An Algorithm for Solving Rule-Sets Based Bilevel Decision Problems

GUANG-QUAN ZHANG
The Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia

Zheng Zheng
Beijing University of Aeronautics and Astronautics, Beijing, China

JIE LU
The Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia

QING HE
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

Address correspondence to Jie Lu, The Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia (telephone: +61-2-95141838, e-mail: 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 certain constraints, and to act and react in an uncooperative and sequential manner. Given the difficulty of formulating a bilevel decision problem by mathematical functions, a rule-sets based bilevel decision model was proposed. This paper presents an algorithm to solve a rule-sets based bilevel decision problem. A case based example is given to illustrate the functions of the proposed algorithm. Finally, a set of experiments is analyzed to further show the functions and the effectiveness of the proposed algorithm.

Key words: Decision making model, rule sets, bilevel decision making, optimization algorithm.

1. INTRODUCTION

A bilevel decision problem can be viewed as a static version of the non-cooperative, two-player (decision maker) game (Stackelberg 1952). The decision maker at the upper level is termed the leader, and at the lower level, the follower. In a bilevel decision problem, the control for decision factors is divided amongst the decision makers who seek to optimize their individual objective functions (Aiyoshi and Shimizu 1981). Perfect information is assumed so that both the leader and the follower know the objectives and feasible choices available to the other. The leader attempts to optimize his/her objective function but he/she must anticipate all possible responses of the follower (Lai 1996). The follower observes the leader's decision and then responds to it in a way that is personally optimal. For example, we consider a logistic companies decision making on how to use commission as a means for its distributors to improve product sale volume. The company, as the leader, attempts to maximize its benefit of product sale through offering a highly competitive commission to its distributors. For each of the possible commission strategies, the distributors, as the follower, will respond on product sale volume which is based on the maximized benefit obtained through the product sale. Therefore, in such a bilevel decision problem described by a bilevel programming (BLP) model, a subset of the decision variables (such as 'commission' in the example) is constrained to be a solution of a given optimization problem parameterized by the remaining variables (such as 'sale volume') (Anandalingam and Friesz 1992; Bard and Falk 1982; Bard and Moore 1992). In mathematical terms, a BLP problem consists of finding a solution for the upper level problem:

$$\max_{y_0} \quad F\left(y_0, y_1, \cdots, y_m\right)$$

$$\text{subject to:} \quad AY < 0,$$

where y_i $(i = 1, 2, \cdots, m)$, for each value of y_0, is the solution of the lower level problem:

$$\max_{y_i} \quad f\left(y_0, y_1, \cdots, y_m\right)$$

$$\text{subject to}: \quad B_i Y < 0,$$

where $Y = (y_0, y_1, \cdots, y_m)^T$ and A, B_i $(i = 1, 2, \cdots, m)$ are matrixes.

The majority of BLP research has centered on the linear version of the problem. Reference (Candler and Townsley 1982) first discussed a linear BLP problem with no upper level constraints and with unique lower level solutions. Later, references (Bard 1984; Bialas and Karwan 1984) proved this result under the assumption that the constraint region is bounded. Following these results, there have been nearly two dozen approaches and algorithms proposed for solving linear BLP problems, for example the *K*th-Best approach (Candler and Townsley 1982; Bialas and Karwan 1984), and the Kuhn-Tucker approach (Bard and Falk 1982; Bialas and Karwan 1982; Hansen, Jaumard, and Savard 1992). There have also been some intelligent approaches to solving linear bilevel programming problems (Lan et al. 2007; Calvete, Galé, and Mateo. 2008), as well as Penalty function approach (Aiyoshi and Shimizu 1981; White and Anandalingam 1993), stability based approach (Liang and Sheng 1992), and a globally convergent approach for solving nonlinear bilevel programming problems (Wang et al. 2007). Mathematicians, economists, engineers and other researchers and developers have delivered contributions to this field.

BLP is the most suitable way to model a bilevel decision problem by assuming that: (1) both the leader and the follower have perfect information about their objectives and constraints; and (2) these objectives and constraints can be written into mathematics functions. However, in real situations, it is often very hard to describe these objectives and constraints by mathematical functions including the determination of their parameters. Let us consider the logistic company example mentioned above. The company can only estimate various feasible choices taken and various costs spent by its distributors. Therefore, in establishing a BLP model for the ‘commission’ problem, we are hard-pressed to arrive at a formula for the objective functions and constraint functions (including their function types and parameters) of the leader and the follower. Some researchers such as (Lai 1996; Sakawa, Nishizaki, and Uemura 2000a; Sakawa, Nishizaki, and Uemura 2000b; Sakawa and Yauchi 2000; Sakawa and Nishizak 2001a; Sakawa and Nishizak 2001b, 2002; Shih, Lai, and Lee. 1996; Zhang and Lu 2005, 2006; Zhang, Lu, Dillon 2007a, 2007b) have developed fuzzy BLP approaches to handle the difficulty in determining the parameters in the objective and constraint functions of a BLP. However, they still assume that all these objective and constraint functions can be established and only their parameters are uncertain. Obviously, this cannot solve the problem where these mathematical functions can not be established.

We have recently observed that in many bilevel decision problems, the leader’s attempts to optimize his/her objectives and all the possible responses from the follower can be described by a number of rules (Zheng et al. 2009). Therefore, when a bilevel problem cannot be formulated by a classical BLP model, we can explore the use of rule sets to describe its objective functions and constrains. We thus proposed a rule-sets based bilevel decision (RSBLD) model. If a bilevel decision problem is modeled by a RSBLD model, we call it a RSBLD problem. We have also developed a modeling approach to establish a RSBLD model (Zheng et al. 2009).

This study considers the challenge of developing a rule-sets based bilevel decision approach for solving a RSBLD problem. We propose a transformation based solution algorithm for the RSBLD problems. The main idea of the algorithm is to first transform a RSBLD model to a single level decision model which has the same optimal solution as the original bilevel one, and then obtain the optimal solution by solving the single level decision model.

