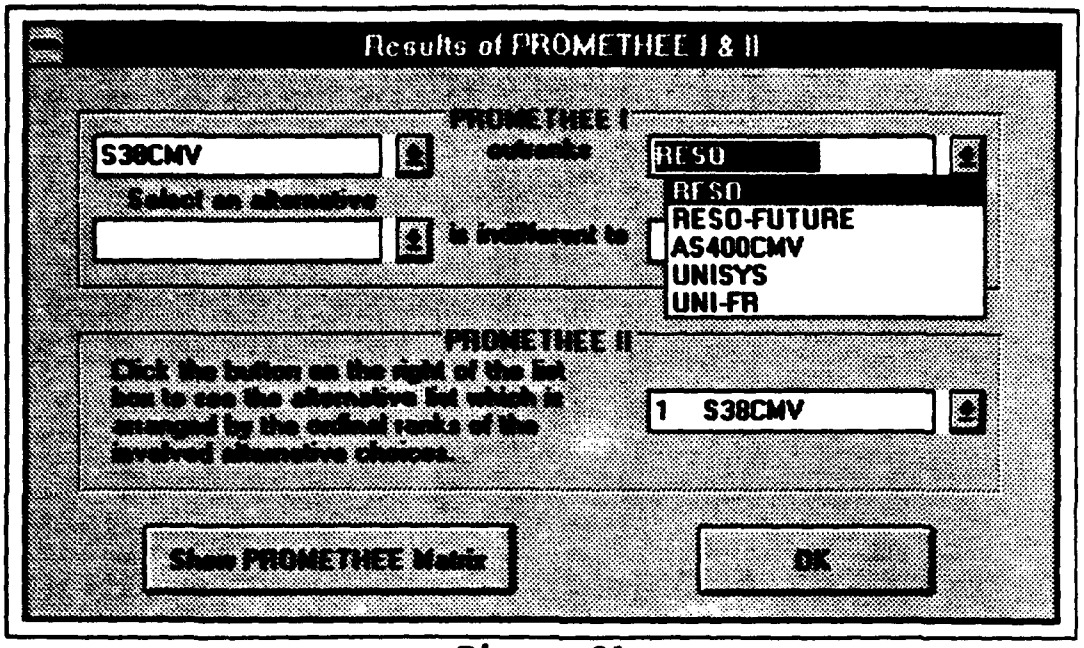


Figure 31
PROMETHEE I,II - Result Screen (Partial Preorder)

