

©2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Developing a Knowledge-Based Multi-Objective Decision Support System

J. Lu¹, M.A. Quaddus¹ and R. Williams²

¹ Graduate School of Business, ² School of Information Systems
Curtin University of Technology, Perth, WA 6845, Australia

E-mail: luj@cbs.curtin.edu.au, quaddusm@gsb.curtin.edu.au, williamsr@cbs.curtin.edu.au

Abstract

Decision Support Systems under Multiple Objectives (MODSS) has been classified as a specific type of system within the broad family of DSS and has been one of the most active areas of research. However, despite its theoretical development large scale real-world applications of multi-objective decision making (MODM) methods had been seriously lacking due to the technical knowledge and expertise needed to select and apply the most appropriate method. This paper explores the possibility of embedding intelligent guidance within MODSS, and outlines a specific guidance framework for the design of knowledge-based guidance for the selection of a suitable MODM method. This framework has been implemented as an intelligent and graphical user interface based multiple objective decision support system prototype. The prototype, with its database, methodology base of MODM and knowledge based system, will help any decision maker in selecting the most appropriate method in solving various MODM problems.

1. Introduction

Many managerial decisions involve the consideration of more than one objective. Despite the research and development of a large number of different kinds of multiple objective decision making (MODM) methods and given that a multiple objective decision support system addresses complex decision problems, there is a lack of success in practical applications of these methods [1]. Why are the developed MODM theories well known, used and generally accepted almost without criticism only in the academic field? Why do the number of managers/decision makers effectively using multi-criteria DSS (MCDSS) remain particularly insignificant [2]? One of the reasons hindering the success of these systems may be due to the large number of different mathematical models and different solution processes involved in many of these MODM methodologies making them difficult for the DMs to determine the most effective methods for any given problem. This deters many corporate DMs, who are not formally trained in the field of MODM, from utilizing many of them [3, 4].

In this paper, we consider the MODM problems as the optimization problems where all the objectives and constraints are structured as pure linear functional forms

of decision variables. It has been found that some methods of MODM are more suitable and efficient than others in the solution of a particular decision problem for particular DMs. Some methods provide a solution analysis procedure but some do not, some provide a set of efficient solutions to DMs so that they can choose a satisfactory solution, while some provide only a unique solution. Hence, the proposed MODSS should preferably contain a sufficient number of MODM methods for the DM's use. DMs can select the most suitable method for solving their decision problem and have the appropriate interaction for the solution analysis. However, such knowledge for selecting methods and analyzing solutions is not readily available. Therefore, to utilize the potential of a methodology base effectively, MODSS must have the capability of guiding the user to select and use different MODM methods from the methodology base. A knowledge-based intelligent front-end is thus necessary to achieve better guidance for the DMs during the decision process [5].

This paper investigates the design and development of knowledge-based guidance on MODSS and presents a specific guidance framework for a knowledge-based MODSS (KB-MODSS). The objectives of the paper are twofold: (i) to present the detailed **process** of developing the KB-MODSS, which will help other researchers to develop optimization based knowledge based systems, and (ii) to present the **features** of the interactive KB-MODSS as implemented with DELPHI and CLIPS. The paper is organized as follows. Section 2 presents the functions architecture of KB-MODSS. Section 3 provides the framework for a knowledge based guidance system. The development of a KB-MODSS prototype is described in section 4. Finally, conclusions are presented in section 5.

2. Functions Architecture of Knowledge-based MODSS Applications

Integrating the knowledge components in the DSS conceptual framework will considerably increase the expertise embedded in the DSS and will improve the capacity for users to enhance this expertise [6, 7, 8]. Some researchers [1, 9] have emphasized the need for research in the intelligent MCDSS and explored the possibility of embedding decision guidance within MCDSS. Klein & Methlie [10] provided a conceptual

framework for a knowledge-based DSS (KB-DSS). They defined the basic functions of KB-DSS applications. When knowledge has been formalized under the form of rules, for example, to constitute a knowledge base, it is possible to request the system to reach a conclusion based on what can be logically derived from the knowledge stored under the form of rules and observed facts. Decision situations under multiple objectives are inherently unstructured. They call for implicit or explicit trade-offs among the criteria and require extensive human intervention to arrive at the 'best compromise' and satisfactory solution. A MODSS needs more in-depth knowledge of MODM methods, supporting the recognition, structuring of MODM problems and the selection of MODM methods. Once knowledge is embedded into an MODSS, the functions of the KB-MODSS application will be enhanced to provide 'intelligent assistance' to the DM with the following functions during its problem-solving task. The functions architecture for knowledge-based support in MODSS is shown in Figure 1.

(1) Supporting the man-machine interaction

Most MODM methods involve a man-machine interactive process. There are three main types of man-machine interaction. The first type of interaction is performed before the solution process even starts. In this type, explicit preference functions of the users are needed. In others the preference information of the users are needed during the solution process. In this type, the users are required to provide on-line preference information, but no explicit preference function is needed. This type of approach is widely known as the interactive approach. The third type of approach requires preference information after a set of candidate solutions has been generated. In this, the users are simply required to choose

the most satisfactory solution from the set [11]. Whatever the type of interaction, a knowledge based guidance plays an important role.