The paper is organized as follows. After this introduction, Section 2 introduces the concepts and notions of information tables and rule sets, which are the preliminaries in this study. Section 3 reviews our previous work including a RSBLD model and its modeling algorithm. How to transform a RSBLD problem into a single level one is discussed in Section 4. Section 5 presents a transformation based algorithm using the proposed transformation theories. A case based example is then shown in Section 6 for illustrating of the proposed algorithm. In Section 7, a set of experiment results are analyzed to show the effectiveness of the proposed algorithm. Finally, the conclusion and proposals for future work are given in Section 8.

2. PRELIMINARIES

For the convenience of describing proposed models and algorithms, we will first introduce some basic notions regarding information tables, formulas, rules, decision rule set functions and rule trees. In addition, we will give some related definitions and theorems which will be used in the following sections.

2.1. Information Tables

To present the definition of a rule, we first describe information table and decision table techniques. In general, an *information table* is a knowledge expressing system which can be used to represent and process knowledge in machine learning, data mining and other related fields. It provides a convenient way to describe a finite set of objects called the universe by a finite set of attributes (Pawlak 1991).

Definition 1 (Information table) (Pawlak 1991): An information table can be formulated as a tuple:

$$S=(U, At, L, \{V_a \mid a\in At\}, \{I_a \mid a\in At\}),$$

where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, L is a language defined using attributes in At, V_a is a nonempty set of values for $a\in At$, I_a: $U\rightarrow V_a$ is an information function. Each information function I_a is a total function that maps an object of U to exactly one value in V_a.

A decision table is a special case of an information table. It 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.

Definition 2 (Decision table) (Pawlak 1991): A decision table is an information table for which the attributes in A are further classified into disjoint sets of condition attributes C and decision attributes D, i.e. $At=C\cup D$, $C\cap D=\Phi$.

Decision attributes in a decision table can be unique or not. In the later case, the decision table can be converted to one with unique decision attributes (Wang 2001). Therefore, in this paper, we assume that there is only one decision attribute in a decision table.

2.2. Formulas and Rules

Usually, the knowledge implicated in information tables is expressed by rules. As formulas are the components of rules we first introduce the definition of formulas.

Definition 3 (Formulas) (Yao and Yao 2002): In the language L of an information table, an atomic formula is given by (a, v), where $a \in At$ and $v \in V_a$. If ϕ and φ are formulas, then so are $\neg\phi$, $\phi \wedge \varphi$, and $\phi \vee \varphi$.

Here, "(a, v)" is a term where a is an attribute and v one of its values. The term covers objects of the information table when the attribute a in At has value v. The semantics of the language L can be defined in Tarski's style (Tarski 1956) through the notions of a model and satisfiability. The model is an information table S, which provides interpretation for symbols and formulas of L.

Definition 4 (Satisfiability of formulas) (Yao and Yao 2002): The satisfiability of a formula ϕ by an object x, written as $x \models_S \phi$ or in short $x \models \phi$ if S is understood, is defined by the following conditions:

(1) $x \models a=v$ iff $I_a(x)=v$,

(2) $x \models \neg\phi$ iff not $x \models \phi$,

(3) $x \models \phi \wedge \varphi$ iff $x \models \phi$ and $x \models \varphi$,

(4) $x \models \phi \vee \varphi$ iff $x \models \phi$ or $x \models \varphi$.

If ϕ is a formula, the set

$$m_S(\phi)=\{x \in U \mid x \models \phi\}$$

is called the meaning of the formula ϕ in S. If S is understood, we simply write $m(\phi)$.

The meaning of a formula ϕ is therefore the set of all objects having the property expressed by the formula ϕ. In other words, ϕ can be viewed as the description of the set of objects $m(\phi)$. Thus, a connection between the formulas of L and subsets of U is established.

TABLE 1. An information table

Object	**height**	**hair**	**eyes**	**Class**
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	dark	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

To illustrate this idea, we consider an information table given by Table 1 (Quinlan 1983). The following expressions are some of the formulas of the language L:

(**height**, tall), (**hair**, dark),

(**height**, tall) $\wedge$ (**hair**, dark),

(**height**, tall) $\vee$ (**hair**, dark).

The meanings of the formulas are given by:

m((**height**, tall))=$\{o_3, o_4, o_5, o_6, o_7\}$,

m((**hair**, dark))=$\{o_4, o_5, o_7\}$,

m((**height**, tall) $\wedge$ (**hair**, dark))=$\{o_4, o_5, o_7\}$,

m((**height**, tall) $\vee$ (**hair**, dark))=$\{o_3, o_4, o_5, o_6, o_7\}$.

Usually, the knowledge implicated in information tables is expressed by rules which can be formulated as follows. A *rule* is a statement of the form: “if an object satisfies a formula, then the object must satisfy another formula”. The expression of rules can be formulated as follows (Pawlak 1991; Yao and Yao 2002).

Definition 5 (Rules): Let $S=(U, At, L, \{V_a \mid a\in At\}, \{I_a \mid a\in At\})$ be an information table, then a rule r is a formula with the form

$$\phi \Rightarrow \varphi,$$

where ϕ and φ are formulas of information table, and S for any $x\in U$,

$$x \models \phi \Rightarrow \varphi \text{ iff } x \models \neg\phi \vee \varphi.$$

Definition 6 (Decision Rules): Let $S=(U, C\cup D, L, \{V_a \mid a\in At\}, \{I_a \mid a\in C\cup D\})$ be a decision table, where C is the set of condition attributes and D is the set of decision attributes. A decision rule dr is a rule with the form $\phi \Rightarrow \varphi$, where ϕ, φ are both conjunction of atomic formulas, for any atomic formula (c, v) in ϕ, $c\in C$, and for any atomic formula (d, v) in φ, $d\in D$.

It is obvious that each object in a decision table can be expressed by a decision rule. The relationship between objects and rules can be defined by the following definition.

Definition 7 (Objects which are consistent or conflict with a rule): An object x is said to be consistent with a decision rule dr:$\phi \Rightarrow \varphi$, iff $x \models \phi$ and $x \models \varphi$; x is said to be conflict with dr, iff $x \models \phi$ and $x \models \neg\varphi$.

2.3. Decision Rule Set Function

We introduced the concept of decision rules in Section 2.2 and now we need to explore how to make decisions based on decision rules. We first describe decision rule sets and then define decision rule set functions.