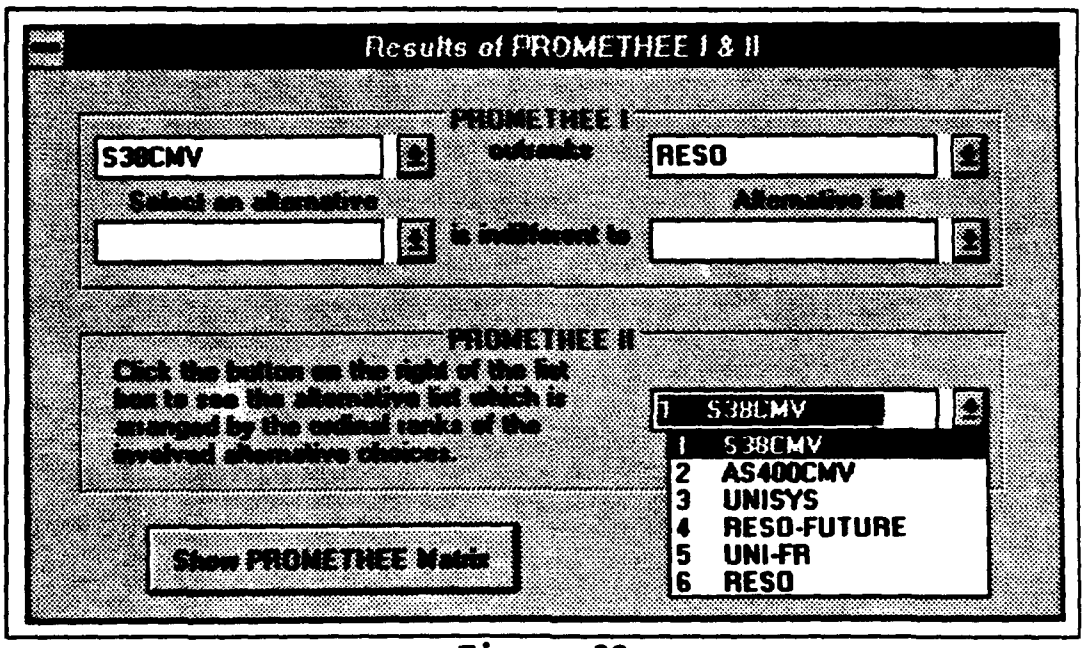


Figure 32
PROMETHEE I,II - Result Screen (Complete Preorder)

To see the complete preorder, the user has to use the list on the lower frame and click the list to see the ranking of alternatives which are arranged in ordinal rankings. Figure 32 shows the screen the user sees when he clicks the outranking list which displays the ordinal ranks of the involved alternatives. For instance, "1 S38CMV" which is the first item on the list means that the ordinal rank of S38CMV is 1.

The ***Show PROMETHEE Matrix*** button opens the PROMETHEE I/II result matrix, while the ***OK*** button closes this screen and returns the user to PROMETHEE I/II main screen.

f. The Result Matrix Screen

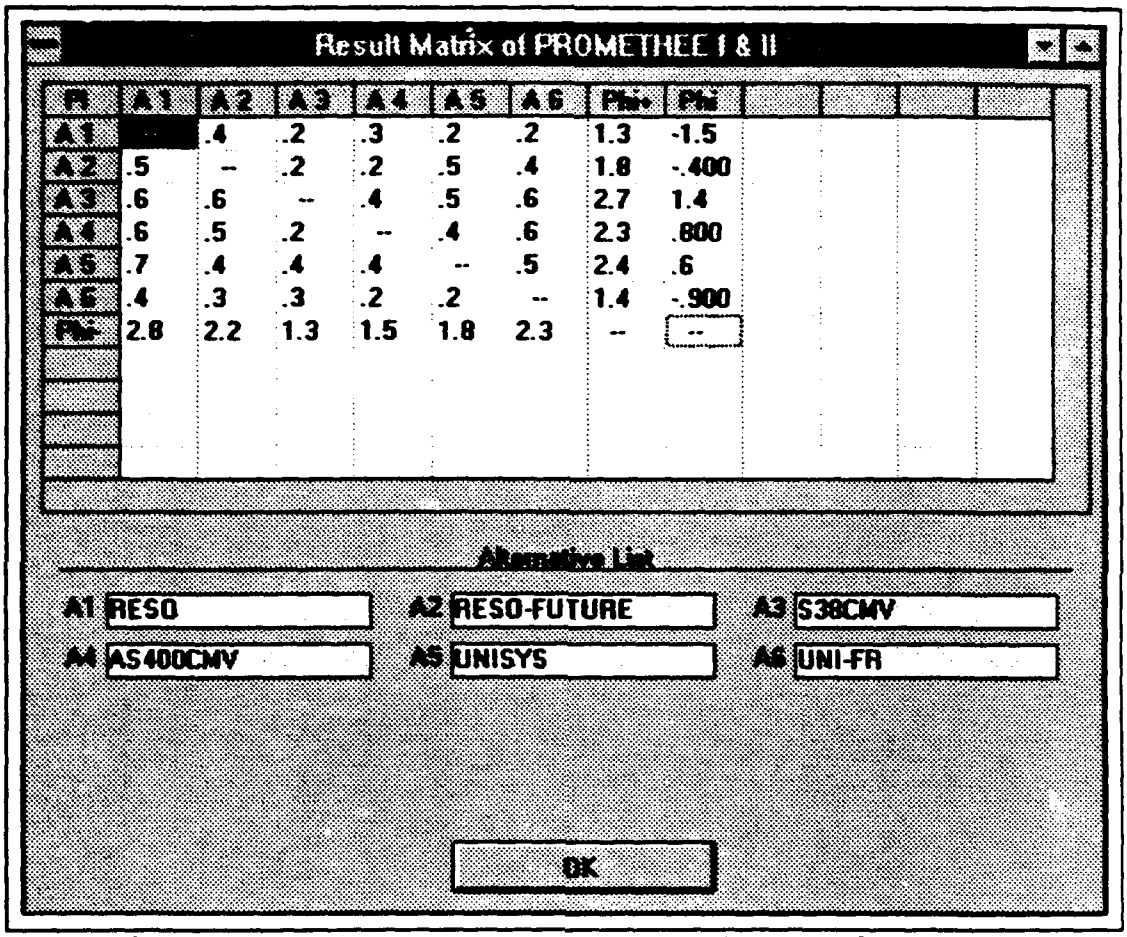
Figure 33 shows the screen display when the user clicks on the ***Show PROMETHEE Matrix*** button on the PROMETHEE I/II result screen. Controls used on this screen are labels, command button, and grid.

