Rule Sets Based Bilevel Decision Model

Zheng Z.[*#], Zhang G.[*], He Q.[#], Lu J.[*], Shi Z.[#]

[*] Faculty of Information Technology, University of Technology, Sydney
PO BOX 123, Broadway, NSW 2007, Australia
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

Corresponding Author Email: jielu@it.uts.edu.au

Abstract

Bilevel decision addresses the problem in which two levels of decision makers, each tries to optimize their individual objectives under constraints, act and react in an uncooperative, sequential manner. Such a bilevel optimization structure appears naturally in many aspects of planning, management and policy making. However, bilevel decision making may involve many uncertain factors in a real world problem. Therefore it is hard to determine the objective functions and constraints of the leader and the follower when build a bilevel decision model. To deal with this issue, this study explores the use of rule sets to format a bilevel decision problem by establishing a rule sets based model. After develop a method to construct a rule sets based bilevel model of a real-world problem, an example to illustrate the construction process is presented.

Keywords: Bilevel programming, Decision making, Decision model, Rough set, Rule set.

1 Introduction

Organizational decision making often involves two levels. In general, the decision maker at the upper level will influence, control or induce the behavior of the decision maker at the lower level but not completely control his action. In addition, the lower level decision maker gains his objective under a given region, although his decision is in a subordinate position. In such a bilevel decision situation, decision maker at each level has individual payoff function, and the upper level the decision maker is at, the more important and global his decision is. Therefore, a bilevel decision model intends to reach certain goals, which reflect the upper level decision makers' aims and also consider the reaction of the lower level decision makers on the final decisions. Such a decision problem is called as a bilevel decision problem. The decision maker at the upper level is known as the leader, and at the lower level, the follower.

Bilevel decision problems have been introduced by Von Stackelberg in the context of unbalanced economic markets in the fifties of the 20th century [Stackelberg 1952]. After that moment a rapid development and intensive investigation of these problems begun both in theoretical and in applications oriented directions [Chen and Gruz 1972] [Candler and Norton 1977] [Bialas and Townsley 1982] [Bard and Falk 1982] [Bard and Moore 1992] [Bard 1998] [Dempe 2002]. Contributions to this field have been delivered by mathematicians, economists and engineers and the number of papers within this field is ever growing rapidly. This interest stems from the inherent complexity and consequent challenge of the underlying mathematics, as well as the applicability of the bilevel decision model to many real-world situations.

From its inception, bilevel decision problems have been introduced to the optimization community. Most of the efforts concentrated on theoretical or applied development for the linear or nonlinear version of the problem, such as K-Best approach [Bialas and Karwan 1984] or Kuhn-Tucker approach [Bard and Falk 1982] for solving linear bilevel programming problems, and Penalty function approach [White and Anandalingam 1993] or stability based approach [Liang and Sheng 1992] for solving nonlinear bilevel programming problems. However, bilevel decision making may involve many uncertain factors in a real world problem. Therefore it is hard to determine the objective functions and constraints when build a bilevel decision model. In addition, even if all the functions are linear, the resultant model may be difficult to be solved by the methods of optimization [Bard 1998]. To handle the two issues, it therefore needs to explore establishing a bilevel model by using uncertain information processing techniques.

Our previous work presented a new definition of solution and related theorem for linear bilevel programming, thus solved a fundamental deficiency of existing linear bilevel programming theory [Shi et al 2005]. We also developed an extended Kuhn-Tucker approach [Shi et al 2005a] and an extended *K*th-best approach [Shi et al 2005b] for solving linear bilevel decision problems. As a new exploration to model and solve a bilevel decision problem, this paper first formulates a bilevel decision problem using decision rule sets. It then applies the methods of rough set to reduce the models. With the methods of value reduction in rough set theory, simpler decision rule sets are extracted from decision rule sets (decision tables) to represent the evaluation methods of the objectives or the constraints. Besides, attribute importance degree based rule trees are used to solve uncertain problems and get the final decision. The structures can be extended beyond two levels with the realization that attending behavioral and operational relations become much more difficult to conceptualize and describe. The paper is divided into five sections. Section 2 introduces the preliminaries of this paper, including some notions about decision tables and decision rules. Section 3 proposes the model of the rule sets based bilevel decision problem, and then the algorithm of modelling. In Section 4, an example is presented. The last section includes the conclusion and future work.