A KB-MODSS should provide an intelligent interface to support the man-machine interaction during the problem-solving process. The interface of an MODSS defines how the systems resources are used and gives a way for DMs to interact with their applications. The interface of an MODSS application is a key element for its success because this is the user's view of the system and the unique way to input objective functions and constraints, to analyze solutions and to receive a satisfactory solution. It is very important in an MODSS because the decision situation under multiple objectives is inherently unstructured. This object oriented graphical interface is usually made of a hierarchy of menus and icons giving access to resources that enable the DM to select various functions in the decision problem solving process.

Choosing and displaying information about decision variables, objective functions, constraint functions, goals, ideal solutions and satisfactory solutions needed for MODM problem solving, are the most usual functions provided by the interface of an MODSS. However, for more complex interactive actions, e.g., to make a solution analysis by relaxation in the STEM method (algorithm) [12], knowledge-based guidance will be needed to support the assistance function to help the DM. The more important roles of the interface are to help users select between alternatives and display/store a report. The application interface is a layer between the DM and the resources of the application. Usually, an MODSS application requests a wide range of techniques for the user interaction. Selection of options in a menu or icons with guidance is fine for calling a resource such as a data file, a database table and an MODM methodology.

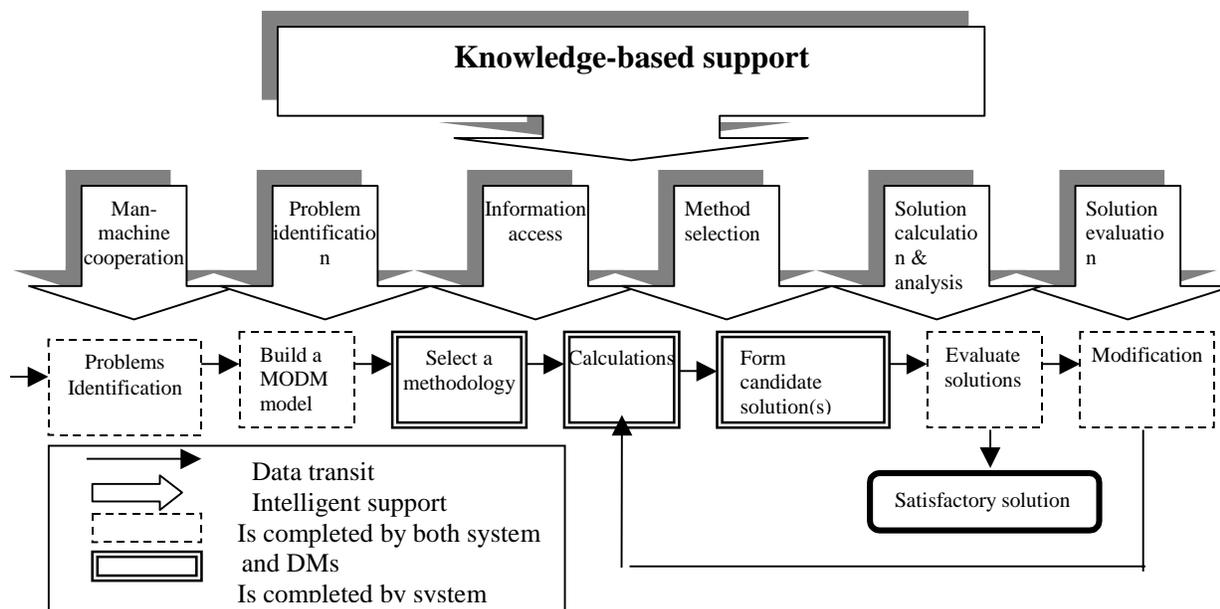


Figure 1. Function Architectural Framework of KB-MODSS

(2) Supporting identification of the application problem

In many real situations, the first step is not even to assess the objectives of an MODM problem but to understand what an objective is, i.e. problem identification and recognition. For the support of the problem identification phase, the DMs should be guided by the system to determine which questions (decision problems) are to be answered, define the hypotheses to be tested and the effects to be estimated.

The first aspect for supporting problem identification is to show data and appropriate messages. Since the essence of problem recognition lies in the ability to detect differences between the present state and expected states, for MODSS, these differences are reflected in differences between goals and efficient objective values. The capacity to present information in windows with color graphics is a key element of a good display of information for identification of application problems. An example of the use of an expert system (ES) in order to build an alternative in multiple attribute DSS (MADSS) is presented in [13] where a frame-based system determines the possible pathology of a patient.

The alerting function of an MODSS is also a very important function with which to support problem recognition. The idea is that the system should be able to set conditions (predicates and rules) based on the DMs application which, if they become true, will trigger the display of a report, or send a message, or, eventually, trigger some other action such as the computation of a model. This function is very useful when the DM is confronted with a large amount of data that is evolving very fast with time. An 'intelligent' alerting system must be defined. The knowledge-based rectification guidance feature is one of the ways of alerting the user.

(3) Supporting access to information during decision problem solving

Easy access to the decision information and the system resources are the first support a DM requests from an MODSS. The resources of the MODSS include the database, the users' decision models, the MODM methodology base, available displays and reports produced. A KB-MODSS should provide information about the resources of the system in terms of data useful for solving the MODM problem. This information should include the meaning of models and methods that are used in studying the problem; the meaning of variables used in MODM models, the conditions of use, and the underlying hypothesis of MODM models; the use of available reports; the ways to select a satisfactory solution, solutions analysis strategies, menu options and help. A classification for accessing information in MODSS is shown in Table 1. A knowledge base can be defined to help the DM using the application to select the right resources in terms of decision models, reports, databases and method bases. The FINSIM KB-DSS [10] application uses several knowledge bases for financial diagnosis, provides more than ten different kinds of reports, various recalculation options in the financial model and can access three large commercial financial databases.