The matrix itself is an example of using grid and is used to show the computation results of $\pi(a,b)$, **phi+**, **phi-**, and **phi**. The ***OK*** button will close this screen and return the user to the PROMETHEE I/II main screen.

2. PROMETHEE V

a. The Main Screen

Figure 34 shows the screen the user sees when he chooses the ***PROMETHEE V*** option on the PROMETHEE main screen (refers to Figure 26). Controls used in this screen are labels, text box, and command buttons. The user inputs the number of constraints in the text box provided. This number is used to construct



Pi	A1	A2	A3	A4	A5	A6	Phi+	Phi-
A1	--	.4	.2	.3	.2	.2	1.3	-1.5
A2	.5	--	.2	.2	.5	.4	1.8	-.400
A3	.6	.6	--	.4	.5	.6	2.7	1.4
A4	.6	.5	.2	--	.4	.6	2.3	.800
A5	.7	.4	.4	.4	--	.5	2.4	.6
A6	.4	.3	.3	.2	.2	--	1.4	-.900
Phi-	2.8	2.2	1.3	1.5	1.8	2.3	--	--

Figure 33. PROMETHEE I,II - Result Matrix Screen

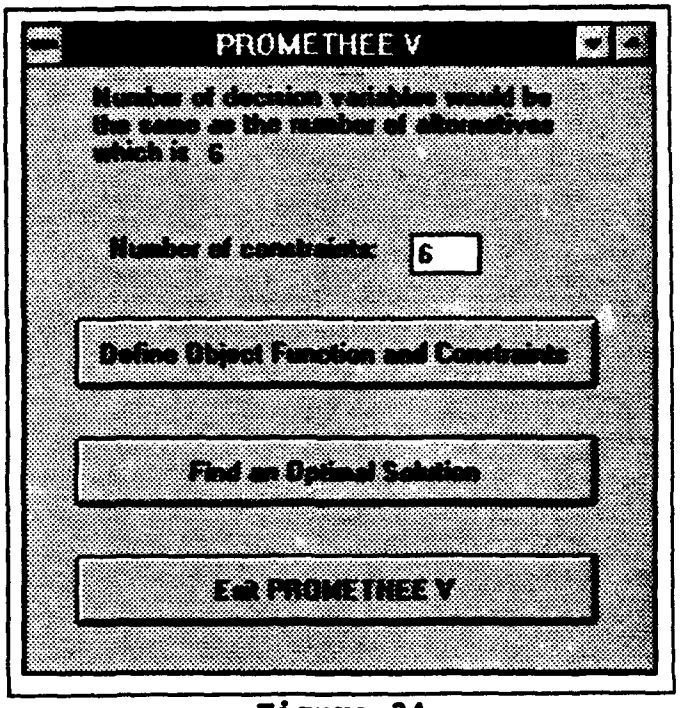


Figure 34
PROMETHEE V - Main Screen

necessary arrays for storing the coefficients of constraints. The ***Exit PROMETHEE V*** button exits this screen and returns to the PROMETHEE main screen. The other two command buttons will be discussed later.

