The Design and Implementation of a Knowledge-Based Guide System in an Intelligent Multiple Objective Group Decision Support System

J. Lu[1], M.A. Quaddus[2,3]

[1] Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway, NSW 2007, Australia
jielu@it.uts.edu.au
[2] Graduate School of Business,
Curtin University of Technology, Perth, WA 6845, Australia
quaddusm@gsb.curtin.edu.au
[3] Corresponding author

Abstract

A multiple-objective decision support system (MODSS) may have several multiple objective decision making (MODM) methods in its method base. As some methods are more suitable than others for some decision cases or for some decision makers, a guide system is very useful to help users systematically towards the selection of their most appropriate methods. This paper presents an intelligent guide subsystem within a multiple-objective group decision support system (IMOGDSS). This intelligent guide subsystem is designed to have a knowledge base to represent human expertise in the specific domains of MODM methods. It first checks a decision maker' requirements (ability and preference), and then uses the knowledge to conduct inference to find a most suitable MODM method to the decision maker. The main compounds of the IMOGDSS are developed in a DELPHI environment. The knowledge-based intelligent guide subsystem is implemented by using an expert system shell CLIPS.

Keywords: Multiple objective decision making, Decision support systems, Knowledge-based systems, Intelligent decision support systems, Systems design

1. Introduction

A large number of methods of multiple-objective decision making (MODM) have been studied and developed because of the theoretical challenge and practical applications to a wide variety of problems. Some methods are more suitable and efficient than others in the solution of a particular decision problem for particular decision makers (DMs). Hence multiple objective decision support systems (MODSS) should preferably contain a sufficient number of MODM methods in its method base for the DMs' use. To utilize the potential of the method base effectively, an MODSS should be designed to have the capability of guiding the DMs to select and use the most suitable MODM methods from the method base for solving their decision problem (Bui & Sivasankaran, 1988; Pinson & Moraitis, 1996; Poh, 1998). A knowledge-based intelligent guide subsystem is necessary to achieve better guidance for the DMs during the decision-making process.

Since organizational decisions are primarily taken in a group, group decision making (GDM) involving MODM has also created interest as shown in the literature. A group based MODSS (MOGDSS) should be built to provide a group of DMs with feedback to individual preferences regarding possible solutions to MODM problems. It then can make an aggregation for all group members' solutions with a group aggregation-method base. A knowledge-based intelligent guide subsystem can be embedded in the group decision system in order to provide guidance during the whole group decision process. While the intelligent guide subsystem is applied in a group, each member can receive a series of guidances during the solution process based on his/her requirement, and can accept a recommendation for an appropriate method. Each member gets a satisfactory solutions through using a most suitable MODM method under the most fitting and proper guidance. These solutions then are aggregated into a compromised solution which represents all the members' preferences and expresses the most confident solution for the members of the group.

We first provided a framework for a knowledge-based intelligent multiple objectives decision support system in previous research (Lu, Quaddus & Williams, 1999). This framework was then expanded into a group working system, by embedding a group subsystem and a group aggregation method base. This framework has been implemented as a GUI based system prototype, and is called an intelligent multiple objective group decision support system (IMOGDSS). One of the advantages of this IMOGDSS is that is help DMs select the most efficient method(s) for each particular decision problem. It also allows the DMs to resolve complex problems that could not otherwise be solved with a single MODM, or to allow the DMs of a group to get solutions from different methods.

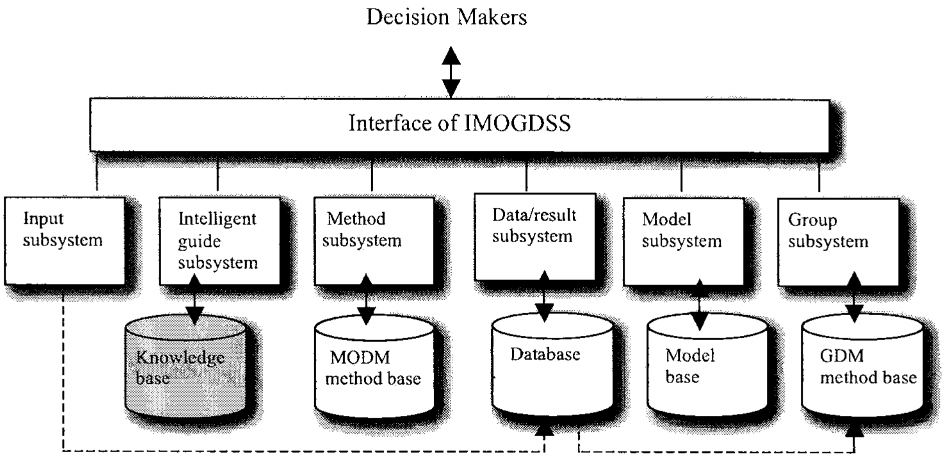


Fig. 1. System Architecture of IMOGDSS

This paper mainly describes the design and implementation of the intelligent guide subsystem. A prototype system framework of the IMOGDSS is proposed in Section 2. Section 3 describes the design process of the knowledge-based intelligent guide subsystem. Section 4 presents the inference process for finding a matched MODM method. The implementation of the intelligent guide subsystem is represented in Section 5. An illustrative example for the prototype application is given in Section 6. Finally, conclusions are presented in Section 7.

2. Framework of IMOGDSS

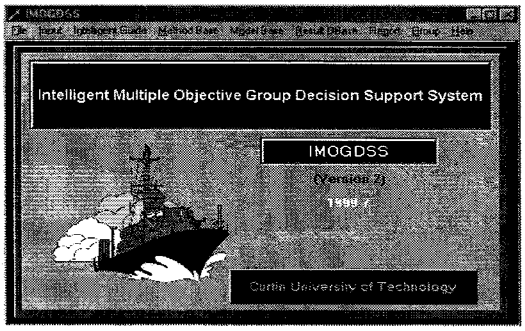
This section will present the architecture of the IMOGDSS and its main subsystems.

2.1 Architecture

The IMOGDSS is a GUI based integrated prototype. It allows selective and flexible use of many popular MODM methods under intelligent guidance. It allows non-technical DMs to interact fully with the system and access to a wide variety of data in a database and a model base. It also shows how a multiple objectives decision problem can be identified, analyzed and solved for a single DM or in a decision group. IMOGDSS has a database, a method base of MODM methods, a method base of GDM methods, a model base and a knowledge base. These resources are accessed by seven major subsystems, namely, the interface subsystem, input subsystem, intelligent subsystem, method subsystem, data/result management subsystem, model management subsystem and group subsystem. The overall architecture of IMOGDSS is shown in Fig.1.

2.2 Interface

The interface subsystem is used to integrate various other subsystems as well as to interact with DMs. It consists of a system desktop with a pull-down menu bar at the top (Fig. 2). There are nine sub-menus that form the functions in this system. The File sub-menu includes New Application, Open Data File, Open Model File, Print Data File, Print Model File and Exit. The Input sub-menu includes Decision Variables Input, Objectives Input and Constraint Input. The Intelligent guide sub-menu includes the Novice Intelligent Guide and the Intermediate Intelligent Guide. The Method sub-menu consists of seven methods' name. 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 Report sub-menu includes Single DM Report and Group Report. The Group sub-menu contains Open Solution File, Input Solution and Shortest Distance Aggregation Method, Weighted Shortest Distance Aggregation Method. The last sub-menu is Help.


IMOGDSS
File Input Intelligent Guide Method Base Model Base Result DBase Report Group Help

Intelligent Multiple Objective Group Decision Support System

IMOGDSS

(Version 2)

1999.7

Curtin University of Technology

Fig 2. System desktop of IMOGDSS

2.3 Method Subsystem

An MODM model considers a vector of decision variables, with multiple objective functions and constraints. The DMs attempt to maximize (or minimize) the objective functions. Since this problem rarely has a unique solution, each DM is expected to choose a solution from among the set of efficient solutions generated from the constraints.

Generally, the MODM problem can be formulated as follows:

$$\left.\begin{array}{l} \textit{Maximize } \{f_i(\bar{x}),\ i=1,\ldots\ldots,n\} \\ \textit{Subject to: } g_j(\bar{x}) \le b_j,\ j=1,\ldots.,m \end{array}\right\} \quad (1)$$

where $\overline{\mathrm{x}} = (x_1,\ldots\ldots,x_k)$ denotes k decision variables, $f_i(\overline{\mathrm{x}})$, $i=1,2,\ldots,n$ represent n conflicting objective functions, and $g_j(\bar{x}) \le b_j$, $j=1,2,\ldots,m$ represent m constraints.