2 Preliminary

For the convenience of description, we introduce some basic notions of decision tables and decision rules. Besides, we also develop some related definitions that will be useful in this paper.

2.1 Decision Tables

A decision table is commonly viewed as a functional description, which maps inputs (conditions) to outputs (actions) without necessarily specifying the manner in which the mapping is to be implemented [Lew and Tamanaha 1976]. The formal definition is as follows.

Definition 2.1[Wang 2001] **(Decision tables):** A decision table is defined as

$$S=<U, R, V, f>,$$

where U is a finite set of objects; $R=C\cup D$ is a finite set of attributes, C is the condition attribute set and D is the decision attribute set; set of its values V_a is associated for every attribute $a\in R$, and $V=\cup_{r\in R}V_r$; and each attribute has a determine function f: $U\times R\rightarrow V$, and f determines the attribute value of each object x.

A decision table is as a special and important knowledge expression system. It shows that, when some conditions are satisfied, decisions, actions, operations or controls can be made. Decision attributes in a decision table can be unique or not. In the latter case, the decision table can be converted to one with unique decision attribute [Wang 2001]. So, we suppose that there is only one decision attribute in decision tables in this paper.

2.2 Decision Rules

Definition 2.2[Wang 2001] **(Decision rules):** Let $S = <U, R, V, f>$ be a decision table, and $B\subseteq C$. Then a decision rule dr is generated from B and D with the form

$$dr: \wedge\{(a, v_a)\}\Rightarrow(d, v_d),$$

where $a\in B$, $v_a\in V_a$, $d\in D$, $v_d\in V_d$, and V_a, V_d is defined by Def. 2.1; $\wedge\{(a, v_a)\}$ is called as the precondition of a decision rule (denoted as Con_{dr}) and (a, v_a) is called as an element in the precondition; (d, v_d) is called as the decision of a decision rule (denoted as Des_{dr}).

It is obvious that objects in decision tables can be expressed by decision rules.

In order to describe the rule sets based bilevel decision model clearly, we present some notions related with decision rules as follows.

Definition 2.3 (Father decision rules): Decision rule dr_1 is said to be the father rule of decision rule dr_2, if each element in Con_{dr1} is also an element in Con_{dr2} and there is at least one element in Con_{dr2} that is not an element in Con_{dr1}. Here, dr_2 is said to be the son rule of dr_1.

Definition 2.4 (Objects which are consistent with a decision rule): A object o is said to be consistent with decision rule dr: $\wedge\{(a, v_a)\}\Rightarrow(d, v_d)$ $(a\in B, d\in D)$, if for $\forall a\in(B\cup D)$, $o_a=v_a$ is satisfied, where o_a is the value of o on attribute a. Given a decision table S, the set of all objects in S that are consistent with decision rule dr is denoted as $[dr]_S$.

Definition 2.5 (Objects which are conflict with a decision rule): A object o is said to be conflict with decision rule dr: $\wedge\{(a, v_a)\}\Rightarrow(d, v_d)$ $(a\in B, d\in D)$, if for $\forall a\in B$, we have $o_a=v_a$, and $o_d\neq v_d$. Given a decision table S, the set of all objects in S that are conflict with decision rule dr is denoted as $[\not{dr}]_S$.

Definition 2.6 (Rules which are consistent with a decision table): A decision rule dr is said to be consistent with a decision table S, if there isn't any object in S that is conflict with dr.

Definition 2.7 (Rule inclusion): Decision rule dr_1 is said to be including decision rule dr_2, if all objects which are consistent with dr_2 are also consistent with dr_1, denoted as Incl(dr_1, dr_2). In this case, if the number of the elements in dr_1's precondition is the same as that in dr_2's precondition, then dr_1 is said to be equal to dr_2.