Given a decision table $S=(U, At, L, \{V_a \mid a\in At\}, \{I_a \mid a\in At\})$, where $At = C \bigcup D$ and $D=\{d\}$. Suppose x and y are two variables, where $x\in X$ and $X=V_{a1}\times...\times V_{am}$, $y\in Y$ and $Y=V_d$. V_{ai} is the set of attribute a_i's values, $a_i\in C$, i=1 to m, m is the number of condition attributes. RS is a decision rule set generated from S.

Definition 8 (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. Here, x is called a condition variable, y is called a 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 objects with decision rule sets. In order to present the method of calculating the value of a decision rule-set function, we introduce a definition below about matching objects to decision rules.

Definition 9 (An object matching a decision rule): An object o is said to be matching a decision rule $\phi \Rightarrow \varphi$, if $o \models \phi$.

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

With the definition, a brief method for calculating the result of a decision rule set function is described as follows:

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

Step 2: Select a decision rule *dr* from MR_{RS}^{o} , where

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

Step 3: Set the decision value of object *o* to be v_d, i.e. $rs(o)=v_d$.

Here *dr* is called the final matching rule matched by object *o* in rule set *RS*. In Step 2, how to select a decision rule from MR_{RS}^{o} is the key task of the process. For example, there is a decision rule set *RS*:

1) $(a, 1) \wedge (b, 2) \Rightarrow (d, 2)$,

2) $(a, 2) \wedge (b, 3) \Rightarrow (d, 1)$,

3) $(b, 4) \Rightarrow (d, 2)$,

4) $(b, 3) \wedge (c, 2) \Rightarrow (d, 3)$,

and an undecided object:

$o = (a, 2) \wedge (b, 3) \wedge (c, 2)$.

With Step 1, $MR_{RS}^{o} = \{(a, 2) \wedge (b, 3) \Rightarrow (d, 1); (b, 3) \wedge (c, 2) \Rightarrow (d, 3)\}$.

With Step 2, if we select the final matching rule as $(a, 2) \wedge (b, 3) \Rightarrow (d, 1)$, then with Step 3,

$rs(o)=1$;

if select the final matching rule as $(b, 3) \wedge (c, 2) \Rightarrow (d, 3)$, then with Step 3,

$rs(o)=3$.

From the above example, we know that there may be more than one rule in MR_{RS}^{o} . In this case, when the decision values of these rules are different, the result would be controlled according to above method, known as uncertainty of a decision rule-set function. The method of selecting the final rule from MR_{RS}^{o} is thus very important, and is called the uncertainty solution method. In our research, we use AID-based rule trees (Def. 11) to deal with the problem.

2.4. Rule Trees

A rule tree is a compact and efficient structure for expressing a rule set. We first introduce the definition of rule trees in (Zheng and Wang 2004) as follows. We use it in this paper as the expression form of rule sets for a bilevel decision model.

Definition 10 (Rule tree):

(1) A rule tree is composed of one root node, some leaf nodes and some middle nodes;

(2) The root node represents the whole rule set;

(3) Each path from the root node to a leaf node represents a rule;

(4) Each middle node represents an attribute testing. Each possible value of an attribute in a rule set is represented by a branch. Each branch generates a new child node. If an attribute is reduced in some rules, then a special branch is needed to represent it and the value of the attribute in this rule is supposed as "*", which is different from any possible values of the attribute.

Figure 1 gives an example of a rule tree, where "Age", "Educational level (Edulevel)", "Seniority", and "Health" are its conditional attributes, and "Grade" is its decision attribute. The values of these attributes are noted beside the branches.

We define the number of nodes between a branch and the root node as the level of the branch (including the root node) in the path. For each rule tree, we make two assumptions as follows:

Assumption 1: The branches at the same level represent the possible values of the same attribute.

Here, an attribute is expressed by the level of a rule tree.

Assumption 2: If a rule tree expresses a decision rule set, the branches at the bottom level represent the possible values of the decision attribute.

Based on Def.10 and the two assumptions, we can improve the rule tree structure by considering the two constraints described in Def. 11.

Definition 11 (Attribute importance degree (AID) based rule tree): An AID-based rule tree is a rule tree, which satisfies the following two additional conditions:

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

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

In the rule tree illustrated in Figure 1, if we suppose

- ID(*a*) is the importance degree of attribute *a*, and

  ID(Age)>ID(Edulevel)>ID(Seniority)>ID(Health);

- (Age, Young) is better than (Age, Middle), and (Age, Middle) is better than (Age, Old);
- (Seniority, Long) is better than (Seniority, Short), and (Health, Good) is better than (Health, Poor),

then the rule tree illustrated by Figure 1 is an AID-based rule tree.

2.5. Rules Comparison and Confliction

Definition 12 (Comparison of rules): Suppose the condition attributes are ordered by their importance degrees as $a_1, \ldots, a_p$. Rule dr_1: $\wedge\{(a_i, v_{a1i})\} \Rightarrow (d_1, v_{d1})$ is said to be better than rule dr_2: $\wedge\{(a_i, v_{a2i})\} \Rightarrow (d_2, v_{d2})$, if there exists an index $k \in \{1, \ldots, p\}$ that satisfies:

(1) v_{a1k} is better than v_{a2k} or the value of a_k is deleted from rule dr_2;

(2) If $k>1$, then for each $j<k$, $v_{a1j}=v_{a2j}$.

If for each attribute a_i, v_{a1i} is with the same importance (or evaluation) degree as v_{a2i}, rule dr_1 has the same importance (or evaluation) degree as rule dr_2.

For example, we have two rules as follows:

dr_1: (Age, Middle) $\wedge$ (Working Seniority, Long) $\Rightarrow$ 2,

dr_2: (Age, Middle) $\wedge$ (Working Seniority, Short) $\Rightarrow$ 3,

and the value "Long" is better than the value "Short" in the attribute "Working Seniority", with Def. 12 we know dr_1 is better than dr_2.