In order to use the selected example to demonstrate PROMETHEE V technique, it is necessary to do some modifications for the example. The revised problem is subject to the followings:

1. CMA decides to buy at most three systems
2. CMA would need at least two systems
3. at least one choice among RESO, RESO-FUTURE, UNISYS, and UNI-FR is desired
4. at most two choices among RESO, RESO-FUTURE, UNISYS, and UNI-FR is desired
5. exactly one choice between S38CMV and AS400CMV is desired
6. the cost is $55,000 for RESO, $48,000 for RESO-FUTURE, $65,000 for S38CMV, $56,000 for AS400CMV, $52,000 for UNISYS, and 50,000 for UNI-FR, it is expected that no more than $160,000 of total purchase

Since there are six alternatives involved, the number of decision variables is six. The number of constrains is six. The objective function is to maximize the overall phi. Referring to Figure 33, the value of phi is -1.5 for RESO, -0.4 for RESO-FUTURE, 1.4 for S38CMV, 0.8 for AS400CMV, 0.6 for UNISYS, and -0.9 for UNI-FR. The user is going to define the objective function and constraints in the following section.

b. The Objective Function and Constraints Definition Screen

Figure 35 shows the screen the user sees when he clicks on the ***Define Objective Function and Constraints*** button on the PROMETHEE V main screen. Controls used on this screen are labels, text boxes, frames, option buttons, and command buttons.

This screen allows the user to input the objective function and constraints to construct the (0 - 1) linear program. The design focusses on providing the user with ease of input of the coefficients for objective function and constraints. Since the space of the screen is limited, only the objective function and one of the constraints will be displayed at any time. Of course, it will be better if the user can see all the constraints on the screen at a glance. However, the user can access any constraint with at most three key-strokes. On the center right of the screen, there is a frame labeled ***Go to***. The user can go to any constraint by typing-in the constraint number and then clicking on the ***Go*** button. The definition of variables will be shown on the lower part of the screen. The ***Show Coefficient Matrix*** button opens the coefficient matrix screen. The other command buttons work the same way as those discussed before.

c. The Coefficient Matrix Screen

Figure 36 shows the screen displayed when the user clicks on the ***Show Coefficient Matrix*** button on the objective function and constraints definition screen. This screen is to display the coefficients of the objective function and constraints input by the user. The controls used on this screen are grid and command button.