Definition 2.8 (Rule conflict): Decision rule dr_1 is said to be conflict with decision rule dr_2, if all objects satisfied dr_2 are conflict with dr_1, which is denoted as Conf(dr_1, dr_2). In this case, if the number of the elements in dr_1's precondition is the same as that in dr_2's precondition, then dr_1 is said to be completely conflict with decision rule dr_2, else dr_1 is said to be partly conflict with dr_2.

Definition 2.9 (Rule length): Rule length is the number of elements in the rule's precondition.

Decision rule set RS is the set of decision rules. It can be divided into the following two categories (Def 2.10 and Def. 2.11) according to whether there are conflicts among its rules.

Definition 2.10 (Consistent decision rule sets): A decision rule set RS is said to be consistent, if there isn't any rule in the rule set conflicting with other rules in the rule set, that is to say, $\forall$ $dr_1, dr_2\in RS$ $(\neg$ Conf$(dr_1, dr_2))$.

Definition 2.11 (Inconsistent decision rule sets): An decision rule set RS is said to be inconsistent, if there is some rule in the rule set conflicting with at least one another rule in the rule set, that is to say, $\exists rule_1 \in RS$ ($\exists rule_2 \in RS$ (Conf($rule_1$, $rule_2$))).

Definition 2.12 (Simplest decision rule sets): Suppose *dr* is a random decision rule in a consistent decision rule set *RS*, if *dr* is replaced by one of its father rules *fdr* and the resultant decision rule set is still consistent, then *RS* is said to be a redundant decision rule set, otherwise, it is said to be a simplest decision rule set.

3 Rule Sets Based Bilevel Decision Problem Modelling

When solving a bilevel decision problem, which objective functions and constraints related are expressed by linear or nonlinear functions, optimization approaches can be used. However, some real-world problems can't be easily formulated or approximated as linear or nonlinear programs. To handle the issue, new models for bilevel decision problems are needed.

A decision table can be used to lay out in a tabular form all possible situations where a decision may encounter and to specify which action to take in each of these situations. They can be used in projects to clarify complex decision making situations. Decision tables are commonly thought to be restricted in applicability to procedures involving sequencing of tests, nested-IFs, or CASE statements. In fact, a decision table can implement any computable function. It was observed that any Turing Machine program can be "emulated" by a decision table by letting each Turing Machine instruction of the form (input, state) + (output, tape movement, state) be represented by a decision rule (or an object in a decision table) where (input, state) are conditions and (output, tape movement, state) are actions. From a more practical point of view, it can also be shown that all computer program flowcharts can be emulated by decision tables [Lew and Tamanaha 1976].

Therefore, in theory, after emulating all possible situations in a domain, constraints of a decision problem can be transformed to a decision table, named as a constraint decision table. In a similar way, objective functions can also be transformed to a decision table, named as objective decision table. That is to say, a bilevel decision problem can be transformed into a set of decision tables, where decision variables are represented by the objects in these decision tables.

Rule sets are more general knowledge generated from decision table and they had stronger knowledge expressing ability than decision table. Rule sets overcome the following disadvantages of decision tables:

1) For complex situations, decision tables may become extremely large;

2) The objects in decision tables lack of adaptability. They can't adapt any new situations and one object can only record a situation.

So, we use rule sets to describe the objectives and constraints. The bilevel decision problem, which objectives and constraints of both leader and follower are described by rule sets, is called as a rule sets based bilevel decision model. And the bilevel decision model based on decision tables is a special case of rule sets based decision model.

3.1 Decision Rule Set Function

To present the model of rule sets based bilevel decision model, the definition of decision rule set function is needed.

Given a decision table S=<U, R, V, f >, where $R=C\cup D$ and $D=\{d\}$. Suppose x and y are two variables, where $x\in X$ and $X=V_{a1}\times\ldots\times V_{am}$, $y\in Y$ and $Y=V_d$. V_r is the set of attribute r's values and $a_i\in C$, i=1 to m and m is the number of condition attributes. *RS* is a decision rule set generated from *S*.