Integrating the MODM methods into DSS to implement a MODSS have long been advocated by the researchers and users in both areas. An important reason for the emergence and development of MODSS is that MODM complements DSS and vice versa due to the differences in underlying philosophies, objectives, support mechanisms and relative support roles (Nazareth, 1993). The MODSS intends to provide the necessary computerized assistance to the DMs. The DMs are encouraged to explore the support tools available in an interactive fashion with the aim of finding a satisfactory solution. The ultimate success of MODSS lies in its ability to help the DMs to produce and to arrive at the 'best compromise' (and satisfactory) solution of MODM problems through direct interaction with analytical models (Eom, 1998).

We identified seven well-established MODM methods from the literature and put them in this IMOGDSS. These methods are: Efficient Solution via Goal Programming (ESGP) (Ignizio, 1981), Interactive Multiple Objective Linear Program (IMOLP) (Quaddus & Holzman, 1986), Interactive Sequential Goal Programming (ISGP), (Masud & Hwang, 1981), Linear Goal Programming (LGP) (Ignizio, 1976), Step Method (STEM) (Benayoun et al., 1971), Steuer (Steuer, 1977) and Zionts and Wallenius (ZW) (Zionts & Wallenius, 1976). These methods are developed as independent executables, to facilitate the flexibility required of the system.

2.4 Intelligent Guide Subsystem

The selection of the most suitable method from such a method base is always difficult to accomplish 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 model(s) with current decision needs. 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 DMs are required to identify the problems and to sequence them according to precedences and match each problem with appropriate MODM methods. Therefore, an intelligent technique is needed to support the selection of methods as one aspect of an intelligent application. A knowledge base system is utilized as an intelligent front-end of the IMOGDSS. This knowledge base system provides the guidance on the selection of suitable MODM methods according to different problem situations and DMs' situations.

2.5 Group Decision Making Subsystem

The process of MODM based GDM is divided into two stages. First, each DM makes a decision for an MODM problem. Second, DMs negotiate about their decisions so as to achieve a compromise solution of this MODM problem (Kersten, 1985). Decision groups are formed to exchange information and ideas, and to identify acceptable and desirable solutions. However, there is no rule for combining individual preferences into a group preference unless interpersonal comparison of utilities is allowed. Consequently, most utility group aggregation methods require explicit interpersonal comparisons of utility and follow a normative approach assuming that a group decision rule can be constructed by aggregating the utility functions of group members (Iz & Jelassi, 1990).

IMOGDSS has an MODM method base to support the first stage of the process. All members are supported by the same IMOGDSS to get an MODM solution through the same or different methods from the method base. Choosing from among different solutions provided by any MODM method, DMs take into account their preferences or wants, which take the form of objective functions. A GDM method base that contains two group aggregation methods is used in the second stage. The two GDM methods construct and follow a group decision rule that the best compromise solution has the shortest distance (or weighted shortest distance) to an "average solution". The GDM method base allows the DMs to select any aggregation method and allocates their MODM solutions as alternative solutions that are obtained by using different methods on an intelligent framework. Generally these group members have conflicting objectives because each member of the group represents a different business function, and not all members of the group have the same information and preference (i.e., goals of the objective functions). All members can work in a decision room or different locations, that is, it can be face-to-face (FTF) communication or computer mediated communication (CMC). The two GDM methods support FTF and CMC respectively. There may be a facilitator in a decision group meeting. He/she receives all solutions from each member by e-mails, discs or hard copies, and enters them into the GDM subsystem. Once a GDM method is determined an aggregation procedure will start. The facilitator has no influence on the final solution of the group. A 'best' compromise solution is then obtained through an interactive or non-interactive procedure within the group.

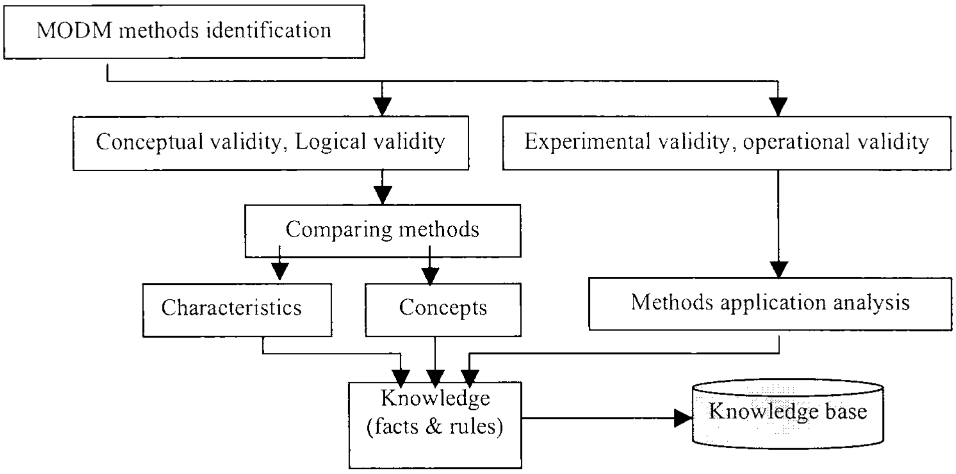

MODM methods identification
Conceptual validity, Logical validity
Experimental validity, operational validity
Comparing methods
Characteristics
Concepts
Methods application analysis
Knowledge (facts & rules)
Knowledge base

Fig. 3: knowledge acquisition process

3. Design of Intelligent Guide Sub-system

The section focuses on how to obtain knowledge about MODM methods, express the knowledge, and use the knowledge to design a question/response interface.

3.1 Knowledge Acquisition Process on MODM Methods

The knowledge acquisition is the process of capturing the experts' knowledge about a domain into a system. The process includes two main phases: the identification and collection of data (knowledge), then the representation of the facts representing the expertise to be kept in a system's knowledge base (Klein & Methlie, 1995). According to the definitions of Gabriella (1990), the following steps are used to identify and collect the knowledge about MODM methods (Fig. 3).

(1) Method identification: identifying a number of traditional and popular MODM methods based on literature review such as Hwang & Masud (1979), Poh, Quaddus & Chin (1995) to build an MODM method base.

(2) Validity recognition: a number of validities are recognized. They are conceptual validity, logical validity, experimental validity and operational validity.

(3) Methods comparison: comparing all methods included in this system through different points of view and classes.

(4) Characteristics and concepts identification: the characteristics and concepts of the MODM methods are identified.

(5) Selection of the type of knowledge representation: there are four main types of knowledge representation schemes in a knowledge base: production rules, semantic nets, frames and logic. We used the type of production rules.

3.2 Characteristics Analysis Models of MODM Methods

To build the knowledge base in the intelligent guide subsystem, the knowledge for the selection of MODM methods was first structured by capturing both the MODM methods and their characteristics. Based on Tecle & Duckstein (1992) and Poh's (1998) researches, the characteristics of MODM methods are classified into four classes, that is DMs-related, Methods-related, Problems-related and Solutions-related characteristics. By studying the characteristics of the seven methods implemented in the IMOGDSS prototype, four analysis models for the four characteristics are produced respectively.

The DMs-related characteristics analysis model includes the characteristics that are related to the DM preference for selecting a method to solve a decision-making problem. Some of these characteristics include the DMs' desire to interact with the system, the DMs' ability to provide data for a specific MODM method. The Methods-related characteristics analysis model consists of the characteristics that are related to the solution process of MODM methods, such as whether to use a linear programming technique or goal programming, whether to define an ideal solution. The Problems-related characteristics analysis model includes the characteristics that are dependent on the actual decision problem. For example, some MODM methods such as IMOLP and LGP require the provision of weights for each objective, while ISGP and LGP need to provide the goals for each objective. The Solutions-related characteristics analysis

Table 1. Characteristics (Char.) and facts related for the novice mode

Char. No.	Char. Name	Characteristic Definition	Char. Facts
1	Interaction	more interaction with the system	Char1
2	Subset	system provides a set of solutions	Char2
3	Unique	system provides a 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 the solutions (e.g. improving/ sacrificing the value of objectives)	Char6
7	Ideal	system defines an ideal solution	Char7
8	Weight	prepare the weight for every objective	Char8
9	Goal	prepare the goal for every objective	Char9
10	Priority	prepare the priorities for every objective	Char10

model consists of the characteristics that are related to the types of solution processed. Some MODM methods such as ESGP, ISGP, and LGP produce only a subset of the efficient solutions, while others such as STEUER produce all efficient solutions.