Definition 13 (Rule confliction): Rule dr_1 is said to be conflict with rule dr_2, if

$$\text{for } \forall x \models dr_1,\ x \models \neg\, dr_2.$$

From Section 2.3, we know there are some uncertainties when make a decision with decision rule sets. The uncertainty can be eliminated through a process of rule selection. We can select a rule rightly only when related information is known. In other words, we are said to be informed only when we can select rules rightly and definitely. In this paper, we present a rule-tree based model to deal with these kinds of uncertainties. After the ordering of importance degrees and attributes' possible values, a rule tree (Def. 10) is improved to become an AID-based rule tree (Def. 11). It can be proved that the following theorems hold from the definition of AID-based rule trees (Zheng et.al 2009).

Theorem 1: In an AID-based rule tree, the rule expressed by the left branch is better than the rule expressed by the right branch.

Theorem 2: After being transformed to an AID-based rule tree, the rules in a rule set are totally in order, that is, every two rules can be compared.

Therefore, we can use an AID-based rule tree to solve the uncertainty problem of decision rule set functions. For example, we can order the rules expressed by the rule tree shown in Figure 1 as follows:

1) (Age, Young) $\wedge$ (Edulevel, High) $\Rightarrow$ 2,

2) (Age, Middle) $\wedge$ (Working Seniority, Long) $\Rightarrow$ 2,

3) (Age, Middle) $\wedge$ (Working Seniority, Short) $\Rightarrow$ 3,

4) (Age, Old) $\Rightarrow$ 4,

5) (Edulevel, Short) $\Rightarrow$ 4,

where rule i is better than rule i+1, i = 1, 2, 3, 4.

3. A RULE-SETS BESED BILEVEL DECISION MODEL AND A MODELLING APPROACH

This section will introduce our previous related work including a RSBLD model and an approach for modeling bilevel decision problems by rule sets.

3.1 A RSBLD Model

In principle, after emulating all possible situations in a decision domain, all objective functions can be transformed into a set of decision tables, known as objective decision tables. As decision rule sets have stronger knowledge expressing ability than decision tables, we use decision rule-set function to represent the objectives of the leader and follower of a bilevel decision problem in the proposed RSBLD model.

Similarly, after emulating all possible situations in a constraint field, the constraints can be formulated to an information table. When the information table is too big to be processed, it can be transformed to rule sets using the “Agrawal” methods provided by references (Agrawal, Imielinski, and Swami 1993; Agrawal and Srikant 1994).

By using rule sets, we have the following definition about constraint functions.

Definition 14 (Constraint Function): Suppose x is a decision variable and RS is a rule set, then a constraint function cf (x, RS) is defined as

$$cf\left(x, RS\right) = \begin{cases} \text{True}, & \text{if} \quad \text{for } \forall r \in RS, \quad x \in m\left(r\right) \\ \text{False}, & \text{else} \end{cases}. \tag{1}$$

The meaning of the constraint function *cf*(x, *RS*) is whether variable x belongs to the region constrained by *RS*.

Now, we can describe a RSBLD model as follows (Zhang et al 2009).

Definition 15 (RSBLD model):

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

$$\text{subject to } cf(x, G_L) = \text{True}$$

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

$$\text{subject to } cf(y, G_F) = \text{True} \qquad (2)$$

where x and y are decision variables (vectors) of the leader and the follower respectively; f_L and f_F are the objective decision rule set functions (Def. 8) of the leader and the follower respectively; *cf* is the constraint function; F_L and G_L are the objective decision rule set and constraint rule set of the leader; and F_F and G_F are the objective decision rule set and constraint rule set of the follower respectively.

3.2 An Approach for Modeling Bilevel Decision Problems by Rule Sets

In Zheng et al. (2009) we proposed an approach for modeling a bilevel decision problem by rule sets as follows.

Algorithm 1 (*An approach for modeling bilevel decision problems by rule sets*):

Input: A bilevel decision problem;

Output: A RSBLD model;

Step 1: Transform the bilevel decision problem with rule sets (information tables are as special cases);

Step 2: Pre-process F_L, such as delete reduplicate rules from the rule sets, eliminate noise, etc.;

Step 3: If F_L needs to be reduced,

then using a reduction algorithm to reduce F_L;

Step 4: Pre-process G_L, such as delete reduplicate rules from the rule sets, eliminate noise, etc;

Step 5: If G_L needs to be reduced,

then using reduction algorithm to reduce G_L;

Step 6: Pre-process F_F, such as delete reduplicate rules from the rule sets, eliminate noise, etc.;

Step 7: If F_F needs to be reduced,

then using a reduction algorithm to reduce F_F;

Step 8: Pre-process G_F, such as delete reduplicate rules from the rule sets, eliminate noise, etc.;

Step 9: If G_F needs to be reduced,

then using a reduction algorithm to reduce G_F.

Complete

In the algorithm, Step 1 is the key step of the modeling process. Decision makers (or experts) complete this step by transforming a bilevel decision problem to a set of information tables or related rule sets. This transformation can be done by laying out all possible situations of the bilevel decision problem.

In Steps 2, 4, 6 and 8, four sets of decision rule sets are pre-processed respectively. As data incompleteness, noisy, and inconsistency are the common characters for a huge real data set, we need to use related techniques to eliminate these problems before using the rule sets to model a bilevel decision problem (Han and Kamber 2001).

In Steps 5, 7, and 9 of Algorithm 1, related rule sets are reduced by applying a reduction algorithm. It is because of at least one of the following three reasons:

(1) When modeling a real-world bilevel decision problem, the rule sets in the model are often in a large scale, which is not convenient to be processed, and cannot be easily interpreted and understood.

(2) The rules in the rule sets are lack of adaptability. In this case, the rule sets cannot adapt new situations well, so it is unable or has poor ability to support decision making.

(3) The rule sets in the model are just original data sets, the patterns in such data sets are needed to be extracted, and the results are more general rules.

The detailes of the algorithm can be obtained from our previous work (Zheng et al. 2009). Now we give some analysis about the complexity of Algorithm 1. Obviously, it can be estimated as the integration of the complexity of Step 1, Steps 2, 4, 6, 8 and Steps 3, 5, 7, 9 respectively.