Definition 3.1 (Decision rule set function): A decision rule set function *rs* from *X* to *Y* is a subset of the cartesian product $X \times Y$, such that for each *x* in *X*, there is a unique *y* in *Y* generated with *RS* such that the ordered pair (x, y) is in *rs*. *RS* is called as the decision rule set related with the function, *x* is called as the condition variable, *y* is called as the decision variable, *X* is the definitional domain and *Y* is the value domain.

Calculating the value of a decision rule set function is to make decisions for undecided objects with decision rule sets, where undecided objects are objects without decision values. In order to present the method of calculating the value of a decision rule set function, we first introduce a definition.

Definition 3.2 (Undecided objects matching a decision rule): An undecided object *o* is said to be matching a decision rule *dr*: $\wedge\{(a, v_a)\}\Rightarrow(d, v_d)$, where $a\in B$, $d\in D$, if for each $a\in B$, $o_a=v_a$ is satisfied, where o_a is object *o*'s value on attribute *a*.

Given a decision rule set *RS*, all rules in *RS* that is matched by object *o* is denoted as MR_{RS}^{o}.

With the definitions, a brief method of calculating the result of a decision rule set function is showed as follows:

Step 1: Calculate MR_{RS}^{o};

Step 2: Select a decision rule *dr* from MR_{RS}^{o}, where $dr:\wedge\{(a, v_a)\}\Rightarrow(d, v_d)$;

Step 3: The value of $rs(o)$ is set to be v_d, that is, $rs(o)=v_d$.

Complete

It is obvious that there may be more than one rule in MR_{RS}^{o}. In this case, when decision values of the rules in MR_{RS}^{o} are different, the result could be various according to above method, which is called as the uncertainty in a decision rule set function. Methods of selecting the final rule from MR_{RS}^{o} are very important, and they are said to be the uncertainty solution methods.

The elimination of uncertainty is a process of selection. We can select a rule rightly only when some information is known. In other words, we are said to be informed only when we can select rightly and definitely. In this paper, we present a rule tree based model to deal with the uncertainty in Section 3.2.

3.2 Rule Trees

Rule tree is a compact and efficient structure expressing a rule set. We first introduce the definition of rule tree, which is developed in our previous work [Zheng and Wang 2004]. Based on the definition of rule tree, we improve the rule tree structure with two constraints.

Definition 3.3 (Attribute importance degree based rule tree): Attribute importance degree based rule tree is a rule tree, and it satisfies the following two conditions:

1) The attribute expressed by the upper level is more important than that expressed by any lower level;

2) Among the branches with the same start node, the value represented by the left branch is more (or better) than the value represented by any right branch. And every possible value is more (better) than the value "*".

Definition 3.4 (Comparison of rules): Rule dr_1: $\wedge\{(a_i, v_{a1i})\} \Rightarrow (d_1, v_{d1})$ is better than rule dr_2: $\wedge\{(a_i, v_{a2i})\} \Rightarrow (d_2, v_{d2})$, if v_{a1k} is better than v_{a2k} or the value of a_k is deleted from rule dr_2, where attribute a_i is more important than a_{i+1}, and for each $j<k$, $v_{a1j}=v_{a2j}$.

Theorem 3.1: The rule expressed by the lefter branch in an attribute importance degree based rule tree is better than the rule expressed by the righter branch.

It is obvious that the theorem holds from Def. 3.4.

Theorem 3.2: After transformed to an attribute importance degree based rule tree, the rules in a rule set are total order, that is to say, every two rules can be compared.

It is obvious that the theorem holds from Def. 3.4 and Theorem 3.1.

3.3 Rule Sets Based Bilevel Decision Model

In the following, the mathematical model of rule sets based bilevel decision model is presented. Here, we suppose there are one leader and one follower. Besides, we suppose that, if x is the undecided object of the leader and y is the undecided object of the follower, then $x \oplus y$ is the combined undecided object of the leader and the follower together.