3.3 Novice and Intermediate Modes

In order to ensure the consistency of knowledge in a knowledge base, the principle of assimilation is applied for combining the characteristics in each characteristic model and to produce the characteristic-method models. To provide the appropriate guidance for the DMs possessing different levels of knowledge about MODM methods, we capture the characteristics into two groups in order to build the question models as a front-end for the knowledge base. Two groups of characteristics are provided, namely the novice and intermediate modes.

The novice mode includes non-technical characteristics that are applied to the DMs who are totally unfamiliar with MODM methods. The novice mode will correspond to a set of general non-technical questions regarding a decision problem, its expected solution(s) and its DMs' preferences. From the answers obtained from the DMs, a most suitable method will be found and recommended. A total of 10 characteristics are identified for the novice mode (Table 1).

The intermediate mode is designed for the DMs who are familiar with some concepts and methods of MODM, or not so familiar with the methods but have basic knowledge on solution process. The technical model consists of 14 characteristics of methods. It will be used to find methods

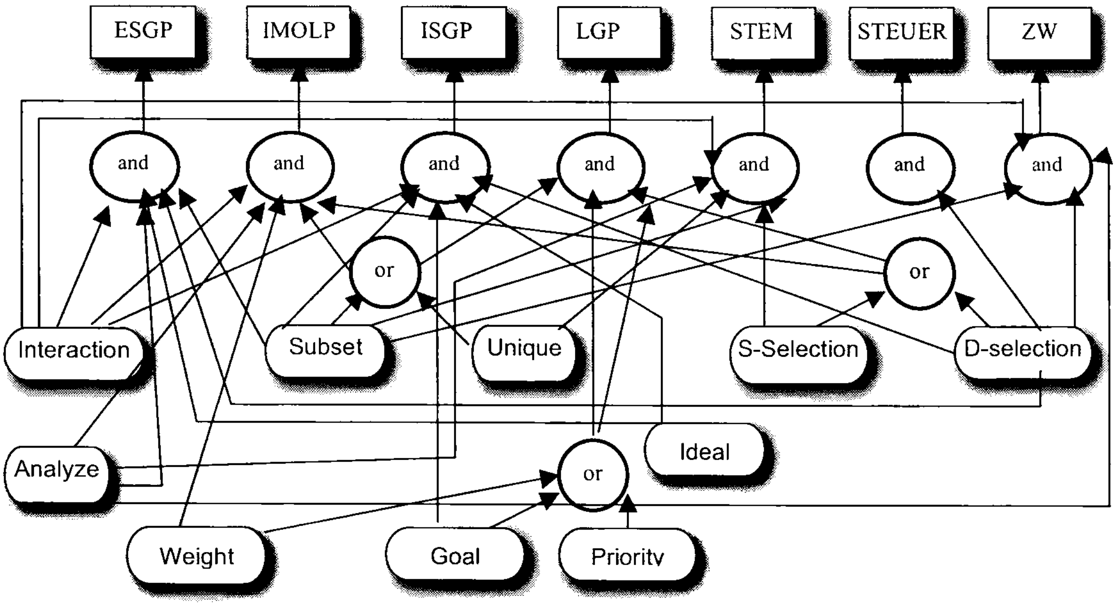


Fig. 4: Logical connectivity between MODM methods and their characteristics

Table 2. Question-answer-characteristic for novice mode

Question No	Questions	Responses	Char. Name	Char. No
Q1.	Would you like to have more interaction with the system?	T	Interactive	1
		B	Not	Not
Q2	Would you like the system to provide a set of solutions or a unique solution?	T	Subset	2
		B	Unique	3
Q3	Would you like the system to select one satisfactory solution or would you like to select a solution?	T	S-Selection	4
		B	D-Selection	5
Q4	Would you like to analyze solutions (e.g. improving/sacrifing the value of objectives)?	T	Analyze	6
		B	Not	Not
Q5	Would you like the system to define an ideal solution?	T	Ideal	7
		B	Not	Not
Q6	Have you prepared a weight for every objective?	T	Weight	8
		B	Not	Not
Q7	Have you prepared a goal for every objective?	T	Goal	9
		B	Not	Not
Q8	Have you prepared a priority for every objective?	T	Priority	10
		B	Not	Not

corresponding to a set of inputs for DMs using the intermediate mode. The DMs can discover which method(s) corresponds to a set of inputs by responding to some technical questions based on their problems, their desired solution and their data preparation.

3.4 Logical Connectivity of MODM Methods and their Characteristics

Bui & Sivasankaran (1988) discussed 4 multiple attribute decision making (MADM) methods for matching their 9 assertions. Poh (1998) identified the relationship between 17 MADM methods (some of them were not implemented) and their 19 characteristics. In our project, 7 MODM methods included in IMOGDSS are thoroughly studied and classified according to one or more of the 10 characteristics for the novice mode and 14 characteristics for the intermediate mode. Fig. 4 shows the logical connectivity between the MODM methods and the 10 characteristics for the novice mode. As an example shown in Fig. 4, the ISGP (M3) method is characterized by the characteristics of the 'interaction', 'subset', 'D-selection', 'ideal', and 'goal'.

3.5 Questions and Responses

The two groups of questions are created based on the two modes and shown to the two levels of DMs through a series of dialog boxes respectively. Each dialog box shows one question, with two response items: Top (T) and Bottom (B) and a set of weights to choose. Each DM can choose one of responses and then go down to the next question. They can also go back to the previous question to change their responses, or exit the question dialog box at any question by taking the default values of responses. These responses are used to match the characteristics of one method and the weights are used to measure which method is the most appropriate if no method fully matches with one DM's preference. The relationships between questions, responses and characteristics are shown in Table 2 for the novice mode.

4. Inference Process for Finding Matched MODM Methods

This section first presents the definitions of completed match and n-step match. An ignoring characteristic match strategy is then introduced.

4.1 Completed Match and *n*-Step Match

We only discuss the Novice mode, as the discussion for the intermediate mode is similar.

Let $M = \{M_1, M_2, ..., M_7\}$ be a method set, $\overline{C}$ be a characteristics set of MODM methods, $C_i = (C_{i1}, C_{i2}, ..., C_{ik})$ and $C_{ij} \in \overline{C}$ (*j=1,2,..,k*) be characteristics of M_i, $R = (R_1, R_2, .., R_k)$ and R_i (*i=1,2...,k*) be characteristics of the DMs preferences (it is covered by the responses of the DMs for the questions) and for any $p \in \{1, 2, ..., k\}$ there exists an i and j such that $C_{ij} = R_p$, $W = (W_1, W_2, .., W_k)$ be a weight vector for R, k =10 for novice mode and k =14 for intermediate mode.

Definition **1**. *RC* Completed match

If there exists $i \in \{1,2,..,7\}$ such that for any $j \in \{1,2,..,k\}$,

$$R_j = C_{ij}.$$

we then say R and M_i is a RC completed match and denote it as $R \equiv C_i$, or $R^0 = C^0{}_i$. A completed match means the characteristics of a method completely match with the DMs preferred characteristics.

Definition **2**. *RC n*-step match

Let $R^n = (R_{j1}, R_{j2}, ..., R_{jk-n})$, and $\{R_{j_1}, R_{j_2}, ..., R_{j_{k-n}}\} \subset \{R_1, R_2, .., R_k\}$, $n = 1, 2, ..k-1$.

If there exists $i \in \{1,2,..,7\}$ such that $\forall\, j \in \{j_1, j_2, .., j_{k-n}\}$,

$$R_j^n = C_{ij},$$

we then say *R* and M_i is a *RC n*-step match and denote it as $R^n = C_i^n$, and *n* is called a match degree, where

$C_i^n = (C_{ij_1}, C_{ij_2}, ..., C_{ij_{k-n}})$ and
$C_i^n = \{C_{ij_1}, C_{ij_2}, ..., C_{ij_{k-n}}\} \subset \{C_{i1}, C_{i2}, .., C_{ik}\}$.

An *n*-step match means that only *k-n* characteristics of a method match with the DMs preferred characteristics.

Theorem **1**. If for any $i \in \{1,2,..,7\}$, and *R* and M_i is not a completed match, then there exists *n<k*, such that *R* and M_i is *RC n*-step match.

Proof.

(1). If there exists $k>m\geq1$ such that $w_{i_j} = \min\{w_1, w_2, ..., w_k\}, j = 1,2,...m$. Since $\{ i_1, i_2, ..., i_m \}$ is an order set, we can take j_0 such that j_0 is the least element of $\{ i_1, i_2, ..., i_m \}$.