Suppose p_{oL} and p_{oF} are the numbers of the rules in the objective decision rule sets of the leader and the follower generated in Step 1 respectively, p_{cL}, and p_{cF} are the numbers of the rules in the constraint rule sets of the leader and the follower generated in Step 1 respectively, and m_L and m_F are the numbers of the condition attributes of the leader and the follower. For Step 1, the complexity is

$$O((m_L + m_F)(p_{oL} + p_{oF} + p_{cL} + p_{cF})) .$$

For Steps 2, 4, 6, 8, different pre-process methods can cause different complexities. For above mentioned pre-process methods, the complexity is between $O((m_L + m_F)p)$ and $O\left((m_L + m_F)p^2\right)$, where p=p_{oL} for Step 2, p= p_{oF} for Step 4, p= p_{cL} for Step 6, and p= p_{cF} for Step 8.

For Steps 3, 5, 7, and 9 the time complexity depends on the sizes of the processed rule sets. Using the methods mentioned above, it has complexity

$$O\left(\left(m_L + m_F\right) \cdot p \cdot \left(p-1\right)\right),$$

where $p=p_{oL}$ for Step 3, $p=p_{oF}$ for Step 5, $p=p_{cL}$ for Step 7, and $p=p_{cF}$ for Step 9.

Therefore, Algorithm 1 has the maximal time complexity

$$O\left(\left(m_L + m_F\right)\left(p_{oL}{}^2 + p_{oF}{}^2 + p_{cL}{}^2 + p_{cF}{}^2\right)\right).$$

In Section 6, we will use a case based example to illustrate the modeling process of a bilevel decision problem by using the proposed algorithm. In Section 7, a set of experiments are designed to test the complexity of the algorithm.

4. TRANSFORMATION THEOREM FOR RSBLD PROBLEMS

In this section, we explore how to transform a RSBLD problem to a single level one, where the two problems have the same optimal solution. A transformation theorem will be proposed to show the solution equivalence for the two problems. First, we give a definition below.

Definition 16 (Combination rule of two decision rules): Suppose dr_1: $\phi_1 \Rightarrow \left(d_1, v_1\right)$ and dr_2: $\phi_2 \Rightarrow \left(d_2, v_2\right)$ are two decision rules and they are not conflict, then the combination rule of them are denoted as $dr_1 \cap dr_2$ with the form

$$\phi_1 \wedge \phi_2 \Rightarrow \left(d, \left(v_1, v_2\right)\right), \tag{3}$$

where d_1, d_2, and d are the decision attributes of dr_1, dr_2 and dr respectively, v_1, v_2 and (v_1, v_2) are the decision values of dr_1 and dr_2 and dr respectively.

Here, v_1, v_2 are called the leader decision and the follower decision of dr respectively.

For example, suppose

dr_1: (Age, Young) $\Rightarrow$ 2,

dr_2: (Working Seniority, Long) $\Rightarrow$ 2,

then the combination of the two rules is

dr: (Age, Young) $\wedge$ (Working Seniority, Long) $\Rightarrow$ (d, (2, 2)).

Suppose the objective rule sets are expressed by AID-based rule trees, then the transformation process can be presented as follows.

Step 1(Initialization): Let CT be an empty attribute importance degree based rule tree;

Step 2 (Construct a new rule tree):

For each rule dr_L in FT_L

For each decision rule dr_F in FT_F

{If dr_L are not conflict with dr_F, then

Add rule $dr_L \cap dr_F$ to CT;}

Complete

Suppose the combined rule set is noted as F, then the single level rule-sets based decision problem can be formulated as:

$$\begin{aligned} &\min_{x,y} \ f(x, y) \\ &\text{s.t. } cf(x, G_L) = \text{True} \\ &\quad\quad cf(y, G_F) = \text{True}, \end{aligned} \tag{4}$$

where x and y are variables of the leader and the follower respectively; f is the objective decision rule-set function; cf is the constraint function; F, G_L, G_F are the objective decision rule set, leader's constraint rule set and follower's constraint rule set respectively.

With the following theorem, we can prove the solution equivalence of the original problems and the transformed problem.

Theorem 3: The RSBLD model presented in Equation (2) has an optimal solution (x, y), iff (x, y) is an optimal solution of its corresponding single level decision model presented in Equation (7).

Proof: Suppose x and y are variables of the leader and the follower respectively, f_L and f_F are the objective rule-set functions of the leader and the follower respectively in Equation (2), and f is the objective rule set function in Equation (7). F_L and F_F are the objective rule sets of the leader and the follower in the RSBLD model, and F is the objective rule set in the single level decision model.

$(\Rightarrow)$

If the optimal solution of the RSBLD model presented in Equation (2) is (x, y), and

$$f_L(x, y)= v_L \text{ and } f_F (x, y)=v_F.$$

Suppose the final matching rules (Section 2) of (x, y) in rule sets F_L and F_F are dr_L and dr_F respectively. Then, from the process of transformation, we know the rule $dr_L \cap dr_F$ belongs to the combined rule set F.

Because (x, y) is the optimal solution of the RSBLD model, dr_L and dr_F must be the best rules having the minimal decision values in F_L and F_F respectively. Thus, $dr=dr_L \cap dr_F$ must be the best rules matched by (x, y) in F. Besides, because (x, y) is the best object satisfying dr_L and dr_F

both, thus (x, y) is the best object satisfying dr. Thus, (x, y) is the optimal solution of the single level decision model presented in Equation (7).

The sufficient condition of the theorem is proved.

$(\Leftarrow)$

If the optimal solution of the single level decision model presented in Equation (7) is (x, y), and

$$f(x, y)=(d, (v_{Ld}, v_{Fd})).$$

Suppose the final matching rule of (x, y) in rule set F is dr, then from the process of transformation, there must be two decision rules dr_L in F_L and dr_F in F_F that $dr= dr_L \cap dr_F$. If there is more than one rule pair dr_L and dr_F satisfying that $dr= dr_L \cap dr_F$, then select the best one among them.

Because (x, y) is the optimal solution of the single level decision model, dr must be the best rules having the minimal decision value in F. Thus, dr_L and dr_F must be the best rules matched by (x, y) in F_L and F_F respectively. Besides, because (x, y) is the best object satisfying dr, thus (x, y) is the best object satisfying dr_L and dr_F both. So, (x, y) is the optimal solution of the bilevel decision model.