Definition 3.5 (Model of rule sets based bilevel decision):

$$\min_{x} \ f_L(x \oplus y)$$

$$\text{s.t.} \ g_L(x \oplus y) \geq 0$$

$$\min_{y} \ f_F(x \oplus y)$$

$$\text{s.t.} \ g_F(x \oplus y) \geq 0, \tag{3.1}$$

where x and y are undecided objects of the leader and the follower respectively. f_L and g_L are the objective decision rule set function and constraint decision rule set function of the leader respectively, f_F and g_F are the objective decision rule set function and constraint decision rule set function of the follower respectively. F_L, G_L, F_F and G_F are the corresponding decision rule sets of above decision rule set functions respectively.

3.4 Modelling Algorithm of Rule Sets Based Bilevel Decision Model

In the following, we present the modelling algorithm of rule sets based bilevel decision model.

Algorithm 3.1 (Modelling Algorithm of Rule Sets Based Bilevel Decision Model)

Input: A bilevel decision problem with its objectives and constraints of both the leader and the follower;

Output: A rule sets based bilevel decision model;

Step 1: Transform the problem with decision rule sets;

Step 2: Preprocess F_L, such as delete reduplicate rules from the rule sets, eliminate noisy and etc.;

Step 3: If F_L need to be reduced,

then using reduction algorithm to reduce F_L;

Step 4: Preprocess G_L, such as delete reduplicate rules from the rule sets, eliminate noisy and etc.;

Step 5: If G_L need to be reduced,

then using reduction algorithm to reduce G_L;

Step 6: Preprocess F_F, such as delete reduplicate rules from the rule sets, eliminate noisy and etc.;

Step 7: If F_F need to be reduced,

then using reduction algorithm to reduce F_F;

Step 8: Preprocess G_F, such as delete reduplicate rules from the rule sets, eliminate noisy and etc.;

Step 9: If G_F need to be reduced,

then using reduction algorithm to reduce G_F;

Complete

Step 1 is the key step of the modeling process. The users can complete the step by lay out all possible situations, that is, transform the problem to decision tables. When the users know the general knowledge (rules) under the problem, they can directly transform the problem to some simpler decision rule sets. In general, the realization of the step depends on the characters of the problem and the users' knowledge related with the problem.

In Step 2, Step 4 and Step 6, the four rule sets are preprocessed. The process is very important, because incomplete, noisy and inconsistency are the common characters of huge and real data. So, we should use some techniques to eliminate these problems in data before modeling. In [Han and Kamber 2001], the issue is discussed in detail.

In Step 5, Step 7, and Step 9 of Alg. 3.1, rule set is reduced by some reduction algorithm. To reduce a decision rule set or extract decision rules from a decision table, the methods based on rough set theory are popular and efficient. Many rough set based decision rule extraction algorithms, named as value reduction algorithms, are developed in rough set theory [Pawlak 1991] [Hu and Cercone 1995] [Mollestad and Skowron 1996] [Wang 2001] [Zheng and Wang 2004]. And the algorithms made successful applications in many fields [Kiak 2001] [Pawlak and Slowinski 1986] [Kiak 2001] [Carlin et al 1998]. Besides, there are some rough set based systems, such as ROSETTA [ROSETTA], RIDAS [Wang et al 2002], RSES [Jan et al 2002] and so on, can be used to extract decision rule sets from decision tables. So, we use rough set theory based methods to reduce the rule sets based models in this paper.

Based on rough set theory, various value reduction algorithms can be developed. Value reduction is a process to find a subset of values in decision rule set which satisfies that removing any value in this subset will definitely cause new inconsistency. There are many value reduction algorithms [Wang 2001] [Hu and Cercone 1995] [Mollestad and Skowron 1996] [Zheng and Wang 2004]. A simplest decision rule set (Def. 2.12) can be extracted from a rule set or decision table with the reduction algorithms of rough sets.

In the following section, we use an example to illustrate the modelling process.

4 Example

Suppose there is a factory with two levels in its staff management. The upper level is the factory executive committee and the lower is a workshop management committee. Now, the factory wants to recruit new workers. The factory executive committee should consider the overall objectives, and the workshop management committee considers its own needs, so the objectives for the two levels may be different. The executive committee of factory could ask the workshop to calculate and submit an optimal production plan as though it were operating in isolation. Once the plans are submitted, they are modified with the overall objective of the factory in mind. An output plan ultimately emerges that is optimal for the factory as a whole.