(4) Supporting the MODM methodology Selection

It is important that an MODSS contains a sufficient number of methodologies in its methodology base for the DMs' use during the solution process. However, it is difficult to select the most suitable method from such a methodology base because of the dearth of expertise and experience needed to understand the specific features of the available MODM methods, as well as the ability to match MODM models with current decision needs.

It has been frequently reported that DMs are often troubled by the choices of suitable methodologies according to their various applications. Usually only experts in the field are able to take full advantage of the MODSS. This is because sophisticated analytical skills on the part of the users are required to identify the problems and to match each problem with appropriate

Table 1. A classification of information access

	Information Source	Acquisition methods	Type	Information Example	
I	Implicit	System information	System message & variables	Text	Error message box
		Guidance information	Instructions	Text & variables	The number of solutions & the caption of forms
		Statement information	Implicit trade offs	Variables	A cycle No in a solution process
II	Explicit	Interactive	Interact with applications	Model	Make a relaxation of a satisfactory objective
		Knowledge support	On-line reports	Data	Questions for select a suitable method
		Records	Users database & model base	Data	All of the above

MODM methods. At the same time, average managers who desire to apply the MODSS to their daily operations encounter frustration due to their lack of in-depth knowledge of the MODM algorithms. In order to avoid misuse and even non-use of the computerized MODSS, this system must be capable of guiding the user possessing various levels of expertise in the use of these methods. Thus, a knowledge base can be defined as an intelligent front-end to help the DMs in selecting the suitable MODM method for each decision problem, and should be utilized to provide guidance on the selection of the right strategies for analyzing their data [14]. Once one method is selected, the system will display input data forms to capture the data needed to run this decision model, and then evaluate the decision models to obtain decision alternatives.

(5) Supporting the Solutions Analysis procedure

The knowledge embedded can also be used either to direct the exploration process in the solution set or to support the finding of more efficient solutions in an MODM method [15]. Once an MODM method has been selected and an MODM model has been defined, it is important to be able to use them easily for finding and analyzing solutions. When studying a set of efficient solutions, it is very useful to be able to distinguish in the model the following: decision variables, goals, objectives and constraint functions. The usual intelligent assistance that a MODSS will provide to support solution finding is the computation of the variables and changing of the objectives of the model as the algorithm progresses.

(6) Supporting solution evaluation

In most of MODM the output is not a choice, it is an ordering of efficient solutions. An MODSS should support an evaluation of efficient solutions according to the DMs' desire for objectives, goals, weights and priorities. In the domain of MODM, a certain number of classical and fuzzy criteria can be used for evaluating alternative solutions. To decide among several alternatives the DM usually considers relevant criteria. These criteria may be numerous and are usually conflicting. Most of the time in real decision making, the selection of one solution is often done intuitively without considering relevant criteria.

Usually, the process of building the users' solution evaluation models comprises two tasks: determining satisfactory objectives to include in the model and finding a vector of decision variables. Pomerol [15] showed the use of an ES to build a symbolic utility function in MCDM. A similar structure, which consists of an ES processing various data to access the value of different objectives, can be applied in MODSS. The main criticism of this technique is that it merges choice and descriptive items. To overcome this difficulty and to separate the choice (preference) from the descriptive (factual) part of the objectives, we can follow [16] to imagine a structure

in which each objective is characterized by a fact (base) and each constraint by a rule base. The rule base is organized as a semantic tree of constraints. The evaluation of an objective is made by applying the rules to describe the facts characterizing each objective. It is also possible to use other techniques, such as non-classical logic and neural networks, to evaluate solutions.

3. Intelligent Guidance System and Knowledge Base in MODSS

3.1 Knowledge and knowledge-based information embedded in MODSS

Knowledge and knowledge-based information can be stored by using different formalisms and types in MODSS. The knowledge and information that is embedded in the MODSS application typically includes data, text, variable names, structure of decision models, structure of methods, structure of reports and rules as shown in Table 2.

Knowledge representation consists of formal knowledge represented in symbolic form, that is, symbolic expressions that can be interpreted. Knowledge about states and knowledge about actions is of two kinds: declarative and procedural. In order to manipulate an MODSS it is necessary to have knowledge of both kinds. Declarative knowledge is a description of facts. It is information about real-world objects and their properties, and the relationships among objects. Procedural knowledge encompasses problem-solving strategies, arithmetical and inferential knowledge. Procedural knowledge manipulates the declarative knowledge to arrive at new declarative knowledge.

3.2 Types of knowledge-based guidance in MODSS

Knowledge bases can be defined to support different decision processes, such as to support decision analysis and to support the method selection, by different types of knowledge bases. Knowledge-based guidance can be located wherever it is needed and can be described in different ways. Based on [1] there are six types of guidance in MODSS. The first type of guidance is forward guidance. When DMs are making normal MODM progress, a guidance system can summarize the current step, describe how the current step relates to the previous and next steps, and display a status window identifying the current step and the next step. Forward guidance is presented in the form of instructions that clarify the objective of the step to be performed and explain how the step in the decision question fits into the overall MODM process. The type of guidance is particularly used in the execution of methods. The second type is backward guidance. The DMs can be led backward to complete an unfinished or partially finished previous step by backward guidance. This may be

necessary if the users vary widely in their evaluations of each efficient solution on the constraints during the decision analysis process. The preventive guidance aims to prevent disruption break points, which are misunderstandings of the information and recommendations provided by the MODM methods. The rectification guidance aims to lead users back to a right way once a wrong way (e.g., an impossible relaxation or an unreasonable goal value) has been taken. Compulsion-receive guidance is presented in the form of commands for an operation to be performed, in which the user selects a unique enabled button. The last type is choice-receive guidance. The user can be given a choice of solutions. For example, for a specific decision problem and associated goals for solution, there may be more efficient solutions for the problem. The user can select one of them under the guidance mechanism to complete the decision task.