Thus, the necessary condition of the theorem is proved. □

From Theorem 3, we know the solutions of the RSBLD problem presented in Equation (2) and its transformed problem shown in Equation (7) are equivalent. Therefore, we can transform any RSBLD problem into a single level decision problem and get a solution through solving the single level decision problem. We need to indicate that although the original bilevel decision problem and the transformed one level problem have the same optimal solution, they are not equivalent. However, the transformation can achieve our aim, that is, to generate a model which can be easily solved but has the same optimal solution with the original bilevel decision model.

5. A TRANSFORMATION BASED SOLUTION ALGORITHM FOR RSBLD PROBLEMS

Based on the transformation theory proposed**,** this section gives a transformation based solution algorithm for RSBLD problems. To describe the algorithm clearly, we first give some important definitions.

Definition 17:

(a) Constraint region of a bilevel decision problem:

$$S = \{(x, y):\ cf\left(x, G_L\right) = \text{True},\ cf\left(y, G_F\right) = \text{True}\} \tag{5}$$

(b) Feasible set for the follower for each fixed x:

$$S(x)=\{y: (x, y)\in S\} \tag{6}$$

(c) Projection of S onto the leader's decision space:

$$S(X)=\{x:\ \exists y, (x, y)\in S\} \tag{7}$$

(d) Follower's rational reaction set for $x\in S(X)$:

$$P(x)=\{y: y\in \arg\min_{y'}\left[\ f_F\left(x, y'\right) : y' \in\ S\left(x\right)\right]\} \tag{8}$$

(e) Inducible region:

$$IR=\{(x, y): (x,\ y)\in S, y\in P(x)\} \tag{9}$$

From the features of the bilevel decision problem, it is obvious that once the leader selects a value of x, the first term in the follower's objective function becomes a constant and can be removed from the problem. In this case, we replace $f_F(x, y)$ with $f_F(y)$.

To ensure that a RSBLD model is well posed it is common to assume that S is in nonempty and compact, and that for all decisions taken by the leader, the follower has some room to respond, i.e. $P(x) \neq \Phi$. The rational reaction set $P(x)$ defines the response while the inducible region IR represents the set over which the leader may optimize. Thus in terms of the above notation, the bilevel decision problem can be written as

$$\min \{ f_L(x, y) : (x, y) \in IR \}.$$

Now, we can give a description of the new algorithm. The algorithm has two stages. It first transforms a bilevel decision problem described by a RSBLD model to a single level one. It then solves the single level problem to get a solution. The solution obtained is of the original RSBLD problem. For simple description, we suppose the importance degrees of the leader's condition attributes are more than those of the follower's. That means, the branches representing the possible values of the leader's condition attributes are at higher levels of AID-based rule trees than those of the follower's. The detail of the transformation based algorithm is as follows.

Algorithm 2 (*A transformation based solution algorithm for RSBLD problems*):

Input: The objective decision rule set $F_L=\{dr_{L1}, \ldots , dr_{Lp}\}$ and the constraint rule set G_L of the leader, the objective decision rule set $F_F=\{dr_{F1}, \ldots , dr_{Fq}\}$ and the constraint rule set G_F of the follower;

Output: An optimal solution of the RSBLD problem (*ob*);

Step 1: Construct the objective rule tree FT_L of the leader by F_L;

Step 1.1: Arrange the condition attributes in ascending order according to the importance degrees. Let the attributes be the discernible attributes of levels from the top to the bottom of the tree;

Step 1.2: Initialize FT_L to an empty AID-based rule tree;

Step 1.3: For each rule R of the decision rule set F_L {

Step 1.3.1: let CN=root node of the rule tree FT_L;

Step 1.3.2: For i=1 to m /**m is the number of levels in the rule tree**/

{ If there is a branch of CN representing the ith discernible attribute value of rule R, then

let CN=node I; /*node I is the node generated by the branch*/

else {Create a branch of CN to represent the ith discernible attribute value;

According to the value order of the ith discernible attribute, put the created branch to the right place;

Let CN=node J /*node J is the end node of the branch*/}}}

Step 2: Construct the objective rule tree FT_F by F_F;

The detail of Step 2 is similar to that in Step 1. What needs to be done is to replace FT_L with FT_F and replace F_L with F_F in the sub-steps of Step 1.

Step 3: Transform the bilevel decision problem to a single level one, and the resultant objective rule tree is CT;

Step 4: Use the constraint rule sets of both the leader and the follower to prune CT;

Step 4.1: Generate an empty new AID-based rule tree CT';

Step 4.2: For each rule dr in G_L and G_F,

Add the rules in CT to CT' that are consistent with dr to FT_L';

Delete the rules in *CT* and *CT*´ that are conflict with ruel *dr*;

Step 4.3: Let *CT*=*CT*´;

Step 5: Search for the lefmost rule *dr* in *CT* whose leader decision and follower decision are both minimal;

Step 6: If *dr* does not exist, then

There isn't an optimal solution for the problem;

Go to End;

Step 7: $OB=\{ob|\ ob \models dr$ and for $\forall\ r \in G_L \cup G_F,\ ob \models r\ \}$;

Step 8: If there is more than one object in *OB*, then

According to Def. 12, select the best or most important object *ob*;

else

ob=the object in *OB*;

Step 9: *ob* is the optimal solution of the RSBLD problem.

Complete

The flow chart of the algorithm is illustrated in Figure 2. By this algorithm, we can obtain a solution for a bilevel decision problem through solving the transformed single level problem. The time complexity of the new algorithm is

$$O\left(n_{oL} n_{oF}\left(n_{cL}+n_{cF}\right)\left(m_L+m_F\right)\right),$$

where n_{oL}, n_{oF} are numbers of the rules in the objective rule sets of the leader and the follower, n_{cL}, n_{cF} are numbers of the rules in the constraint rule sets of the leader and the follower, m_L and m_F are the numbers of the condition attributes of the leader and the follower respectively.