If $i_{j_0} = 1$, we take

$$R^1 = (R_2, ..., R_k)$$
$$C_i^1 = (C_{i2}, ..., C_{ik})$$
$$W^1 = (W_2, ..., W_k)$$

If there exists *i∈{1,2,..,7}*, such that

$$R^1 = C_i^1$$

Then *R* and M_i is *1*-step match.

If $1< i_{j_0} <k$, we take

$$R^1 = (R_1, .., R_{i_j-1}, ..., R_{i_j+1}, ..., R_k)$$
$$C_i^1 = (C_{i1}, ..., C_{ij_{j_0}-1}, C_{ij_{j_0}+1} ..., C_{ik})$$
$$W^1 = (W_1, ..., W_{i_{j_0}-1}, W_{i_{j_0}+1}, ..., W_k)$$

If there exists *i ∈{1,2,..,7}*, such that

$$R^1 = C_i^1$$

Then *R* and M_i is *1*-step match.

If $i_{j_0} = k$, we take

$$R^1 = (R_1, ..., R_{k-1})$$
$$C_i^1 = (C_{i1}, ..., C_{ik-1})$$
$$W^1 = (W_1, ..., W_{k-1})$$

If there exists *i∈{1,2,..,7}*, such that

$$R^1 = C_i^1$$

Then *R* and M_i is *1*-step match.

(2). If $w_1 = w_2 = ... = w_k$, we take

$$R^1 = (R_2, ..., R_k)$$
$$C_i^1 = (C_{i2}, ..., C_{ik})$$
$$W^1 = (W_2, ..., W_k)$$

If there exists *i∈{1,2,..,7}*, such that

$$R^1 = C_i^1$$

Then *R* and M_i is *1*-step match.

(3) If *R* and M_i is not a *1*-step match, we replace R^1, C_i^1 *and* W^1 with *R*, C_i *and W*, and repeat the above-mentioned process. If *R* and M_i is a *2*-step match, we have finished the proof of this theorem. If *R* and M_i is not a 2-step match, we replace R^2, C_i^2 *and* W^2 with R^1, C_i^1 *and* W^1 and repeat the above process again. Finally, as for any $p \in \{1,2,...,k\}$ there exists a *i* and *j* such that $C_{ij} = R_p$, there exists *n<k* such that *R* and M_i is an *n*-step match.

4.2 An Ignoring Characteristic Match Strategy (ICMS)

DMs have different real decision problems, different knowledge backgrounds and different preferences for decision making. Their different favorite choices for each question response and intensity of importance of these responses are obtained by using a set of question dialog boxes and weight boxes. The responses and weight marks are converted to a response vector *R* that consists of the characteristics the DM needs, and a weight vector *W* that consists of the weight of each characteristic. If a DM's responses are an *RC* completed match with the characteristics of a MODM method, this method is recommended and the Ignoring Characteristic Match Strategy (ICMS) is not executed. However, it is not often that a DM's responses exactly match the characteristics of one method (that is an *RC* completed match). An ICMS is thus used based on *Theorem 1* to find M_{i0} such that a *RC n*-step is found. The objective of this method is to combine

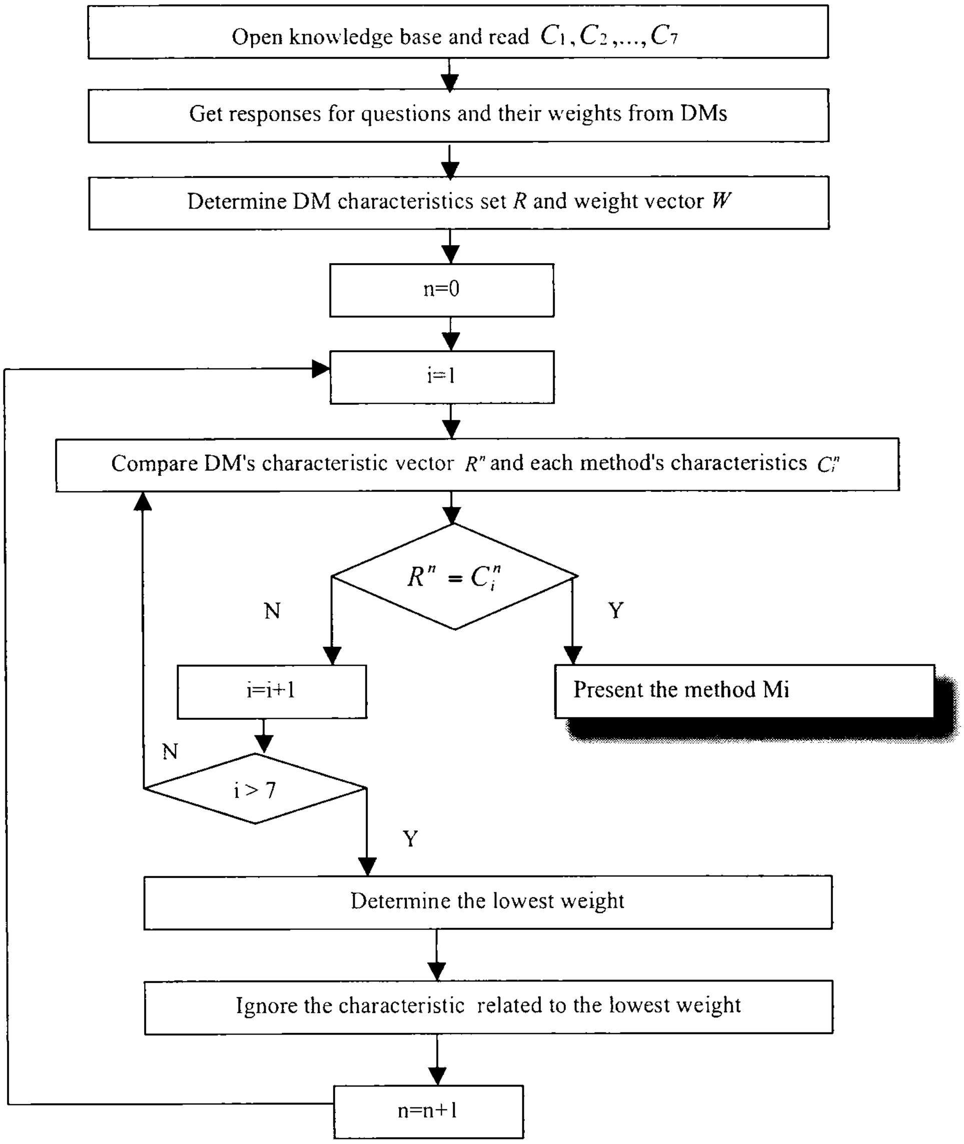

Open knowledge base and read $C_1, C_2,..., C_7$
Get responses for questions and their weights from DMs
Determine DM characteristics set R and weight vector W
n=0
i=1
Compare DM's characteristic vector R^n and each method's characteristics C_i^n
$R^n = C_i^n$
N
Y
i=i+1
Present the method Mi
N
i > 7
Y
Determine the lowest weight
Ignore the characteristic related to the lowest weight
n=n+1

Fig.5 Ignoring characteristic process

the DMs' preferences and the weights for each characteristic to find a most suitable method that best satisfies the DM's requirement.

The algorithm is centered on an ignoring process based upon a weight vector W whose elements represent the intensity of the importance of the characteristics that the DMs prefer. Through this weight vector, a lowest weight element W_l ($1 \le l \le k$) is obtained from weight vector W. R_l and Ci_l ($i=1,...,7$) that according to W_l are then found and ignored, if for any i, R *and* Ci is not a completed match. If there is an existing M_i such that R and C_i is a 1-step match, this method M_i is then recommended to the DMs. Otherwise, the second lowest weight is determined, another characteristic is missed and a RC 2-step match is measured. Based on *Theorem 1*, an n-step match method will be found after ignoring process *(n<k) n* times. Fig.5 shows the ignoring process of the ICMS. The ICMS is expected to cause the least regret among the DMs and the greatest comfort and the best coefficient for the method selection process.