3.3 A knowledge based intelligent guidance system in MODSS

The major advantage of integrating a sufficient number of MODM methodologies in MODSS is to allow the DMs to use the most appropriate methods for each particular decision problem. It also enables the resolution of complex problems that could not otherwise be solved with a single MODM, and to allow DMs to get solutions from different methods.

Based on the framework shown in Figure 1, a knowledge based intelligent guidance system is designed to guide the users systematically towards the selection of the most appropriate method to use in MODSS. A good way to assist the user to select the right options is to define the knowledge about these methods under the forms of facts and rules.

By requesting assistance, the application will run the inference engine that will automatically ask questions to decide which method should be selected. The guidance system works by depending only on inputs from the DMs

and to determine the best MODM method that best matches the needs of the specific problem. Inputs to the guidance system, elicited from the DMs, include the problem nature, data or information availability, and the desired nature of the solution and the DMs' situation. With the architecture of the intelligent front-end, KB-MODSS can achieve better integration of the different available models and allow selective and flexible use of many popular methods [17, 18]. Figure 2 shows a framework of a knowledge based intelligent subsystem for the selection of a suitable method.

The question subsystem first questions DMs by an elicitation technique. The responses are received by the response subsystem. The responses to each question are asserted in the working memory by the inference engine, and responses to the weight of each question are sent to the 'ignore characteristic strategy' (ICS) subsystem. If a suitable method is found the name of method will be displayed to the DMs at the end of the inference process by the method-show subsystem, else a fuzzy matched method with the lowest un-matched degree is provided through related facts as asserted by the ICS subsystem.

3.4 The modes of intelligent guidance

To provide the appropriate guidance for users possessing different levels of expertise, two modes of guidance have been incorporated, namely the intermediate and novice modes.

The intermediate mode is designed for DMs who are familiar with some concepts and methods of MODM, or not so familiar with the methods but do have access to the various inputs. It will attempt to find methods corresponding to a set of inputs. The DMs can discover which are the methods that correspond to a set of inputs by responding to some technical questions based on their problems, their desired solution and their data preparation.

Table 2. Knowledge representation

	Representation Form	Kind	Use
1	Information as data	Declarative knowledge	Describing the objectives, constraints and decision variables used in the problem
2	Information under the form of text	Declarative knowledge	Describing the meaning of concepts used in the problem formulation
3	Information under the form of variable names	Declarative knowledge	In decision models and decision methods
4	Structure of MODM methodologies	Procedural knowledge	To provide a list of MODM methods to be used in solving relevant decision problems
5	Structure of decision models	Procedural knowledge	To provide a list of relevant variables to be used in solving a class of problems and the causal relationship between these variables
6	Structure of reports	Procedural knowledge	To display the information to maximize decision support
7	Rules	Procedural knowledge	Be composed of a set of conditions and several conclusions of MODS in knowledge base

For DMs who are totally unfamiliar with MODM, the novice mode will prompt the user with a set of general non-technical questions regarding the decision problem, expected solutions and the users' preferences.

3.5 The analysis of the characteristics of MODM methods

The acquisition of knowledge is needed to create a knowledge base. This refers to the process of capturing into the system the expert's knowledge about the MODM domain. It includes the collection of data representing the expertise to be stored in the knowledge base. The knowledge acquired is often structured in the forms of facts and rules. To build the knowledge base, we structured the knowledge necessary for the selection of the MODM methods by classifying the MODM methods according to their characteristics. The characteristics of a MODM method can be divided into four classes:

DM-related characteristics are related to the DMs' preference for selecting a method to solve a problem. Some of these characteristics include the DMs' desire to interact with the system, the DMs' knowledge of a specific MODM method, and so on.

Method-related characteristics are related to the solution process of MODM methods. Some of these characteristics include whether to use a linear

programming technique or goal programming, to define or not to define an ideal solution, and so on.

Problem-related characteristics are dependent on the actual decision problem. For example, some MODM methods such as IMOLP[20] and LGP[22] require provision of weights for each objective of the problem. ISGP [21] and LGP need to provide goals for each objective.

Solution-related characteristics are related to the types of solution processed by the MODM methods. For example, some MODM methods such as ESGP [19], ISGP, LGP produce only a subset of efficient solutions, while others such as STEUER [23] produce all efficient solutions in the neighborhood.

Based on the above classification for characteristics of MODM methods, a total of 14 characteristics are identified for the intermediate mode and 10 characteristics are identified for the novice mode (Table 3). All the MODM methods included in our systems were thoroughly studied and classified according to one or more of the 10 or 14 characteristics. Figure 3 shows the logical connectivity between the 7 methods and the 10 characteristics for Novice mode. As an example, as shown in Figure 3, the ISGP method is characterized by the characteristics of the 'interaction', 'subset', 'D-selection', 'ideal' and 'goal'.