6. A CASE BASED EXAMPLE

A factory's human resource management system is distributed into two levels. The upper level is the factory executive committee and the lower is the workshop management committee. For the recruitment policy, the executive committee mainly considers how to meet the overall business objectives with a long term development plan, and the workshop management committee concentrates on the current daily needs of workers. Obviously, their objectives are different. However, their objectives are transparent to each other though they may operate in separate ways. A recruitment action will ultimately emerge that is the optimal result for the company as a whole but will also consider current daily needs. This is a typical bilevel decision problem, in which the company executive committee is as the leader, and the workshop management committee, the follower.

When determining whether a person could be recruited for a particular position, the factory executive committee mainly considers the following two factors, the "age" and "education level (edulevel)" of the person, and the workshop management committee mainly considers another two factors, "seniority" and "health". Suppose the condition attributes in ascending order according to the importance degree are "age", "edulevel", "seniority", and "health".

Obviously, it is hard for the two committees to express the conditions of the workers whom they want to recruit to linear or nonlinear functions. But they have the data of the workers having already been recruited in their databases. We can therefore build two decision tables as shown in Tables 2 and 3, and then generate decision rule sets from these two tables to represent the objectives of the two committees. The condition attributes of the two decision tables are the factors; the decision attributes of the two decision tables are acceptance grades of the workers.

The constraints of the two committees are expressed by simple rule sets (Equations 13, 14), which define the constraint regions.

Now, we use algorithm 1 to establish a RSBLD model from the problem.

Alg. 1-Step 1: Transform the problem with decision rule sets.

As indicated above, the objective rule sets and constraint rule sets of the leader and the follower are described in Tables 2, 3 and Equations 13, 14 respectively.

TABLE 2. Objective rule set of the leader

Age	Edulevel	Seniority	Health	Grade
Young	High	Middle	Good	2
Middle	High	Long	Middle	2
Young	Short	Short	Poor	4
Young	Middle	Middle	Middle	2
Middle	Middle	Short	Middle	3
Middle	Middle	Long	Middle	2
Old	High	Long	Middle	3
Young	Short	Middle	Poor	2
Middle	Short	Short	Middle	4
Old	Short	Middle	Poor	4
Middle	Short	Long	Good	3
Middle	Short	Long	Middle	2
Old	High	Middle	Poor	3
Old	High	Long	Good	2
Old	Short	Long	Good	4
Young	High	Long	Good	4
Young	Short	Long	Middle	3

The constraint rule set of the leader is:

$$G_L = \{\text{True} \Rightarrow (\text{Age, Young}) \vee (\text{Age, Middle})\} \tag{10}$$

TABLE 3. Objective decision table of the follower

Age	Edulevel	Seniority	Health	Grade
Young	High	Long	Good	2
Old	Short	Short	Good	4
Young	High	Short	Good	2
Old	High	Long	Middle	3
Young	Short	Long	Middle	4
Middle	High	Middle	Poor	3

Middle	Short	Short	Poor	4
Old	Short	Short	Poor	4
Old	High	Long	Good	2
Young	Short	Long	Good	2
Young	Short	Middle	Middle	3
Middle	Short	Middle	Good	3
Old	High	Long	Good	2
Middle	High	Long	Good	2
Middle	High	Short	Poor	4

The constraint rule set of the follower is:

$$G_F=\{\text{True} \Rightarrow (\text{Seniority, Long}) \vee (\text{Seniority, Middle})\} \tag{11}$$

Because the scale of the data is very small, the preprocess steps (Steps 2, 4, 6 and 8) are passed over. Besides, the constraint rule sets of the leader and the follower are brief enough, so the reduction steps of G_L and G_F (Step 5 and Step 9) can be ignored.

Alg. 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 the reduced objective rule sets of the leader and the follower as shown in Equations (15) and (16). Here, we use the decision matrices based value reduction algorithm (Ziarko 1996) in the RIDAS system (Wang, Zheng, and Zhang 2002).

The refined objective rule set of the leader is:

F_L= {(Age, Young) $\wedge$ (Seniority, Middle) $\Rightarrow$ (Grade, 2)

(Age, Middle) $\wedge$ (Edulevel, High) $\Rightarrow$ (Grade, 2)

(Edulevel, Short) $\wedge$ (Seniority, Short) $\Rightarrow$ (Grade, 4)

(Edulevel, Middle) $\wedge$ (Seniority, Short) $\Rightarrow$ (Grade, 3)

(Edulevel, Middle) $\wedge$ (Seniority, Long) $\Rightarrow$ (Grade, 2)

(Age, Old) $\wedge$ (Health, Middle) $\Rightarrow$ (Grade, 3)

(Age, Old) $\wedge$ (Edulevel, Short) $\Rightarrow$ (Grade, 4)

(Age, Middle) $\wedge$ (Health, Good) $\Rightarrow$ (Grade, 3)

(Age, Middle) $\wedge$ (Seniority, Long) $\wedge$ (Health, Middle) $\Rightarrow$ (Grade, 2)

(Age, Old) $\wedge$ (Edulevel, High) $\wedge$ (Health, Good) $\Rightarrow$ (Grade, 2)

(Edulevel, High) $\wedge$ (Health, Poor) $\Rightarrow$ (Grade, 3)

(Age, Young) $\wedge$ (Edulevel, High) $\wedge$ (Seniority, Long) $\Rightarrow$ (Grade, 4)

(Age, Young) $\wedge$ (Edulevel, Short) $\wedge$ (eniority, Long) $\Rightarrow$ (Grade, 3)} (12)

The refined objective rule set of the follower is:

F_F={ (Edulevel, High) $\wedge$ (Health, Good) $\Rightarrow$ (Grade, 2)

(Edulevel, Short) $\wedge$ (Seniority, Short) $\Rightarrow$ (Grade, 4)

(Age, Old) $\wedge$ (Health, Middle) $\Rightarrow$ (Grade, 3)

(Age, Young) $\wedge$ (Seniority, Long) $\wedge$ (Health, Middle) $\Rightarrow$ (Grade, 4)

(Seniority, Middle) $\Rightarrow$ (Grade, 3)

(Seniority, Long) $\wedge$ (Health, Good) $\Rightarrow$ (Grade, 2)

(Seniority, Short) $\wedge$ (Health, Poor) $\Rightarrow$ (Grade, 4)} (13)

With above steps, we get the RSBLD model of the decision problem as follows:

$$\begin{aligned} &\min_x \ f_L(x, y) \\ &\quad \text{subject to } cf(x, G_L) = \text{True} \\ &\quad \min_y \ f_F(x, y) \\ &\quad\quad \text{subject to } cf(y, G_F) = \text{True}, \end{aligned} \tag{17}$$

where f_L, f_F are the corresponding decision rule set functions of F_L, F_F respectively.

