A Logic Based Framework for Multi-Objective Decision Making

Qing Zhou[1] and Donglei Chen[2]

[1] The Software Institute,
Zhongshan University,
Guangzhou, Guangdong 510275, P.R. China
lnszq@mail.sysu.edu.cn
[2] The Water Control Center, Guangzhou Water Supply Co.,
Guangzhou, Guangdong 510600, P.R. China
cdonglei@gmail.com

Abstract. In this paper a framework for multi-objective decision making, *FMODM*, is proposed. The framework finds the best option for our goal based upon the knowledge and the facts in our hands and hence it is quite close to the "what ... if ..." analysis often used by human beings making decision. The whole framework is described in details and the theoretical analysis is discussed comprehensively. Computability of the framework is discussed through a few theorems proven in the paper. Examples in the paper clearly illustrate the whole working of the framework.

Keywords: Multi-objective, supported degree, expected value.

1 Introduction

In this paper a framework for multi-objective decision making is proposed.

We may describe the decision-making problem that we face in life as following: Assume that we have a problem P and have a few options $c_1,...,c_n$ which are proposed as the possible options. The decision making problem is rooted in the need to find the best option from the option set $\{c_1,...,c_n\}$. It might happen that our goal may consist of a few objectives $G_1, ..., G_m$. The multi-objective decision making problem is to find a c_i in the option set which is the best option for the problem P when all the objectives $G_1, ..., G_m$ are considered. Multi-objective decision making problems appear everywhere in our real life, such as military, utility, management, finance, etc.

For the problem P we want to solve, there are usually some corresponding domain knowledge. Such knowledge can be divided into two parts: one is the group of basic rules in the area (called "knowledge set"); the other are facts we connected (called "evidence set"). Of course, we want to use such knowledge in our hands to analyze the situation on the problem. The principle here is obviously: if an option can brings us the most advantages and the least disadvantages that is the one we want. Therefore we need domain knowledge to analyze what advantages and disadvantages an option will bring to us for each option. So the more knowledge we have used, the more reliable our option is.

G. Yang (Ed.): Proceedings of the ICCEAE2012, AISC 181, pp. 961–966.
springerlink.com

In most cases, those advantages and disadvantages can be derived by deductions with an option is assumed, and are supported by evidences (i.e. they can be derived by evidences, too). So the common consequences of both the evidences in our hands and an option give supports to that option. It is also quite obvious that the more such common consequences an option has, the more possible for the option to be true. We call the number of such common consequences the "supported degree" of the option.

Most of our knowledge can be expressed by first-order languages and first-order logic has strong deduction power. So *FMODM* is built within first-order logic so that we can use our domain knowledge and the facts we gathered for the problem to make reasoning needed for calculating the supported degree.

For the decision making with multi-objectives we want the results of our decision is the best as a whole. Then we have to use our knowledge not only to analyze what advantages and disadvantages an option will bring to us for each objective but also to make a balanced consideration of the affects of all the objectives. This also needs to use our knowledge to analyze. With this our decision is easy to make. Then we need a way to express these impacts. In many cases, such impacts can be expressed by functions which can be obtained either from theories or from experience. We give an example on this situation.

Example 1**:** Suppose that we want to do some investments. We think that two objectives must be considered: "business-familiarity" (means how much we know about the business we are going to invest) and "development-affect" (means how much it affects our own company). We think that "business-familiarity" impacts the investment *70%*, "development-affect" impacts the investment *-30%*. Then once we know all the supported degrees about these two objectives from the options we have to use such impact factors to composite functions to calculate the impact of the whole problem of investment. We called them "Impact degree".

Once we have the supported degree for each objective, and the impact degree of each option, we can compose them to a value, called "expected value", for each option. The higher expected value an option has, the more benefits we can have from that option if we take it as our decision. This is the idea of this paper.

The remaining sections of this paper include: Outlined in Section 2 discusses how to calculate expected value to solve multi-objectivess decision making. Then make conclusion in section 3.

2 Basic Definitions

In this paper, we assume the logical reasoning system T containing our knowledge is fixed and we use the following notations:

1) L is the language of T;
2) The knowledge set K, a finite set of formulae of L;
3) The evidence set E, a finite set of closed formulae of L;
4) The objective set G, a finite set of formulae of L.