When decide whether a person could be recruited, the factory executive committee considers the following two factors, which are team spirit and organizational ability of the person; and the workshop management committee considers two factors, which are age and eyesight of the person. Suppose the condition attributes in ascending order according to the importance degree are "Team Spirit", "Organizational Ability", "Age", "Eyesight".

The two committees can't express the conditions of the workers they want recruit to linear or nonlinear functions. But they have a base recorded the worker's information having been recruited. So, we can transform the base to two decision tables (Table 4.1, 4.2), which are the objective rule sets of the leader and the follower. The objects of the decision tables represent workers. The condition attributes of the decision tables are the factors; the decision attributes of the two decision tables are both the accept grade of the worker represented by the condition attribute values. The constraints of the two committees are expressed by simple rule sets (Equation 4.1, 4.2), which define the constraint region.

Then, we use Alg. 3.1 to transform the problem to rule sets based bilevel model.

Alg. 3.1-Step 1: Transform the problem with decision rule sets. Table 4.1 represents the objective rule set of the leader, Table 4.2 represents the objective rule set of the follower, Equation 4.1 represents the constraint rule set of the leader and Equation 4.2 represents the constraint rule set of the follower.

Table 4.1 Objective rule set of the leader

Team Spirit	Organizational Ability	Age	Eyesight	Accept Grade
Poor	Middle	Middle	Middle	2
Good	Middle	Middle	Middle	1
Good	Fine	Old	Middle	1
Middle	Poor	Young	Poor	3
Poor	Poor	Middle	Middle	3
Middle	Poor	Old	Poor	3

Good	Middle	Middle	Good	1
Good	Fine	Middle	Middle	1
Middle	Fine	Old	Poor	2
Good	Fine	Old	Good	1
Good	Poor	Old	Good	3
Good	Fine	Young	Good	1
Good	Poor	Young	Middle	3

The constraint rule set of the leader:

G_L= { (Team Spirit, Good) ⇒ (*pc*, 1)

(Team Spirit, Middle) ⇒ (*pc*, 1)

} (4.1)

Table 4.2 Objective decision table of the follower

Organizational Ability	Age	Eyesight	Accept Grade
Fine	Young	Poor	2
Poor	Old	Good	2
Fine	Young	Good	1
Fine	Old	Middle	1
Poor	Young	Middle	3
Middle	Middle	Poor	2
Poor	Middle	Poor	3
Poor	Old	Poor	3
Fine	Old	Good	1
Poor	Young	Good	2
Middle	Young	Middle	2
Poor	Middle	Good	2
Fine	Old	Good	1
Middle	Middle	Good	2
Fine	Middle	Poor	2

The constraint rule set of the follower:

G_F= {(Eyesight, Poor) ⇒ (*pc*, 0)} (4.2)

Because the scale of the data is very small, the preprocess steps(Step 2, Step 4, Step 6 and Step 8) are not needed. Besides, the constraint rule sets of the leader and the follower are very brief, so the reduction steps of G_L and G_F (Step 5 and Step 9) are not needed.

In the constraint rule sets, we suppose that, if the decision of a rule is (*pc*, 0), any undecided objects consistent with the rule are not in the constraint region; if the decision value of a rule is (*pc*, 1), any undecided objects consistent with the rule are in the constraint region. We can also use some other formats of the constraint rule to express the constraint region.

Alg. 3.1-Step 3 and Step 7: Reduce the objective rule sets of the leader and the follower.

After reducing the decision tables based on rough set theory, we can get reduced objective rule sets of the leader and the follower (4.3, 4.4). Here, we use the decision matrices based value reduction algorithm [Ziarko et al 1996] in RIDAS system [Wang et al 2002].