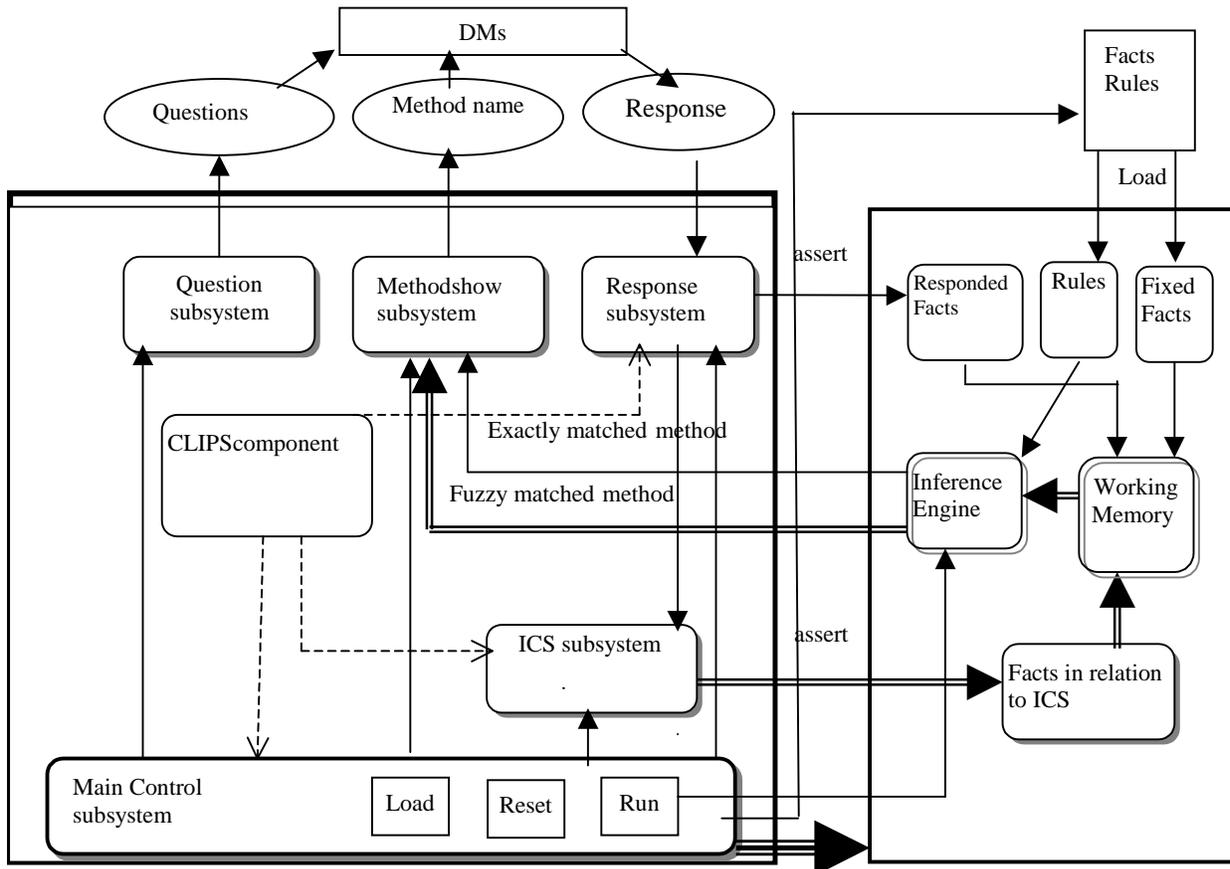


Figure 2. Intelligent subsystem and its working principle

Table 3 Characteristics and Facts for Novice mode

No	Name	Definition	Facts
1	Interaction	more interaction with the system	Char1
2	Subset	system provides a set of solutions	Char2
3	Unique	system provides unique solution	Char3
4	S-Selection	select one satisfactory solution by system	Char4
5	D-Selection	select one satisfactory solution by yourself	Char5
6	Analyze	analyze solutions (e.g. improving/ sacrificing the value of objectives)	Char6
7	Ideal	system defines an ideal solution	Char7
8	Weight	prepare weight for every objective	Char8
9	Goal	prepare goal for every objective	Char9
10	Priority	prepare priorities for every objective	Char10