4.3 Determination of Weight Vector

4.3.1 Users Weight Vector

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

Table 3. Characteristics-Method Matrix for the Novice mode

Char Fact	ES GP	IMOLP	ISGP	LGP	STEM	STEUER	ZW	Pi
Char1	●	●	●		●		●	5
Char2	●	●	●	●		●	●	6
Char3		●		●	●			3
Char4		●		●	●			3
Char5	●	●	●	●		●	●	6
Char6	●	●			●		●	4
Char7	●		●		●			3
Char8		●		●				2
Char9			●	●				2
Char10				●				1

Table 4. The weights and ignoring ranking

Char NO	Char Name(Ĉ)	P	Q	Q	W	Ignore Ranking (n)
1	Interaction	5	VH	8	6.5	(10)
2	Subset	6	E	6	6	(8)
3	Unique	3	E	6	4.5	(6)
4	S-Selection	3	E	6	4.5	(7)
5	D-Selection	6	E	6	6	(9)
6	Analyze	4	A	4	4	(5)
7	Ideal	3	A	4	3.5	(4)
8	Weight	2	H	2	2	(2)
9	Goal	2	H	2	2	(3)
10	Priority	1	H	2	1.5	(1)

method completely matches *R*. The characteristic corresponding to the second lowest weight is missed if no method is found to meet the 1-step match. According to this methodology, two different methods may be recommended to two different DMs because they are assigned different weights for characteristics even though their responses for the questions were the same.

Four levels of the weights are defined in the system. The DMs can choose any one from a dialog box for each question. The choices are:

- Very important,
- Important,
- General,
- Less important.

$$W_i = \begin{cases} 2 & \text{if } R_i \text{ is less important} \\ 4 & \text{if } R_i \text{ is general} \\ 6 & \text{if } R_i \text{ is important} \\ 8 & \text{if } R_i \text{ is very important} \end{cases}$$

The DMs' assignment for the weight can generate a weight vector $W=(W_1,W_2,\ldots,W_k)$, Where based on *Theorem 1* the set of weights can be ranked and for any $i<j$, when $W_i = W_j$, W_i, is first taken.

4.3.2 System Weight Vector

This system allows the DMs to default the weight assignment. In this case the system will provide a weight vector *W* automatically. This weight vector *W* is defined as follows:

$$W = (W_1, W_2, \ldots, W_k)$$
$$W_i = (P_i + Q_i)/2, i = 1, 2, \ldots, k,$$

where $P_i \in P$ is a relative coefficient of the characteristic set $C_i \subseteq \overline{C} = C_1 \cup C_2 \cup \ldots C_7$, Q_i is a changeable coefficient for R_i, $i=1,2,\ldots,k$.

The vector *P* is obtained through analyzing the relative degree C_i and M_i, $i=1,2,\ldots,7$ in Table 3. For example, Char1 is relative with five methods, so P_1=5. For example, Char1 is relative with five methods, so =5. If a characteristic has a higher relative coefficient then it should have a higher weight. We have obtained

$$P = (P_1, P_2, \ldots\ldots, P_{10}) = (5,6,3,3,6,4,3,2,2,1)$$

Some changes in the DMs' request for characteristics can be easily accepted by the DMs, others are harder to accept. The acceptability for changes is thus considered. The objective is to produce the least influence upon the DMs' selection process when a characteristic is ignored. We define

E -- easy to accept the changes;
L -- less hard to accept the changes;
H -- hard to accept the changes;
VH -- very hard to accept the changes,

and the scores of E, L, H and VH are 2, 4, 6 and 8 respectively. We have a vector

$$Q = (Q_1, Q_2, \ldots, Q_{10}) = (8,6,6,6,6,4,4,2,2,2)$$

If a characteristic is hard to be accepted for changes by a DM then it should have a high weight. Therefore, we generate the *W* vector.

$$W = (6.5, 6, 4.5, 4.5, 6, 4, 3.5, 2, 2, 1.5).$$

In this case the characteristic 10 in relation to the least weight (1) will be ignored when a failed search match happened, while characteristic 1 in relation to weight (6.5) will not be ignored during the whole searching process. Table 4 shows the vectors *P*, *Q*, *W* and the ignore (missing) ranking.

5. Implementation of the Intelligent Guide Subsystem

This section presents the design of the facts, rules, and inference process for selecting a MODM method.

5.1 Composition

The intelligent guide subsystem consists of the question subsystem, response subsystem, method-show subsystem, ICS subsystem, main-control subsystem and a knowledge base that includes a set of facts to define the knowledge about the methods and a set of rules for finding a suitable method.

IMOGDSS uses the inference engine provided by the expert system shell CLIPS. 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 ICS subsystem. If a suitable method(s) is found the name of method(s) will be displayed to the DMs at the end of the inference process by the method-show subsystem, else a fuzzy (*n*-step) matched method with the lowest match degree *n* is provided through related facts as asserted by the ICS subsystem.

5.2 Design of the Facts

Facts are one of the basic high-level forms for representing information in a knowledge base system. Each fact represents a piece of information that has been in the current list of 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.

The knowledge is represented using 'def-templates' and 'def-facts'. Every 'def-facts' defines directly a fact. A def-template defines a group of related fields in a pattern similar to the way in which a record is a group of related data. Definitions of three pieces of def-templates and def-facts are as shown in the following code:

(1) *method:* seven MODM methods and their various characteristics.

```
(deffacts  Method1
      (Method
         (Number 1)
         (Name ESGP)
         (Char1 interaction)
         (Char2 subset)
         (Char5 d-selection)
         (Char6 analyze)
         (Char7 ideal)
      )
)
```

(2) *response*: A set of facts relating the response of each question to the facts to be asserted by the inference engine into the working memory;

```
 (deftemplate Response
      (field Question
        (type INTEGER)
        (default ?NONE)
      )
      (field Answer
         (type INTEGER)
         (default 0)
      )
)
```

(3) *characteristics-question:* A set of facts relating each characteristic to its corresponding question;

```
(deftemplate Char-to-Quest
      (field Char
         (type SYMBOL)
         (default ?NONE)
      )
      (field Quest_No
         (type INTEGER)
         (default ?NONE)
      )
)
```

5.3 Design of Rules

Rules are used to represent heuristics to specify a set of actions to be performed for a given situation. This study defines a set of rules which collectively work together for method selection. The method selection knowledge base system attempts to match all the characteristics of a method to those already asserted into the working memory. If the match failed, a characteristic which has the least weight will be ignored/missed. A method will be selected if all its characteristics (or after missing) are found in the working memory. We have also incorporated many heuristics that assist the system in the conflict resolution phase of the inference. For example, the rule to inform the user that a suitable method has been found shall have priority over other rules. The definitions of two examples of rules are shown in the following code:

(1) *call-question:* a rule relating to get the questions' number and its responses' number.

```
(defrule get-question
   declare (salience 10))
   ?v1 <- (Question (Number ?num1))
   test (neq ?num1 -1))
=>
    (retract ?v1)
    (bind ?response (quest ?num1))
    (assert (Response
       (Question ?num1)
       (Answer ?response))
     )
)
```

(2) *question-action:* after asking the DMs a question and getting the response, the rule checks and compares the question number, answer number and facts between facts question_answer_action and response. If the numbers of question and answer match one of the facts question_answer_action, the numbers of question and fact are asserted.

```
(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))
      )
)
```

All patterns must be satisfied by facts in the fact-list for the rules to fire. A program will not start running unless there are rules whose left-hand side (LHS)'s are satisfied by the facts. The inference engine sorts the activations according to their salience. This sorting process eliminates the conflict of deciding which rule should fired next.

5.4 N-Step Match & Inference Process

All questions concerning method selection are shown on serial question-boxes. The DMs answer each question by clicking each Radio button. Every question-box window includes a weight-box which contains four Radio buttons: "very important", "important", "general importance" and "less important". The DMs should choose one of them to indicate the degree of importance of this question at the same time as answering each question. After inputting these responses, including all answers for the questions and their degree of importance, the intelligent subsystem puts them in the response subsystem and ICS subsystem respectively. The data in the response subsystem will be then converted into the CLIPS-facts and is asserted to the fact base as the DM's response facts. The knowledge base attempts to match the characteristics of a method to the characteristics already asserted in the working memory. A suitable method(s) is found once all its characteristics are matched with those in the working memory. The data in Weight-Array will be applied to find an *n*-step match method when an *RC* completed matched method doesn't exist.