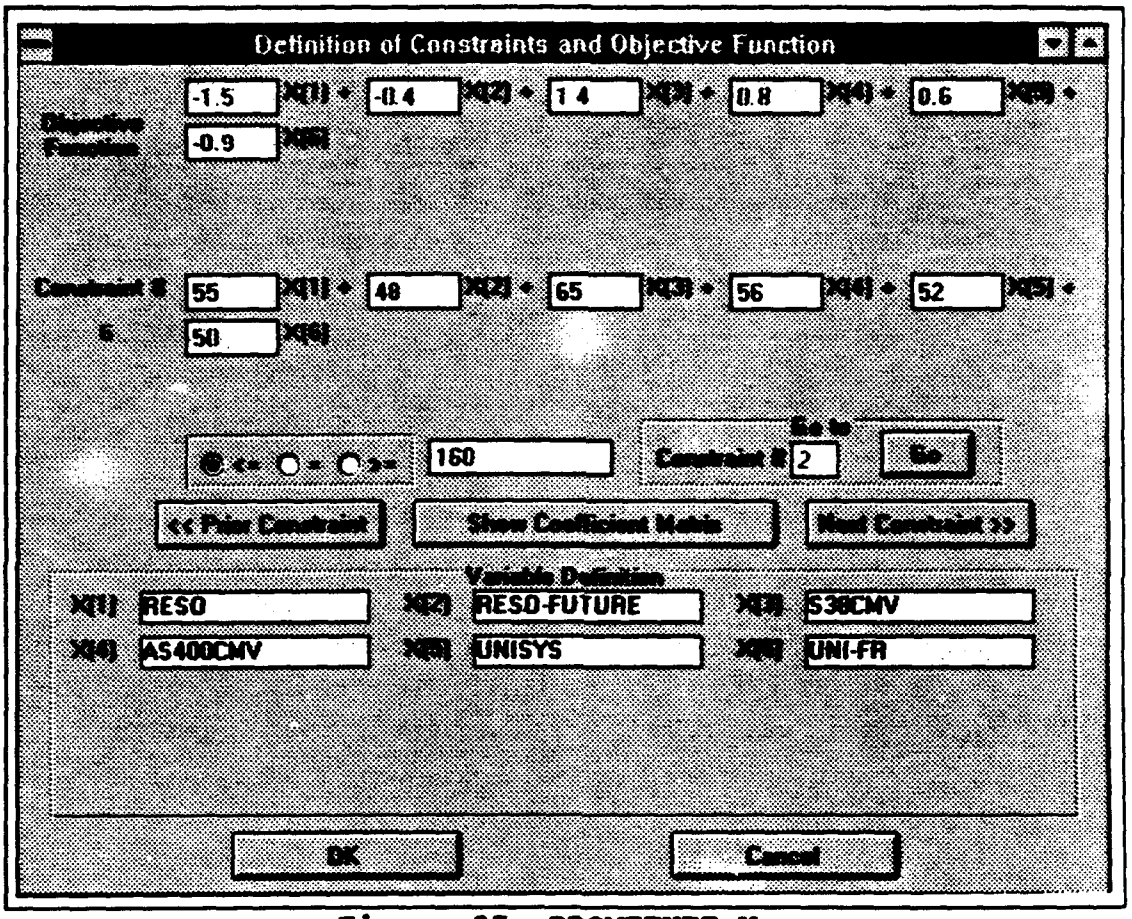


Figure 35. PROMETHEE V -
Objective Function and Constraints Definition Screen

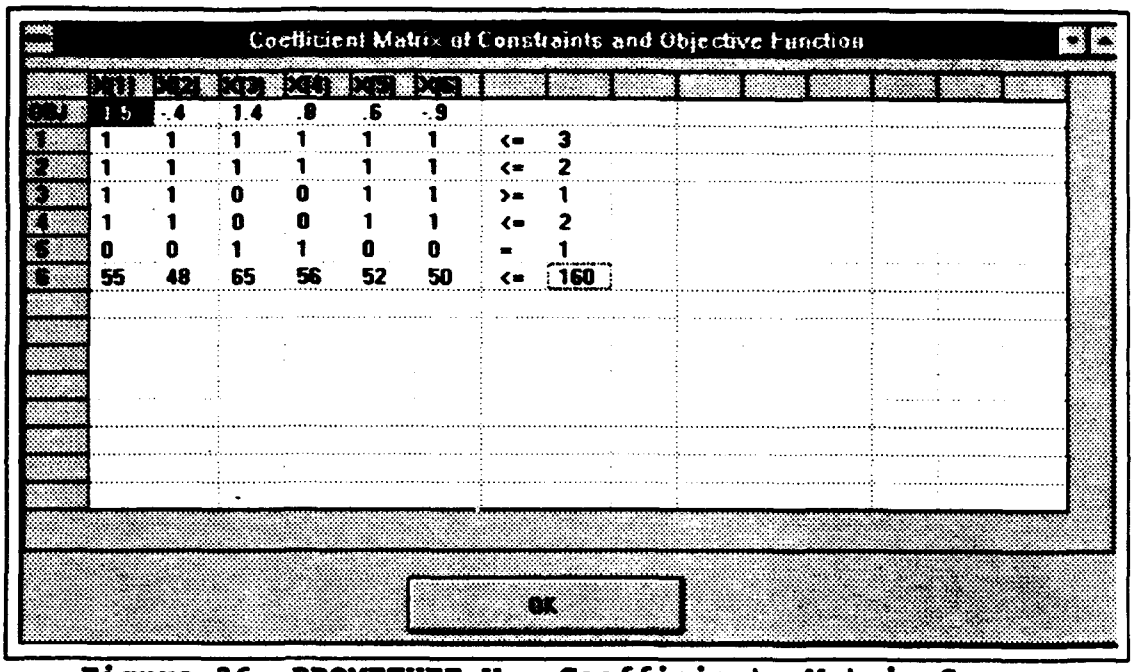


Figure 36. PROMETHEE V - Coefficients Matrix Screen

d. The Result Screen

Figure 37 shows the screen the user sees when he clicks on the ***Find an Optimal Solution*** button on the PROMETHEE V main screen. Since PROMETHEE V is to solve (0 - 1) linear problems, the decision variables with value 1 will be selected as the optimal solution. If an alternative is selected, it is marked by a check mark. The value of the objective function will also be displayed. As can be seen on the screen, the S38CMV and UNISYS are the optimal solution for this example. Controls used on this screen are frame, labels, picture boxes, and command button. The ***OK*** button is used to close this screen and return the user to the PROMETHEE V main screen.