The reduced objective rule set of the leader:

F_L={(Team Spirit, Poor)∧(Organizational Ability, Middle) ⇒ (Accept Grade, 2)

(Team Spirit, Good) ∧(Age, Middle) ⇒ (Accept Grade, 1)

(Team Spirit, Good)∧(Organizational Ability, Fine) ⇒ (Accept Grade, 1)

(Organizational Ability, Poor) ⇒ (Accept Grade, 3)

(Team Spirit, Middle)∧(Organizational Ability, Fine) ⇒ (Accept Grade, 2)

} (4.3)

The reduced objective rule set of the follower:

F_F={(Organizational Ability, Fine) ∧(Eyesight, Poor) ⇒ (Accept Grade, 2)

(Organizational Ability, Poor) ∧(Eyesight, Good) ⇒ (Accept Grade, 2)

(Organizational Ability, Fine) ∧(Eyesight, Good) ⇒ (Accept Grade, 1)

(Organizational Ability, Fine)∧(Age, Old) ⇒ (Accept Grade, 1)

(Organizational Ability, Poor) ∧(Eyesight, Middle) ⇒ (Accept Grade, 3)

(Organizational Ability, Middle) ⇒ (Accept Grade, 2)

(Organizational Ability, Poor) ∧(Eyesight, Poor) ⇒ (Accept Grade, 3)

} (4.4)

With above steps, we get the rule sets based bilevel decision model of the problem, that is

$$\begin{aligned} &\min_{x} \ f_L(x \oplus y) \\ &\quad \text{s.t. } g_L(x \oplus y) \geq 0 \\ &\quad \min_{y} \ f_F(x \oplus y) \qquad (4.5) \\ &\quad\quad \text{s.t. } g_F(x \oplus y) \geq 0, \end{aligned}$$

where f_L, f_F, g_L, g_F are the corresponding decision rule set functions of F_L, F_F, G_L, G_F.

5 Conclusion and Future Work

In this paper, we explore the use of rule set approach to format a bilevel decision problem by establishing a rule sets based model. We have seen that the common features of bilevel decision problems are:

a) Interactive decision making units exist within a predominantly hierarchical structure;

b) The lower level executes its policies after, and in view of, decisions made at the upper level;

c) Each unit independently maximizes net benefits (minimizes net costs), but is affected by the actions of other units through externalities;

d) Extramural effects enter a decision maker's problem through his objective function and feasible strategy set.

Rule sets based bilevel decision problems incorporate above features. From these features, it is obvious that to solve a rule sets based bilevel problem should be based on the solving method of rule sets based multiple objectives decision problems. Besides, we can divide the algorithms solving rule sets based decision problems into three categories, that is, forward algorithms, reverse algorithms and mixed algorithms. These issues would be discussed in our future work.

6 Acknowledgments

The work presented in this paper was supported by Australian Research Council (ARC) under discovery grants DP0557154 and DP0559213, and University of Technology, Sydney (UTS) under grant

7 References

Bard, J.F. (1998), *Practical Bilevel Optimization: Algorithms and Applications*, Kluwer Academic Publishers, USA.

Bard, J.F. and Falk, J.E. (1982), An Explicit Solution to the Multi-Level Programming Problem, *Computers & Operations Research* **9**, 77-100.

Bard, J.F. and Moore, J.T. (1992), An Algorithm for the Discrete Bilevel Programming Problem, *Naval Research Logistics* **39,** 419-435.

Bialas, W. F. and Karwan, M. H. (1982), On Two-Level Optimization, *IEEE Trans Automatic Control* **AC-26,** 211-214.

Bialas, W. and Karwan, M. (1984), Two-Level Linear Programming, *Management Science* **30,** 1004–1020.

Candler, W. and Norton, R. (1977), *Multilevel Programming and Development Policy*, World Band Staff Work No. 258, IBRD, Washington, D.C..

Carlin, U. S., Komorowski, J. and Ohrn, A. (1998), Rough set analysis of patients with suspected of acute appendicitis, *Proc. IPMU'98*, Paris, France, 1528–1533.