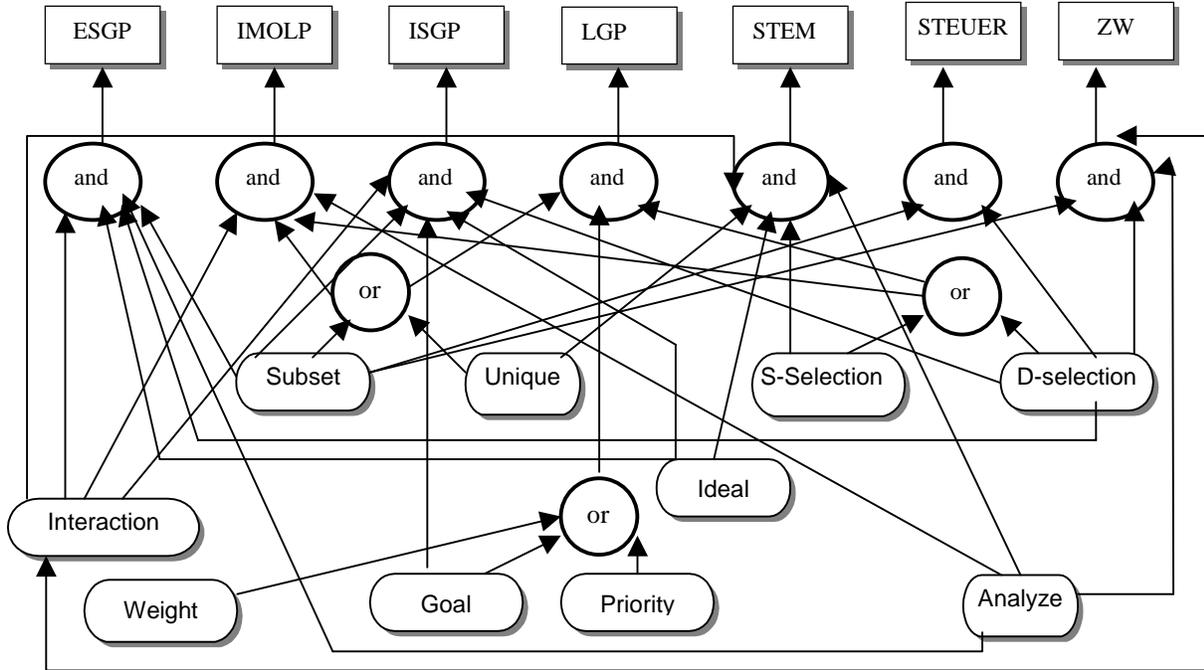


Figure 3. Logical connectivity between MODM methods and their characteristics

3.6 Facts and rules

We applied an expert system tool called CLIPS. CLIPS is a complete environment for developing expert systems and it provides the basic elements of an expert system such as fact-list, knowledge-base which contains all the rules, the rule-base, and inference engine that controls overall execution of rules. There are three ways to represent knowledge in CLIPS. One is by rules, which are primarily intended for heuristic knowledge based on experience. The second one is by defining functions, which are primarily intended for procedural knowledge. Object-oriented programming, also primarily intended for procedural knowledge, allows rules that may pattern match on objects and facts.

The knowledge base for the selection of MODM includes several groups of facts that have different functions. The basic knowledge about each MODM method and its various characteristics are described by a group of facts. Another group of facts is to relate the

response of each question to the facts to be asserted by the inference engine into the working memory. We also need to get a number of facts to relate each characteristic to its corresponding question. The next group of facts relates to follow-up questions to follow given responses. It is necessary to get a set of facts to relate facts that are grouped under the same class. The last set of facts is used to initialize the inference process. An example of a piece of knowledge (facts) representation for MODM method in the system follows:

```
(defacts M3
  (Method
    (Number 3)
    (Name ISGP)
    (Char1 interaction)
    (Char2 subset)
    (Char5 D-selection)
    (Char7 ideal)
    (Char9 goal))
)
```

We used production rules in this prototype. A set of rules, including get-question, characteristic-question, question-question, question-action, found-method and other rules, are defined. A question-action rule is shown in the following code:

```
(defrule quest_action
  (declare (salience 20))
  ?v1 <- (Q_A_Action (Question ?num1) (Answer
?ans1) (Facts $?facts))
  ?v2 <- (Response (Question ?num1) (Answer
?ans1))
  (test (neq ?facts no))
=>
  (retract ?v1 ?v2)
  (assert (Data (Question ?num1) (Facts ?facts)))
)
```

4. Developing the Knowledge-based MODSS Prototype

4.1 Design of IMODSS

We present a design of an integrated MODSS which allows selective and flexible use of many popular MODM methods under intelligent guidance. The prototype system is GUI based, and is called IMODSS. It shows how a multiple objectives decision problem can be identified, analyzed and solved by the prototype IMODSS. The model and solution can then make a display, save to database, and show a graph or report.

IMODSS has a database, a methodology base of 7 well-established MODM methods, a model base and a knowledge base using a state of the art GUI environment. To provide appropriate guidance for the users' decisions, a knowledge base system is utilized as an intelligent front-end. This knowledge base system provides guidance on the selection of suitable MODM methods according to

different problem situations and DMs' situations. The overall architecture for IMODSS is shown in Figure 4. It contains 7 major sub-systems namely: the interface sub-system, input sub-system, intelligent sub-system, method sub-system, data/result management sub-system, model management sub-system and report sub-system.

4.2 Interface of IMODSS

The interface sub-system serves to integrate various other sub-systems as well as to interact with DMs. It consists of a system desktop with a pull-down menu bar at the top. There are eight sub-menus that form the functions in this system. The File sub-menu includes New, Open file, Open model, Print data file, Print model, and Exit. The Input sub-menu includes decision variables input, objectives input and constraint input. The Intelligent sub-menu includes the Novice Intelligent Guide and the Intermediate Intelligent Guide. The Model base sub-menu includes Current User Model and Model Base. The Result Database sub-menu includes Current Data-Results and Result Database. The Rule-base sub-menu and Report sub-menu are contained in this desktop. Figure 5 shows the system desktop.