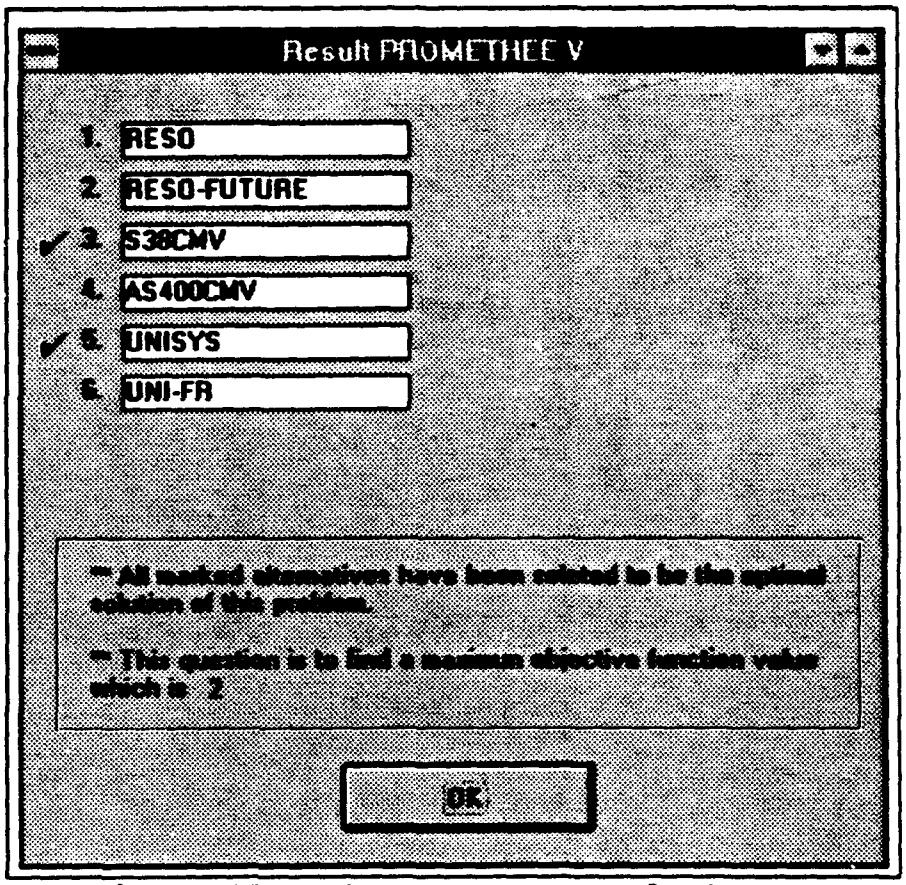


Figure 37. PROMETHEE V - Result Screen

VI. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

A. SUMMARY

The objective of this research is to implement the Multiple Criteria Model Base in the Windows-based Co-op with a graphical user interface generator. In other words, this thesis involves the design and implementation of both the MCDM programs and the graphical user interface.

The implemented MCDM programs and the graphical user interface are to be operated in the Windows environment. When presenting the Multiple Criteria Model Base, the Co-op applications framework and communication parameters must be maintained. With the help of commercial GUI generator and the general principles of graphical user interface design, the implemented GUI presents a complex set of decision support tools in a way that is easy to understand, use, and control. The screen designs are consistent both in presentation and control devices, and will provide a clear conceptual picture of the Co-op system. The main goal in the user interface design is to allow the user control over the application and not the converse.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

At present this Co-op system is still in the integration phase. Several follow on studies are suggested :

- The design of the result screen for PROMETHEE I and PROMETHEE II is not good enough due to the limited capability of the GUI generator. A dynamic graphic presentation of the results is recommended.

- A study could be conducted to measure user preference between the GUI-based and the current character-based programs. Since the task sets of the programs are identical, a valid comparison could be made.

- Additional capabilities could be added to the original program to further enhance the support of Multiple Criteria Decision Making process.

APPENDIX A

A. WHAT IS VISUAL BASIC?

A *graphical user interface* (GUI) is what a user sees when a Windows application is opened. Microsoft Visual Basic is a powerful graphical programming system that enables software developers to create Windows applications (or GUIs) with BASIC code. For an experienced programmer who is unfamiliar with programming in Windows, Visual Basic provides the necessary tools to easily create the graphical elements that are common to Windows applications.