The preliminary work is very important, however. Before running the inference engine, the intelligent subsystem first links with CLIPS. A support program TclipsFact is then embedded in this intelligent subsystem so that all CLIPS' functions and operations can be executed in a DELPHI environment. A DELPHI-CLIPS interface program is used to support the execution of the CLIPS operations in the DELPHI working environment and to provide a user interface. We don't build any CLIPS' user interfaces because (1) the design ability provided by DELPHI for interfaces is better than CLIPS; and (2) a software package should keep a unified style of interface. In this DELPHI-CLIPS interface program, the intelligent subsystem can assert a set of facts by a public method or function, such as AssertString through the TClips code. The subsystem also can use FactCount and Fact properties for getting all the facts in the fact base, such as the Assert and Retract method to assert and retract a fact. The Tclips component also has a set of events to be used. We can use them to monitor CLIPS and its execution.

When CLIPS is called, the intelligent guide subsystem first checks if the CLIPS supporting files are in the correct location. The subsystem then calls "InitializeCLIPS" for initialization. The subsystem again calls procedure "Clear" to clear the fact base. The next step is to load the CLIPS file that includes all fixed facts and rules. After this file is loaded, the subsystem executes "Reset procedure" and all fixed facts are entered into the agenda. The last function, "Run" is then called. All responses of the DMs will be converted into the facts and the intelligent subsystem asserts them in the fact base. At the same time, the rules are fired and the subsystem starts an inference process. The CLIPS system attempts to match the patterns of rules with the facts in the fact-list. If all the patterns of a rule match the facts, the rule is activated and put on the agenda. The agenda is a collection of activations that are those rules that match pattern entities. The intelligent subsystem and its working principle are shown in Fig.6.

5.5 Method Selection Procedure

Based on the working principles of the intelligent subsystem, Fig. 7 shows the method selection process.

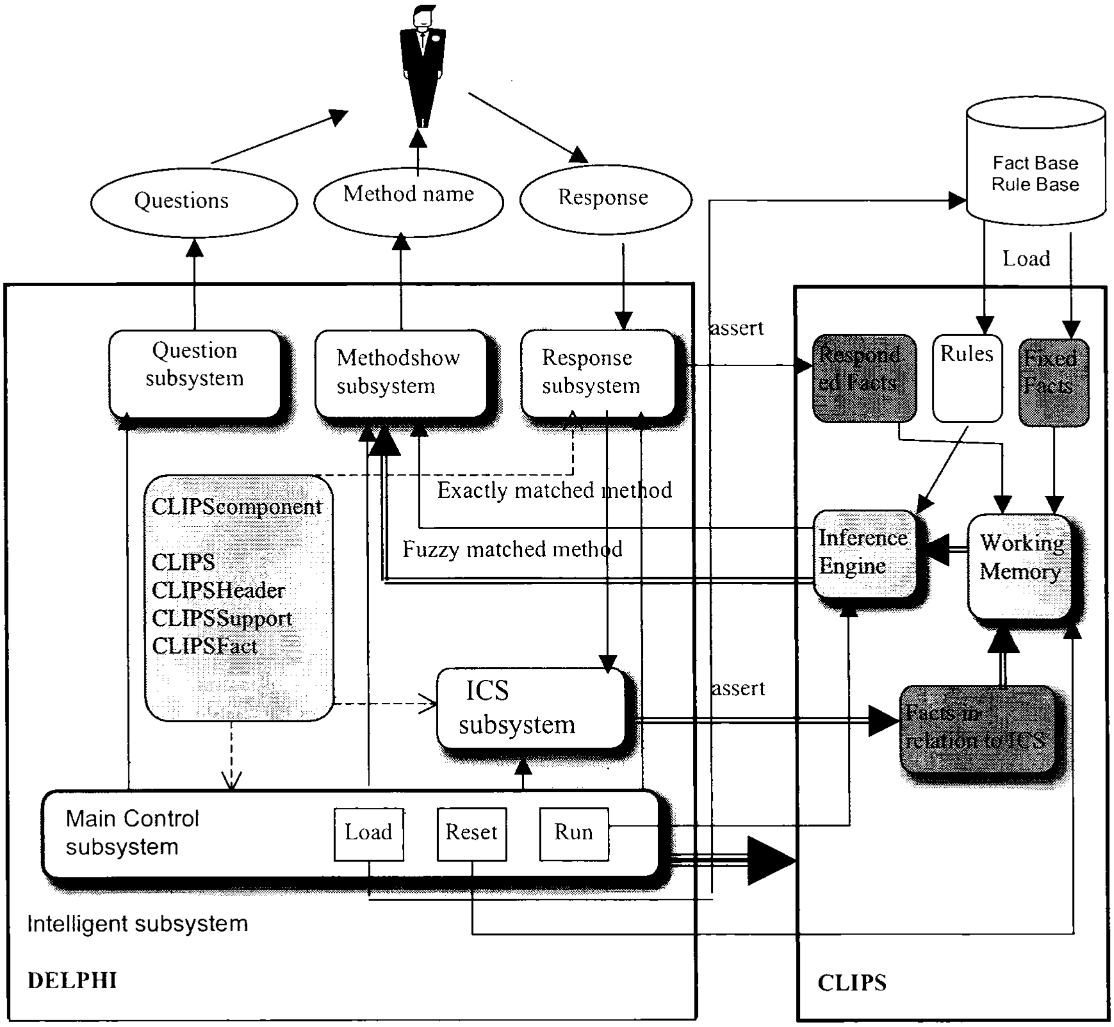


Fig. 6. Intelligent guide subsystem and its working principle

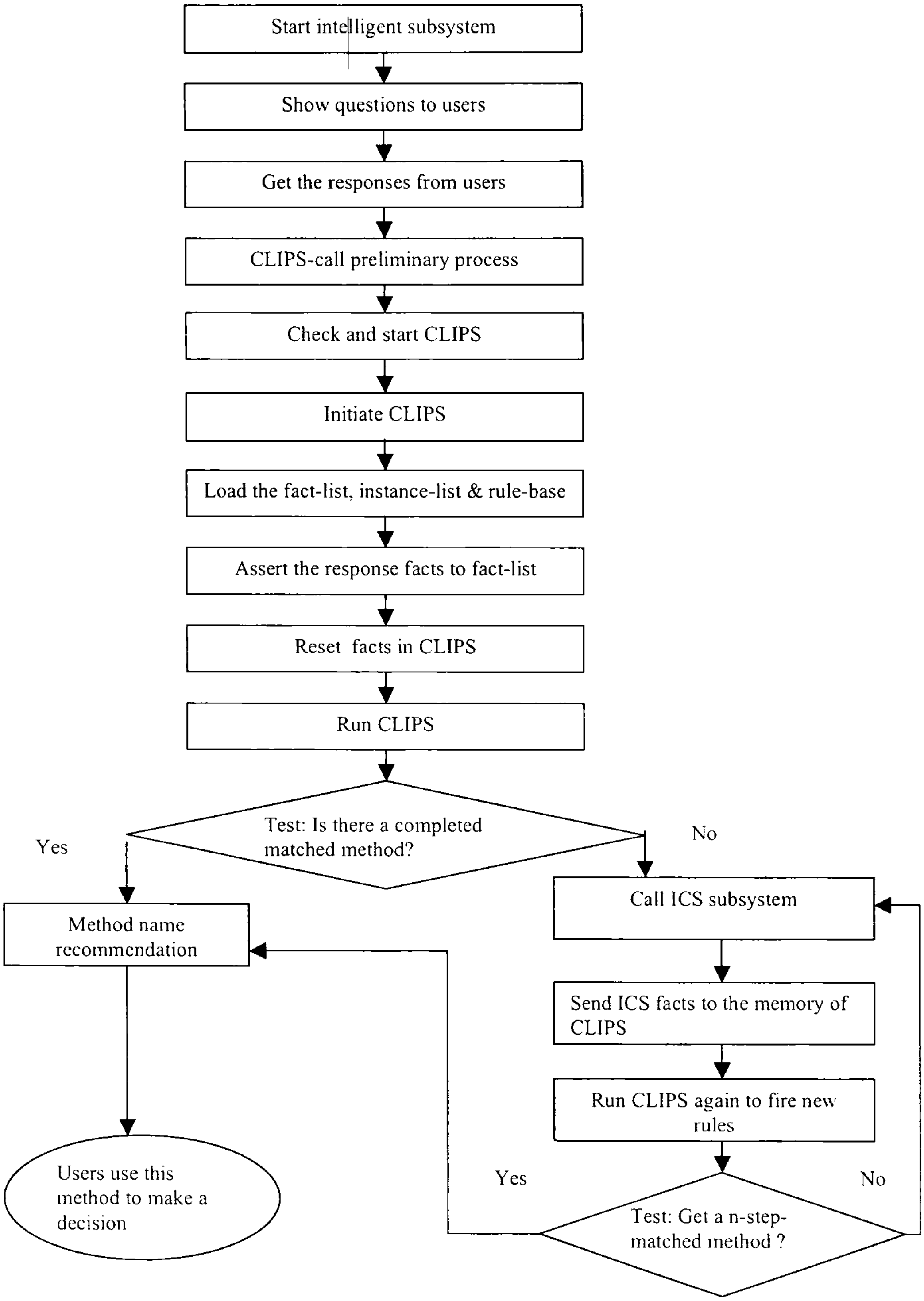
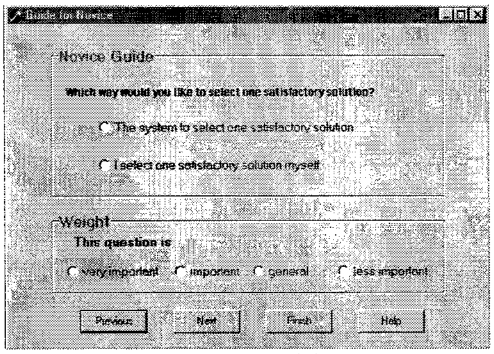