We assume that $K \cup E$ is consistent.

Now we define:

***Definition 1* (*Multi-Objectives Decision Making Problem*):** A multi-objectives decision making problem P is: For P, we have objectives $G_1, G_2, \ldots, G_m \in G$, where P and $G_1, G_2, \ldots, G_m$ are predicates. Each objective corresponds to real number $v_j \in [-1,1]$, and v_j is called the impact degree for G_j, where $j \in N$. Then the function $F=f(s_1 \times v_1, s_2 \times v_2, \ldots, s_m \times v_m)$ is called the expected value of P, where f: $[-1,1]^m \rightarrow [-1,1]$ is a function, which is called the "expectation function". The symbols $s_1, s_2, \ldots, s_m$ will be defined later.

***Example 2*:** The objectives in *Example 1* are G_1=business-familiarity, G_2=development-affect. Once we have the impact degrees of "business-familiarity" and "development-affect" given, we can use the expectation function $f(s_1 \times v_1, s_2 \times v_2) = s_1 \times 0.7 - s_2 \times 0.3$ to get the expected value F from the description in *Example 1*.

Given P, we usually have a few options which are supposed to be possible options of P. By decision-making problem, we mean that we have to choose an option from them. Let us call them possible options. This lead to:

***Definition 2* (*Option Set*):** A option set C is a finite set of formulae $\{C_1, C_2, \ldots, C_n\}$ of L such that for all $i \in \{1, 2, \ldots, n\}$, $K \cup E \nvdash C_i$ and $K \cup E \nvdash \neg C_i$.

***Example 3*:** In *Example 2* we consider three investment portfolios as options: investment portfolio 1 and investment portfolio 2. These portfolios consist of some businesses: investment portfolio 1 consists of business a_1, business a_2, and business a_3; investment portfolio 2 consists of business a_2 and business a_4. Then $C=\{C_1=$ *investment portfolio 1*, $C_2=$*investment portfolio 2*$\}$ is the option set, and our purpose is to find the best one in C.

To know what we will have from a option for each objective, we observe that what we will get have to have something in common with both the evidences in our hands and the objective considered. Then we have:

***Definition 3* (*Compatible Result Set*):** Suppose $C_i \in C$ and $G_j \in G$. A formula A is called a compatible result of C_i to G_j if:

- $K \cup E \vdash A$
- $K \cup \{C_i\} \vdash A$
- $K \cup \{G_j\} \vdash A$
- $K \nvdash A$

Then we use D_{ij} to denote the set of all compatible results of C_i to G_j, that is:

$$D_{ij}=\{A \mid A \text{ is a compatible result of } C_i \text{ to } G_j\}.$$

***Example 4*:** In the *Example 3* the language L has the following formulas:

1) Objectives: $G_1(t)$, which indicates business-familiarity; $G_2(t)$, which indicates development-affect, where t is a set of investment projects;
2) Options: C_1, which indicates investment portfolio 1; C_2, which indicates investment portfolio 2;
3) $Prj(x)$, which indicates investing to the business x;
4) $BfHigh(x)$, which indicates the degree of business-familiarity is high; $DAHigh(x)$, which indicates the degree of development-affect is high.

The corpus of knowledge set K consists of the following axioms:

1) $G_1(\{a_1, a_2, a_3, a_4\}) \rightarrow BfHigh(a_1) \wedge \neg BfHigh(a_2) \wedge BfHigh(a_3) \wedge BfHigh(a_4)$, which indicates that to the objective G_1, the degree of the business-familiarity is high to business a_1, and is not high to business a_2, and is high to business a_3, and is high to business a_4;
2) $G_2(\{a_1, a_2, a_3, a_4\}) \rightarrow DAHigh(a_1) \wedge \neg DAHigh(a_2) \wedge DAHigh(a_3) \wedge \neg DAHigh(a_4)$, which indicates that to the objective G_2, the degree of the development-affect is high to business a_1, and is not high to business a_2, and is high to business a_3, and is not high to business a_4;
3) $C_1 \rightarrow Prj(a_1) \wedge Prj(a_2) \wedge Prj(a_3)$, which indicates that if investment portfolio 1 chosen, it has business a_1, a_2, and a_3;
4) $C_2 \rightarrow Prj(a_2) \wedge Prj(a_4)$, which indicates that if investment portfolio 2 chosen, it has business a_2, and a_4;
5) $Prj(a_1) \rightarrow BfHigh(a_1) \wedge DAHigh(a_1)$, which indicates that business a_1 has high degree of business-familiarity and development-affect;
6) $Prj(a_2) \rightarrow BfHigh(a_2) \wedge \neg DAHigh(a_2)$, which indicates that business a_2 has high degree of business-familiarity and has not high degree of development-affect;
7) $Prj(a_3) \rightarrow \neg BfHigh(a_3) \wedge DAHigh(a_3)$, which indicates that business a_3 has not high degree of business-familiarity and has high degree of development-affect;
8) $Prj(a_4) \rightarrow \neg BfHigh(a_4) \wedge DAHigh(a_4)$, which indicates that business a_4 has not high degree of business-familiarity and development-affect.

We assume that the system has two deduction rules: 1. if $\alpha \rightarrow \beta$ and α, then β; 2. if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$.

The evidence set E consists of:

$BfHigh(a_1)$, $\neg BfHigh(a_2)$, $BfHigh(a_3)$, $BfHigh(a_4)$, $DAHigh(a_1)$, $\neg DAHigh(a_2)$, $DAHigh(a_3)$, $DAHigh(a_4)$, $BfHigh(a_2)$, $\neg BfHigh(a_3)$, $\neg BfHigh(a_4)$, $\neg DAHigh(a_4)$.

Then we have made up the construction of the system, Let us look at how can we get the compatible result set:

To get the compatible result set of C_1 to G_1, for example, we can get $\{BfHigh(a_1)$, $\neg BfHigh(a_2)$, $BfHigh(a_3)$, $BfHigh(a_4)\}$ by $K \cup \{G_1\}$, get $\{BfHigh(a_1)$, $BfHigh(a_2)$, $\neg BfHigh(a_3)$, $DAHigh(a_1)$, $\neg DAHigh(a_2)$, $DAHigh(a_3)\}$ by $K \cup \{C_1\}$, and get E by $K \cup E$. Then the compatible result set of C_1 to G_1 is $\{BfHigh(a_1)\}$. The other compatible result sets will be gotten like this.

Now as we know that although the presentation of some conclusions is not the same, but the meaning of them is the same, such as "A's father is B", has the same meaning with "A is B's son and B is man". Then our compatibility result set must merge these results. In logic, if the conditions for the establishment of two conclusions identical to that of these two conclusions are equivalent, such as "A's father is B" implies itself and "A is B's son and B is man" implies "A's father is B", then we take them as the same, and that means these two are equivalence. The equivalence relation thus defined the equivalence class to essentially the same relationship can be combined into one. Therefore, the following statement :

Set $A_1, A_2 \in D_{ij}$, if for all $E' \subseteq E$, $K \cup E' \vdash A_1$ if and only if $K \cup E' \vdash A_2$, called A_1, A_2 is equivalent in D_{ij}, denoted by $A_1 \sim A_2$.

***Definition 4* (*Equivalence Class Set*):** B_{ij} is the set of equivalence class of D_{ij} to $\sim$, which $B_{ij}=\{U:U$ is the maximum sub-set of D_{ij} which satisfy that for all $A_1, A_2 \in U, A_1 \sim A_2\}$.

***Example 5*:** In *Example 4*, we can see that the equivalence class of the compatible result set such as $\{BfHigh(a_1)\}$ is the equivalence class of $BfHigh(a_1)$.