Figure 5. IMODSS desktop

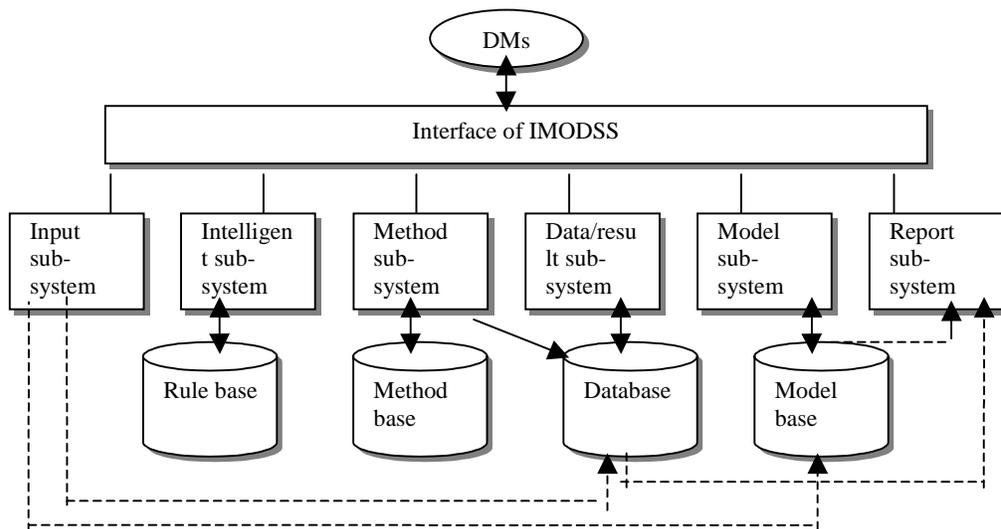


Figure 4. System architecture of IMODSS

4.3 Method base sub-system

The method base sub-system is used to run a method base that has 7 well-established MODM methods available to the DMs. These methods are: Efficient Solution via Goal Programming (ESGP) [19], Interactive Multiple Objective Linear Program (IMOLP) [20], Interactive Sequential Goal Programming (ISGP) [21], Linear Goal Programming (LGP) [22], Step Method (STEM) [12], Steuer [23] and Zionts and Wallenius (ZW) [24]. These methods are developed as independent executables, to facilitate the flexibility required of the system. These methods share similar data acquisition routines. These routines were developed as independent modules so that data acquired could be accessed by all the methods. Figures 6(a) and 6(b) show two fragments of the solution process of methods STEM and ISGP for an application.

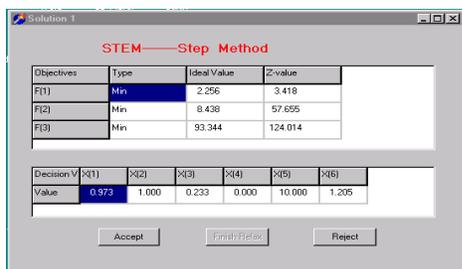


Figure 6(a). An efficient solution has been produced by the STEM method with a ideal value under an intelligent guidance; you can accept or reject it

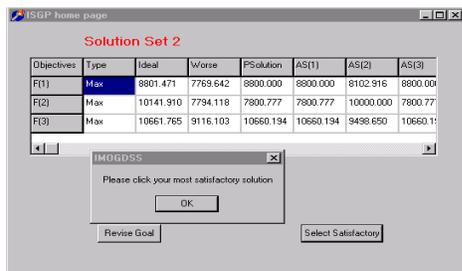


Figure 6(b). A set of solutions with ideal, worst and goal values are shown by method ISGP; you can choose one satisfactory solution

4.4 Intelligent subsystem

IMODSS used the inference engine provided by the expert system shell CLIPS. We applied a package that contains the files for encapsulating CLIPS in Delphi 3. When IMODSS starts up, the DMs select a data file or enters data for objectives and constraints to be used for their decision problem. The DMs then go to the intelligent menu and choose novice mode or others as appropriate. The DMs will go through a series of dialog boxes with the IMODSS that raises questions such as the type of solution

expected, choice of solution analysis, and direct entry of goals for objectives. Each characteristic is given an intensity of importance by the DMs. A weight vector of characteristics is therefore built. A lowest weight is then obtained by ranking this weight vector. The characteristic that corresponds to the lowest weight is considered to be ignored first if no method completely matches the DM's requirement. An example of a dialog box that a DM will see under the novice guidance mode is shown in Figure 7.

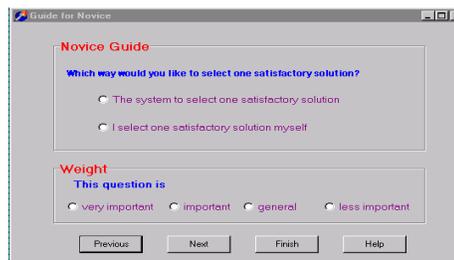


Figure 7. An example dialog box in Novice mode

IMODSS will give a recommendation of one suitable (completed matched or closest fuzzy matched) method for DMs based on their responses. For example, the ESGP method is arrived at if the DMs wish to interact with the system, look for a subset of solutions, select solution by themselves, make a solution analysis, and is able to indicate directly the ideal of the objectives (Figure 8). If the DMs accept the recommended method, the specific module for the method will be invoked automatically.

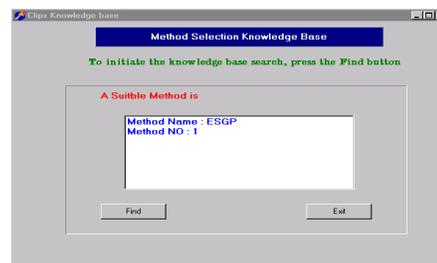


Figure 8. A recommendation

5. Conclusions

In this paper we have described the frameworks of KB-MODSS and its prototype IMODSS. IMODSS supports decision making of linear optimization problems under multiple objectives. We highlighted how knowledge could be used to assist users in areas of scarce expertise and facilitate the operation of a flexible modular decision making system.