The Visual Basic programming system allows programmer to create *objects*, set and change their properties, and then attach the functional BASIC code to them. The Visual Basic philosophy of programming is first to create objects, such as windows, icons, and menus, and then to write the procedures that invoke each of these objects. This is different from the traditional method of writing a program, in which structures exist for controlling program flow from one procedure to another in a logical manner until the program ends.

The primary programming interface for Visual Basic is the Windows, a visually-oriented, graphical operating environment for DOS. The major advantage of using Windows is that it provides a consistent and manageable interface across different applications. The following section will discuss the controls used in the design of Co-op.

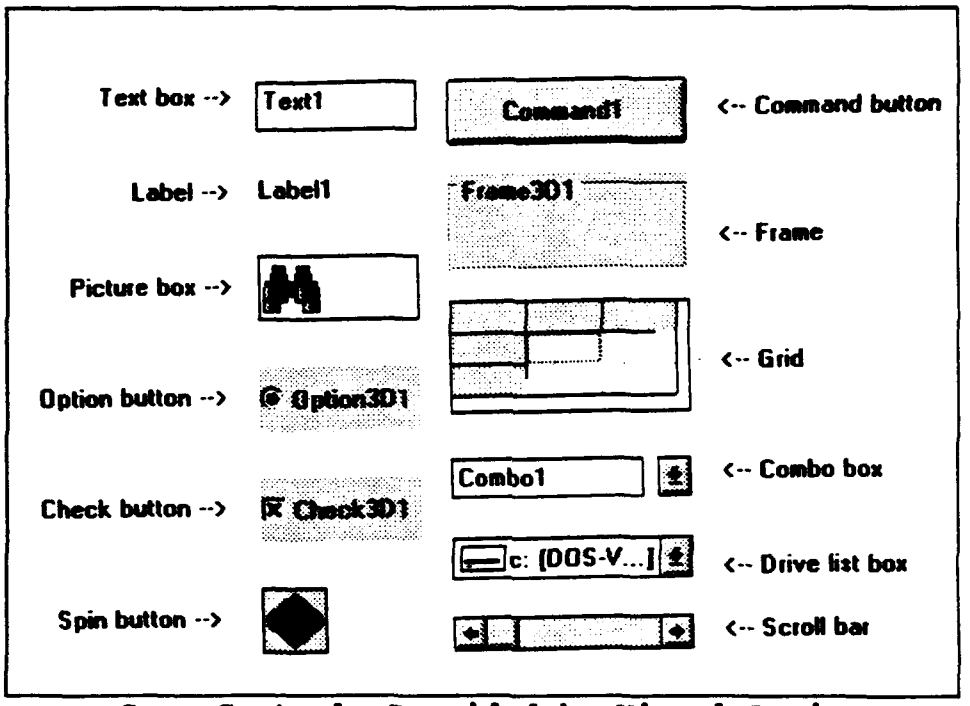
B. CONTROLS USED IN THE DESIGN OF CO-OP

Each window itself is a form in the Visual Basic during the design time. Therefore, the design of every screen in the Co-op system or any application must begin with a form. It can also be stated in such a way that a form is the foundation of any Visual Basic application that will eventually run as a stand-alone program in Windows. A *control* is the name for any object that a programmer draws on a form. A programmer designs a screen by placing controls, such as text boxes and command buttons on a form and then setting the properties of the form and controls. Finally, the programmer writes the code to bring the application to life. A *module* is a structure for writing the code that a programmer attaches to a form and its controls.

Visual Basic uses the metaphor of the "event" to describe its programming paradigm. The programmer will always use this event-driven approach when creating an application. *Event-driven* means that all controls a programmer draws on a form specify how the interface will behave. In other words, Visual Basic's controls wait for particular events to happen before they respond. An *event* is an action that is recognized by a control such as clicking, double-clicking, key-pressing, etc. This section will briefly discuss the controls used in the Co-op system, as shown in Figure.