The equivalence class set B_{ij} have been merged the repeated compatibility results in D_{ij}, then the size of B_{ij} is just the possibility of option C_i to objective G_j. In other words, it is how deep the option supports the objective, just what we called "supported degree". Therefore defined as follows:

***Definition 5* (*Supported Degree*):** $s(i, j)$ is the supported degree of C_i to G_j. And $s(i, j)$ calculated according to the following:

$$s(i, j)=|B_{ij}|/ (\sum^{n}_{i=1}|B_{ij}|)$$

***Example 6*:** So we can calculate the supported degrees of all the options as the following:

Table 1. Supported degree of all the options

	G_1	G_2
C_1	1/1	3/4
C_2	0/1	1/4

Supported degree $s(i, j)$ is the degree the option C_i supports the objective G_j. And then we can merge all the objects as a whole by the expectation function mentioned in *Definition 1*:

***Definition 6* (*Expectations*):** Given a option C_i, F_i is the expected value of C_i to the problem P, and F_i is calculated by the following:

$$F_i=f(s(i, 1)\times v_1, s(i, 2)\times v_2, \ldots, s(i, m)\times v_m)$$

Then we can get the option which has the highest expected value as the best option to the problem P.

***Example 7*:** All the expected values are calculated as following:

$$F_1=0.7\times 1-0.3\times 0.75=0.475$$

$$F_2=0.7\times 0-0.3\times 0.25=0.075$$

We can see that the option C_1 ("investment portfolio 1") is the best.

***Theorem 1* (*Computability of Expected Value*):** $(s(i, j)\times v_j) \in [-1, 1]$

From this theorem, that $s(i, j)\times v_j$ is finite, and their summation, the expected value, is finite too, and it is computable.

3 Summaries

In this paper a framework *FMODM* of decision making for multi-objectives is proposed. The framework is built within first-order logic and is completely based upon our knowledge on the decision we want to make. This makes *FMODM* is reasonable and has a wide range of applications. Competing approaches in literature mainly contains those based upon probability theory or fuzzy logic. In these approaches decisions are making only relying on static data which may or may not be available or be reliable. Computability of *FMODM* is discussed through a few theorems. The whole working of *FMODM* is illustrated via a unified example. A significant advantage of *FMODM* is that some natural words related to decision making, such as "advantage" or "disadvantage", are precisely defined and hence all the knowledge on the decision we are going to make can be well utilized so that it makes our decision more reliable.

Many works on various directions can be done from our work. Here we only mention a couple of them. In the field of decision making a lot of work has been done by using technologies, such as genetic algorithm, neural network. The work in this paper can be combined with such technologies so that knowledge can be well used in these technologies. We believe that this will expand the range of applications of computer-aided decision making and make it more reliable. Another direction can go along the direction on computational efficiency. Although it is proven "*FMODM*" is computable, but there are rooms of improving the efficiency of searching right results in a logical system.

Given the state of art on decision making with multi-objectives we believe that this is a very nice step forwards. The suggestions mentioned above can be greatly increasing the power of our proposal and improve the research in this field.

References

[1] Shen, X., Guo, Y., Chen, Q., Hu, W.: A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic. Computational Optimization and Applications 46(1), 159–188 (2008)

[2] Rollón, E., Larrosa, J.: Bucket elimination for multiobjective optimization problems. Journal of Heuristics 12(4-5), 307–328 (2006)

[3] Pappa, G.L., Freitas, A.A.: Evolving rule induction algorithms with multi-objective grammar-based genetic programming. Knowledge and Information Systems 19(3), 283–309 (2009)

[4] Zhou, Q., Peng, W.: Uncertain Reasoning and Decision Making. In: Proceedings of International Conference on Intelligent Information Processing (IIP 2004). Springer (2004)

[5] Zhou, Q., Deng, X., Jones, J.D., Zhang, K.: A diagnostic system based upon knowledge and experience. The First Supporting Degree, pp. 280–283 (2008)

[6] Datta, D., Deb, K., Fonseca, C.M.: Multi-objective Evolutionary Algorithms for Resource Allocation Problems. Class Timetabling and Land-Use Management Problems as Multi-objective Optimization Problems, pp. 403-405; NSGA-II-UCTO and NSGA-II-LUM, pp. 405-409 (2007)

[7] Cai, Z., Gong, W., Huang, Y.: A Novel Differential Evolution Algorithm Based on ε-Domination and Orthogonal Design Method for Multiobjective Optimization. Our Approach: ε-ODEMO 2007 (2007)

[8] Zhou, Q., Ju, S.: The Concept of Uncertain Reasoning and its Supporting Degree. In: Proceedings of the ISCA 9th International Conference (2000)