This prototype has been tested with a range of practical problems in both experimental and field environment. Our ongoing research activities involved to extend the system to tackle group problem solving.

Acknowledgements: The methodology base developed in this IMODSS is based on the algorithm of MOLP-PC developed by Poh, Quaddus and Chin [25].

References

- [1] M. Limayem and A. Chelbi, "Improving Multicriteria Group Decision Making with Automated Decision Guidance," presented at *The 1997 IEEE international conference on systems, man, and cybernetics*, Orlando, FL, USA, 1997.
- [2] A. A. Angehrn and T. Jelassi, "Supporting Multicriteria Decision Making: New perspectives and New Systems," presented at *ISDS Conference*, 1990.
- [3] H. B. Eom, "The Current State of Multiple Criteria Decision Support Systems," *Human Systems Management*, pp. 113 - 119, 1989.
- [4] S. B. Eom and H. Min, "The Changing Role of Multiple Criteria in Decision Support Systems," *Human Systems Management*, vol. 11, pp. 137 - 144, 1992.
- [5] J. Lu, M. A. Quaddus, and R. Williams, "A Framework and Prototype for Intelligent Multiple Objectives Decision Support System," presented at *1999 Asia Pacific Decision Sciences Institute Conference*, Shanghai, China, 1999.
- [6] R. K. Michel and B. M. Leif, *Knowledge-Based Decision Support Systems with Application in Business*: John Wiley & Sons, New York, 1995.
- [7] W. Reitman, "Generic Expert Systems for Management Applications: The Operations Advisor and the Management Advisor," *Computer Science Economic Management*, vol. 3, pp. 167-175, 1990.
- [8] D. Budgen and M. Marashi, "MDSE Advisor: Knowledge-Based Techniques Applied to Software Design Assessment," *Knowledge-Based Systems*, vol. 1, pp. 235-239, 1988.
- [9] P. H. Iz and L. R. Gardiner, "Analysis of Multiple Criteria Decision Support Systems for Cooperative Groups," *Group Decision and Negotiation*, pp. 61-79, 1993.
- [10] M. R. Klein and L. B. Methlie, *Knowledge-Based Decision Support Systems with Applications in Business*: John Wiley & Sons, Chichester, 1995.
- [11] M. A. Quaddus, "Computer Aided Learning of Multiple Criteria Decision Making: An Integrated System and Its Effectiveness," *Computers and Education*, vol. 28, pp. 103-111, 1997.
- [12] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Larichev, "Linear Programming with Multiple Objective Functions: Step Method (STEM)," *Mathematical Programming*, vol. 1, pp. 366-375, 1971.
- [13] P. D. Bois, J. P. Brans, F. Cantraine, and B. Mareschal, "Medics: An Expert System for Computer-Aided Diagnosis Using the PROMETHEE Multicriteria Method," *European Journal of Operational Research*, vol. 39, pp. 284-292, 1989.
- [14] K. Poh, "Knowledge-Based Guidance System for Multi-Attribute Decision Making," *Artificial Intelligence in Engineering*, vol. 12, pp. 315-326, 1998.
- [15] J. C. Pomerol, "From Aggregating by Rules to the Integration of Expert Systems in Multicriteria DSSs," presented at *The IEEE International Conference on Systems, Man and Cybernetics*, 1993.
- [16] P. Levine, M.-J. Pomerol, and R. Saneh, "Rules Integrate Data in a Multicriteria Decision Support System," *IEEE Transactions on Man, Machine and Systems*, vol. 20, pp. 678-686, 1990.
- [17] S. Pinson and P. Moraitis, "An Intelligent Distributed System for Strategic Decision Making," *Group Decision and Negotiation*, vol. 6, pp. 77-108, 1996.
- [18] W. Spangler, "The Role of Artificial Intelligence in Understanding the Strategic Decision-Making Process," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, pp. P145-159, 1991.
- [19] J. P. Ignizio, "The Determination of a Subset of Efficient Solutions via Goal Programming," *Computer and Operations Research*, vol. 8, pp. 9-16, 1981.
- [20] M. A. Quaddus and A. G. Holzman, "IMOLP: An Interactive Method for Multiple Objective Linear Programs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-16, pp. 462 - 468, 1986.
- [21] A. S. M. Masud and C. L. Hwang, "Interactive Sequential Goal Programming," *Journal of the Operational Research Society*, vol. 32, pp. 391-440, 1981.
- [22] J. P. Ignizio, *Goal Programming and Extensions*, Lexington Books, Massachusetts, 1976.
- [23] R. E. Steuer, "An Interactive Multiple Objective Linear Programming Procedure," *TIMS Studies in the Management Sciences*, vol. 6, pp. 225-239, 1977.
- [24] S. Zionts and J. Wallenius, "An Interactive Programming Method for Solving the Multiple Criteria Problem," *Management Science*, vol. 22, pp. 652-663, 1976.
- [25] K. L. Poh, M. A. Quaddus, and K. L. Chin, "MOLP-PC: An Interactive Decision Support Environment for Multiple Objective Linear Optimization," in *OR Applications in Singapore*, M. C. T. L. Goh, Ed.: Operational Research Society of Singapore, 1995, pp. 25-39.