Start intelligent subsystem
Show questions to users
Get the responses from users
CLIPS-call preliminary process
Check and start CLIPS
Initiate CLIPS
Load the fact-list, instance-list & rule-base
Assert the response facts to fact-list
Reset facts in CLIPS
Run CLIPS
Test: Is there a completed matched method?
Yes
No
Method name recommendation
Call ICS subsystem
Send ICS facts to the memory of CLIPS
Run CLIPS again to fire new rules
Users use this method to make a decision
Yes
No
Test: Get a n-step-matched method ?

Fig 7: Method Selection Process



Guide for Novice

Novice Guide

Which way would you like to select one satisfactory solution?

○ The system to select one satisfactory solution

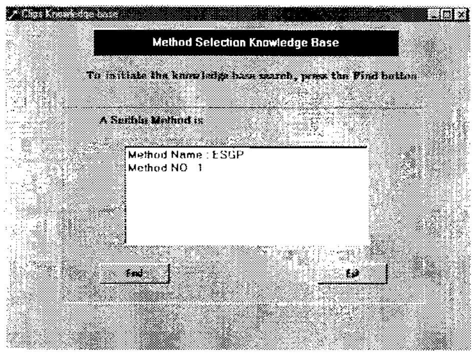
○ I select one satisfactory solution myself

Weight

This question is

○ very important ○ important ○ general ○ less important

Fig. 8 A question and the DM's response



Clips Knowledge base

Method Selection Knowledge Base

To initiate the knowledge base search, press the Find button

A Suitable Method is

Method Name : ESGP
Method NO 1

Fig. 9 A recommendation

6. An Application for the Intelligent Guide System

An example will illustrate the application of the intelligent guide subsystem and the IMOGDSS in this section.

6.1 A MODM Problem and a Decision Group

A manufacturing company has six machine types - milling machine, lathe, grinder, jig saw, drill press and band saw - whose capacities are to be devoted to produce three products X1, X2 and X3. The DM has three objectives of maximizing profits, quality and worker satisfaction (Lai, 1995). The initial input data is listed in Table 5. A group of three DMs A, B and C are assumed to be responsible for determining a compromise solution to the MODM problems. The MODM model is shown below.

6.2 An Intelligent Guide Process for Selection of MODM methods

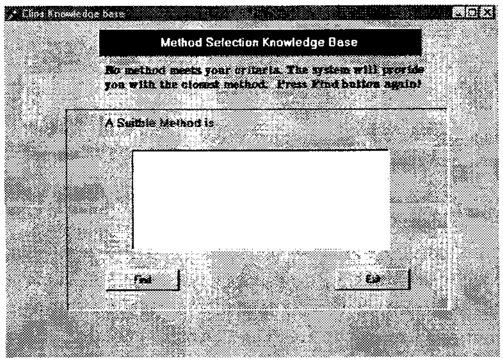
When this guidance system starts up, a set of questions will be displayed in a series of "Radio Group" dialog boxes. Fig. 8 shows a Radio Group dialog box. The DMs can choose one of the answers and mark one of the weight values. The DMs can also go down to the next question, back to the previous question to change their responses and stop the question scheme at any question.

Consider the above-mentioned MODM problem. The DM *A* is assumed to be the marketing officer of this company and to be a novice at applying an MODSS. The DM *A* likes to get recommendation from the intelligent guide system for a suitable MODM method. The DM *A* runs the intelligent guide subsystem and answers all questions. His

Table 5 Production data

Machine	*Product x1*	*Product x2*	*Product x3*	*Machine hours available*	*Unit cost ($)*
Milling machine	12	17	0	1400	20
Lathe	3	9	8	1000	30
Grinder.	10	13	15	1750	25
Jig saw	6	0	16	1325	25
Drill press	0	12	7	900	35
Band saw	9.5	9.5	4	1075	20
Profits	**50**	**100**	**17.5**		
Quality	**92**	**75**	**50**		
Worker Satisfaction	**25**	**100**	**75**		

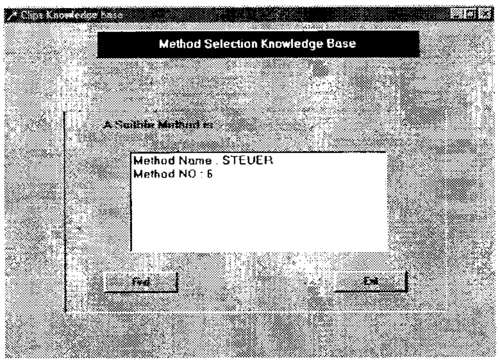
$$
\begin{aligned}
&Max\ \ f_1(x) = 50x_1 + 100x_2 + 17.5x_3\\
&Max\ \ f2(x) = 92x1 + 75x2 + 50x3\\
&Max\ \ f3(x) = 25x1 + 100x2 + 75x3\\
&\ \ subject\ \ to:\\
&\qquad g1(x) = 12x1 + 17x2 \le 1400\\
&\qquad g2(x) = 3x1 + 9x2 + 8x3 \le 1000\\
&\qquad g3(x) = 10x1 + 13x2 + 15x3 \le 1750 \qquad (1)\\
&\qquad g4(x) = 6x1 + 16x3 \le 1325\\
&\qquad g5(x) = 12x1 + 7x3 \le 900\\
&\qquad g6(x) = 9.5x1 + 9.5x2 + 4x3 \le 100\\
&\\
&\qquad x1, x2, x3 \ge 0
\end{aligned}
$$



No method meets your criteria. The system will provide you with the closest method. Press Find button again!

A Suitble Method is

Fig. 10. To find an n-step match method



A Suitble Method is

Method Name . STEUER
Method NO : 6

Fig. 11 A 1-step match method is recommended

responses for the eight questions and weights for each question are shown in Table 6. After the Finish button is clicked, the system goes to the knowledge base supported by CLIPS. The DM *A*'s responses match exactly the characteristics of method ESGP. The name of ESGP is shown in the screen as a recommendation (Fig. 9). As a result, the DM *A* starts to use ESGP method to solve this decision problem.

Another DM ***B*** is assumed to be the finance officer of this company and be a novice.

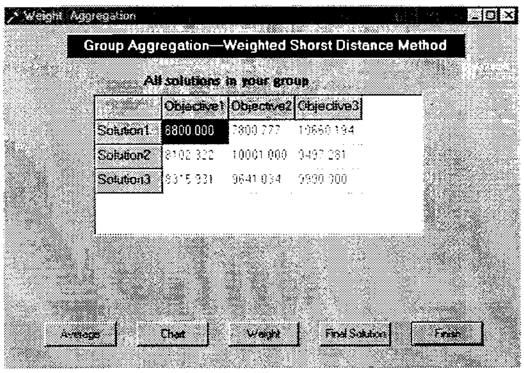
The DM ***B'*** responses and weights for the eight questions are shown in Table 7. The system couldn't find any method to match completely his needs (Fig. 10). After the DM ***B*** presses the "Find" button the system applies "Ignoring (missing) characteristic strategy (ICS)" to find a 1-step matched method. Characteristics 6 (analysis) has the lowest weight and so it is ignored. A 1-step matched method named STEUER is then found and recommended (Fig. 11). DM ***C*** is an "intermediate" person and works as a manufacturing officer. After responding to 12 questions, a fuzzy (n-step) matched method named IMOLP is recommended. ***C*** uses IMOLP to solve this decision problem.