1. Text box

A text box can accept any text input by the user and will store the input text in ASCII format. It can also display text designated by the programmer at the design time. If a text box is designed to receive numeric input from the user, its code will be written in a way that only digit characters and decimal point (if desired input is floating



Some Controls Provided by Visual Basic

number) are accepted. The input is still stored in ASCII format, while it will be translated to numeric format when it is used for computation.

2. Label

Label can be seen in most of the Windows applications. It is used to display messages or to label another control by the use of its caption property. The caption must be specified by the programmer during design time. The user cannot interact with or modify its caption.

3. Picture Box

A picture box is used to display bitmaps, icons, or Windows metafiles. It can also used to provide an area to display text or to act as a visual container of other controls. The text or picture it displays can not be modified by the user.

4. Frame

A frame is used to provide a visual and functional container for controls. For example, the programmer can construct a multiple choice option through the use of frame, so that only one option can be chosen in such a functional container.

5. Command Button

This is one of the most commonly used controls in Windows applications. In almost every window, there must be at least one command button to perform some sort of function, such as opening another window, closing current window, etc. The purpose of this control is to carry out a command or action when a user clicks on it.

6. Check Box

It is used to display a True/False or Yes/No option. Any number of check boxes on a form can be checked at any one time.

7. Option Button

It is used in an option group that contains two or more option buttons. The option group displays a series of multiple choices from which the user can select only one.

8. Spin Button

It is used to combine with label or text box to allow the user to select choice or to manipulate input. Though a label is not able to receive any input from the user, when combined with spin button they can act like an input receiver. It is usually used when the number of possible input is limited, and when the false input is definitely not expected by the programmer. The input is normally numeric, but it also can be a choice selection.

9. Grid

It is used to display a series of rows and columns and at the same time allows the programmer to manipulate data in its cells during the design time. However, the user would not be able to modify any of the displayed data. If a table is needed to present some data of information, this is a good control to use since the programmer does not have to build the table by drawing a text box or a label at a time.

There are many more controls available in Visual Basic, some of which are *combo box*, *scroll bar* and *drive list box* as shown in Figure. All the controls provided by Visual Basic are easy and convenient to use. They also assist in ensuring that the design of any software is consistent.

LIST OF REFERENCES

1. Brans, J. P., *PROMETHEE: A New Family of Outranking Methods in Multicriteria Analysis*, Elsevier Science Publishers B.V., 1984.

2. Brans, J. P., *"PROMETHEE V: MCDM Problems with Segmentation Constraints."* *INFOR*, Vol 30, Num 2, May 1992.

3. Bui, T.X., *La Methode ELECTRE I, Manuel d'Utilisation'*, University of Fribourg, 1982.

4. Bui, T.X., *Co-op: A Group Decision Support System for Cooperative Multiple Criteria Group Decision-Making*, pp. 7-24, Springer-Verlag, 1987.

5. Price, C. T. Jr., *Multiple Criteria Decision Models for the Co-op Group Decision Support System*, Thesis, Naval Postgraduate School, Monterey, CA, March 1991.

6. Roy, B., "*ELECTRE III: UN ALGORITHM DE RANGEMENT FONDE SUR UNE PRESENTATION FLOUE DES PREFERENCES EN PRESENCE DE CRITERES MULTIPLES*", Cahiers du Centre d'Etudes de Recherche Operationnelle, Paris, 1978.

7. Roy, B., "*ELECTRE IV: CONSTRUCTION DE LA RELATION DE SURCLASSMENT FLOUE PAR L'INTRODUCTION DES RELATIONS DE SURCLASSMENT EMBOITEES*", Cahiers du Centre d'Etudes de Recherche Operationnelle, Paris, 1992.

8. Sabene, R., *Designing a Graphical User Interface for a Bilateral Negotiation Support System*, Thesis, Naval Postgraduate School, Monterey, CA, March 1992.

9. Zeleny, M., *Multiple Criteria Decision Making*, McGraw-Hill, Inc., 1982.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey, CA 93943-5100	2
3.	Professor Tung X. Bui Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Myung Suh Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Lieutenant Hsin-Yen Tsai No. 13, Kuang-Fu Lane, Chun-Lin Village, Li-Kang Ping-Tung County, Taiwan, R.O.C.	2
6.	Naval Academy Library Tso-Ying P.O. Box 90175 Kao-Hsiung, Taiwan, R.O.C.	2