Chen, C.I. and Gruz, J.B. (1972), Stackelberg Solution for Two Person Games with Biased Informtaion Patterns, *IEEE Trans. On Automatic Control* **AC-17,** 791-798.

Dempe, S. (2002), *Foundations of Bilevel Programming*, Kluwere Academic Publishers.

Han, J. and Kamber, M. (2001), *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers.

Hu, X.H. and Cercone, N. (1995), Learning in relational database: a rough set approach, *Computational Intelligence*, **11**, 323-338.

Kiak, A. (2001), Rough Set Theory: A Data Mining Tool For Semiconductor Manufacturing, *IEEE Transaction on Electronics Packaging Manufacturing*, **24**, 44-50.

Jan, G. B., Marcin, S. S. and Jakub, W. (2002), A New Version of Rough Set Exploration System, *Rough Sets and Current Trends in Computing,* Publisher? 397-404.

Lew, A. and Tamanaha, D. (1976), Decision table programming and reliability, *Proc. 2nd Intl. Conf. Software Engineering,* San Francisco, 345-349.

Liang, L. and Sheng S.H. (1992), The Stability Analysis of Bilevel Decision and Its Application, *Decision And Decision Support System* **2**, 63-70.

Mollestad, T. and Skowron, A. (1996), A rough set framework for data mining of propositional default rules. *Proc. Foundations of Intelligent Systems of the 9th International Symposium,* 448-457, Springer-Verlag.

Pawlak, Z. (1991), *Rough sets Theoretical Aspects of Reasoning about Data*, Boston, Kluwer Academic Publishers.

Pawlak, Z. and Slowinski, K. (1986), Rough Classification of Patients After Highly Selective Vagotomy Duodenal Ulcer, *International Journal of Man-Machine Studies* **24**, 413-433.

Pooch, U.W. (1974), Translation of decision tables, *ACM Computing Survey* **6**, 125-151.

ROSETTA: The ROSETTA Homepage, http://www.rosettaproject.org/.

Schlimmer, J. C. and Fisher, D A. (1986), Case study of incremental concept induction, *Proceedingsof the Fifth National Conf. on Artificial Intelligence*, **1,** 496-501.

Shan, N., Ziarko, W., Hamilton, H. J., and Cercone, N. (1995), Using rough sets as tools for knowledge discovery, *Proc. 1st Int. Conf. Knowledge Discovery Data Mining*, Menlo Park, CA, 263–268.

Shi, C., Lu, J. and Zhang, G. (2005a), An extended Kuhn-Tucker approach for linear bilevel programming, *Applied Mathematics and Computation* **162,** 51-63.

Shi, C., Lu, J. and Zhang, G. (2005b), An extended *K*th-Best approach for linear bilevel programming, *Applied Mathematics and Computation* **164,** 843-855.

Shi, C., Zhang, G. and Lu, J. (2005), On the definition of linear bilevel programming solution, *Applied Mathematics and Computation* **160,** 169-176.

Stackelberg, H. V.(1952), *The Theory of Market Economy*, Oxford, Oxford University Press.

Skowron, A. and Polkowski, L. (1998), *Rough Sets in Knowledge Discovery*, Physica Verlag, Heidelberg.

Wang, G.Y. (2001), *Rough set theory and knowledge acquisition*, Press of Xi'an Jiaotong University (In Chinese).

Wang, G. Y., Zheng, Z. and Zhang, Y. (2002), RIDAS-A Rough Set Based Intelligent Data Analysis System, *Proceedings of the First International Conference on Machine Learning and Cybernetics*, 646~649.

White, D. and Anandalingam, G. (1993), A Penalty Function Approach For Solving Bi-Level Linear Programs, *Journal of Global Optimization* **3,** 397–419.

Zheng, Z. and Wang, G.Y. (2004), RRIA:A Rough Set and Rule Tree Based Incremental knowledge Acquisition Algorithm, *Fundamenta Informaticae* **59,** 299-313.

Ziarko, W., Cercone, N. and Hu, X. (1996): Rule Discovery from Databases with Decision Matrices, *9th Int. Symposium on Foundation of Intelligent Systems,* 653-662.