Table 6. *A*'s responses for questions and weights for Novice mode

No	Questions	Answer	Weight
Q1.	Would you like to have more interactive with the system?	Yes	8
Q2	Would you like the system to provide a set of solutions or a unique solution?	System to provide a set of solutions	6
Q3	Would you like the system to select one satisfactory solution or would you like to select a solution yourself?	Select a solution myself	4
Q4	Would you like to analyze solutions (e.g. improving/sacrifing the value of objectives)?	Analysis	4
Q5	Would you like the system to define an ideal solution?	Yes	2
Q6	Have you prepared a weight for every objective?	Have not prepared	6
Q7	Have you prepared a goal for every objective?	Have not prepared	4
Q8	Have you prepared a priority for every objective?	Have not prepared	4

Table 7. *B*'s responses for questions and weights for Novice mode

No	Questions	Answer	Weight
Q1.	Would you like to have more interactive with the system?	No	6
Q2	Would you like the system to provide a set of solutions or a unique solution?	System to provide a set of solutions	8
Q3	Would you like the system to select one satisfactory solution or would you like to select a solution yourself?	Select a solution myself	8
Q4	Would you like to analyze solutions (e.g. improving/sacrifing the value of objectives)?	Analysis	2
Q5	Would you like the system to define an ideal solution?	No	4
Q6	Have you prepared a weight for every objective?	Have not prepared	6
Q7	Have you prepared a goal for every objective?	Have not prepared	4
Q8	Have you prepared a priority for every objective?	Have not prepared	4



Weight Aggregation

Group Aggregation—Weighted Shorst Distance Method

All solutions in your group

	Objective1	Objective2	Objective3
Solution1	8800.000	7800.777	10660.194
Solution2	8102.322	10001.000	9497.281
Solution3	8315.931	9641.034	9990.000

Fig 12: Three solutions from DMs

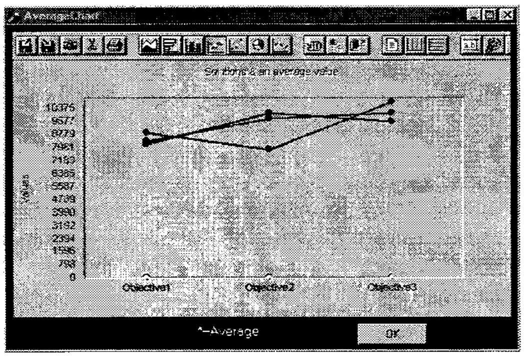
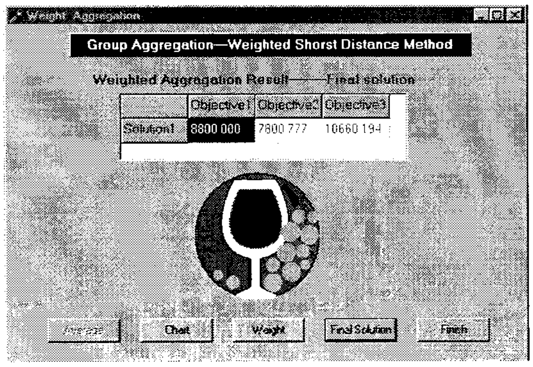


Fig 13: Three solutions are shown by a chart



Weight Aggregation

Group Aggregation—Weighted Shorst Distance Method

Weighted Aggragation Result——Final solution

	Objective1	Objective2	Objective3
Solution1	8800.000	7800.777	10660.194

Fig 14: A Final compromise Solution

6.3 Aggregation Process

The DMs *A, B* and *C* applied the ESGP, STEUER and IMOLP respectively to solve the MODM problem. This MODM problem involves three conflicting objectives, and *A, B* and *C* are each responsible for one of three functional areas involved in the three objectives of the company. Therefore three different solutions are obtained from the three members. These solutions are passed respectively to the group subsystem (Fig. 12) and shown these solutions by a chart (Fig. 13). A group aggregation method in the GDM methodology, "weighted shortest distance method", is applied to reach a 'best' compromise solution through applying a set of degrees of importance for each objective function from each DM. Fig. 14 shows the final compromise solution of the group.

7. Conclusions

The role of the intelligent guide subsystem presented in this paper is to provide a MODM method selection function for decision makers. This intelligent guide subsystem has a knowledge base about the characteristics of MODM methods. It uses the knowledge to check a decision maker' requirement (ability and preference), and then conducts inference to find a most suitable MODM method to the decision maker. By providing decision makers with the best suitable methods, the IMOGDSS is more easily and flexibly to be used in practical decision making.

References

[1] Benayoun, R., de Montogolfier, J., Tergny, J. & Larichev, O. 1971, 'Linear programming with multiple objective functions: step method (STEM)', *Mathematical Programming*, Vol. 1, no. 3, pp. 366-375.

[2] Bui, X. T. & Sivasankaran, T. R. 1988, 'An Intelligent Front End for MCDM Based Decision Support Systems', in Proceedings of the VIIIth International Conference on Multiple Criteria Decision Methods, , , Manchester, England.

[3] Eom, S. B. 1998, 'Relationships between the decision support system subspecialties and reference disciplines: an empirical investigation', *European journal of operational research*, Vol. 104, pp. 31-45.

[4] Hwang, C. L. & Masud, A. S. M. 1979, *Multiple Objective Decision Making - Methods and Applications: A State of the Art Survey*, Springer, New York.

[5] Ignizio, J. P. 1976, *Goal Programming and Extensions*, Massachusetts.

[6] Ignizio, J. P. 1981, 'The determination of a subset of efficient solutions via goal programming', *Computer and operations research*, Vol. 8, pp. 9-16.

[7] Iz, P. H. & Jelassi, M. T. 1990, 'An Interactive Group Decision Aid for Multiobjective Problems: An Empirical Assessment', *International Journal of Management Science*, Vol. 18, no. 6, pp. 245-253.

[8] Kersten, G. E. 1985, 'NEGO-group decision support system', *information & management*, Vol. 8, pp. 237-246.

[9] Klein, M. R. & Methlie, L. B. 1995, *Knowledge-based decision support systems with applications in business*, John Wiley & Sons.

[10] Lu, J., Quaddus, M. A. & Williams, R. 1999, 'A framework and prototype for intelligent multiple

objectives decision support system', in 1999 Asia Pacific Decision Sciences Institute Conference, Shanghai, China.

[11] Masud, A. S. M. & Hwang, C. L. 1981, 'Interactive Sequential Goal Programming', *Journal of the Operational Research Society,* Vol. 32, pp. 391-440.

[12] Nazareth, D. L. 1993, 'Integrating MCDM and DSS: Barriers and Counter Strategies', *Infor,* Vol. 31, no. 1, pp. 1-15.

[13] Pinson, S. & Moraitis, P. 1996, 'An intelligent distributed system for strategic decision making', *Group decision and negotiation,* Vol. 6, pp. 77-108.

[14] Poh, K. 1998, 'Knowledge-based guidance system for multi-attribute decision making', *Artificial intelligence in engineering,* Vol. 12, no. 3, pp. 315-326.

[15] Poh, K. L., Quaddus, M. A. & Chin, K. L. 1995, 'MOLP-PC: An Interactive Decision Support Environment for Multiple Objective Linear Optimization', In *OR Applications in Singapore*, ed. M. C. T. L. Goh, Operational Research Society of Singapore, , pp. 25-39.

[16] Quaddus, M. A. & Holzman, A. G. 1986, 'IMOLP: An Interactive Method for Multiple Objective Linear Programs', *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-16, no. 3, pp. 462 - 468.

[17] Steuer, R. E. 1977, 'An interactive multiple objective linear programming procedure', *TIMS Studies in the management sciences,* Vol. 6, pp. 225-239.

[18] Tecle, A. & Duckstein, L. (eds) 1992, *A procedure for selecting MCDM techniques for forest resource management.* Springer-Verlag, New York.

[19] Zionts, S. & Wallenius, J. 1976, 'An interactive programming method for solving the multiple criteria problem', *management science,* Vol. 22, no. 6, pp. 652-663.

AUSTRALIAN JOURNAL OF INTELLIGENT INFORMATION PROCESSING SYSTEMS

Volume 9, No. 1
2006

Australian Journal of Intelligent Information Processing Systems

ISSN: 1321-2133

Volume 9, No. 1
2006

CONTENTS

This is a refereed journal.
Abstracted by INSPEC.

Publication date December 2006