Now, we use the Alg. 2 to solve the RSBLD problem. We suppose the four condition attributes are ordered as 'age', 'edulevel', 'seniority', and 'health'.

Alg. 2-Step 1: Construct the objective rule tree FT_L of the leader by F_L, and the result is illustrated by Figure 3;

Alg. 2-Step 2: Construct the objective rule tree FT_F of the follower by F_F, and the result is illustrated by Figure 4;

Alg. 2-Step 3: Transform the RSBLD problem to a single level one, and the resulted objective rule tree *CT* is illustrated by Figure 5;

Alg. 2-Step 4: Use the constraint rule sets of both the leader and follower to prune *CT*, and the result is illustrated by Figure 6;

Alg. 2-Step 5: Search for the leftmost rule *dr* in *CT* whose leader decision and follower decision are both minimal, and the result is

dr: (Age, Young) $\wedge$ (Edulevel, High) $\wedge$ (Seniority, Middle) $\wedge$ (Health, Good) $\Rightarrow$ (*d*, (2, 2));

Alg. 2-Step 6: *OB*={*ob*| *ob* is the object satisfying:

(Age, Young) $\wedge$ (Edulevel, High) $\wedge$ ((Seniority, Middle) $\wedge$ (Health, Good) };

Alg. 2-Step 7: *ob*=(Age, Young) $\wedge$ (Edulevel, High) $\wedge$ (Seniority, Middle) $\wedge$ (Health, Good);

Alg. 2-Step 8: *ob* is the final solution of the RSBLD problem.

In Figures 3 - 6, these attribute values are represented by its first letter.

7. EXPERIMENTS AND ANALYSIS

In order to test the effectiveness of the proposed rule-sets based bilevel decision problem modeling algorithm (Algorithm 1) and solution algorithm (Algorithm 2), we implemented these two algorithms within Matlab 6.5. We then used some classical data sets from the UCI database to test them by a set of experiments. UCI database (http://www.ics.uci.edu/~mlearn/MLRepository.html) consists of many data sets that can be used by the decision systems and machine learning communities for the empirical analysis of algorithms.

For each data set we chosen, we first select half of a data set as the original objective rule set of the leader, and the remaining as the original objective rule set of the follower. We assume that there are no constraints, which means all objects consistent with the objective rule sets are in the constraint region. Besides, we suppose the first half of the condition attributes are the ones for the leader and the others for the follower. The importance degrees of the condition attributes are descending order from the first condition attribute to the last condition attribute. The two experiments are processed on a computer with 2.33GHz CPU and 2G memory space. We describe these two experiments respectively as follows.

Experiment 1: Testing of Algorithm 1 with the data sets in the UCI database.

Step 1. Randomly choose 50% of the objects from the data set to be the original objective decision rule set of the leader, and the remaining 50% of the objects to be the original objective decision rule set of the follower;

Step 2. Apply Algorithm 1 to construct a rule-sets based bilevel decision model by using the chosen rule sets. Here, we use the decision matrices based value reduction algorithm (Ziarko, Cercone, and Hu 1996) in the RIDAS system (Wang, Zheng, and Zhang 2002) to reduce the sizes of original rule sets.

Experiment 2: Testing of Algorithm 2 with the data sets in the UCI database.

Following Steps 1 and 2 in Experiment 1, we have

Step 3. Apply Algorithm 2 to get a solution from the generated rule-sets based bilevel decision model in Experiment 1.

The complexity of the two algorithms (algorithms 1 and 2) is also tested through conducting these two experiments. As showed in Table 4, p_{OL} and p_{OF} are the numbers of objects in the original decision rules of the leader and the follower respectively (Refer to Step 1 of Algorithm 1), m_L and m_F are the condition attribute numbers of the leader and the follower respectively, n_{OL} and n_{OF} are the numbers of the rules in the reduced objective decision rule sets of the leader and the follower respectively, t_1 and t_2 are the processing times of Algorithms 1 and 2 respectively.

TABLE 4. Testing results of Algorithms 1 and 2

Data Sets	p_{OL}	p_{OF}	m_L	m_F	n_{OL}	n_{OF}	Alg. 1	Alg. 2
							t_1(sec.)	t_2 (sec.)
LENSES	12	12	2	3	6	3	<0.01	0.03
HAYES-ROTH	50	50	2	3	21	24	<0.01	0.09
AUTO-MPG	199	199	4	4	80	76	0.08	0.39
BUPA	172	172	3	3	159	126	0.06	3.10
PROCESSED_ CLEVELAND	151	151	6	7	115	127	0.28	5.20
BREAST-CANCER-WISCONSIN	349	349	5	5	47	47	0.51	0.63

From the results shown in Table 4 we can find that

1) The processing time of Alg. 1 highly relates with the numbers of the rules in the original objective decision rule sets and the condition attribute numbers of the leader and the follower respectively, expressed by the symbols p_{OL}, p_{OF}, m_L and m_F.

2) The processing time of Alg. 2 highly relates with the numbers of the rules in the reduced objective decision rule sets and the condition attribute numbers of the leader and the follower respectively, expressed by n_{OL}, n_{OF}, m_L and m_F.

These are consistent with our complexity analysis results in Sections 3 and 5.

8. CONCLUSION AND FUTURE WORK

Bilevel decision making is a common issue in organizational management activities. As many bilevel decision problems are difficult to model with mathematical functions, RSBLD models are proposed, in which all objective functions and constraint functions are expressed by rule sets. Based on our previous research, this paper presents a transformation based algorithm to solve a RSBLD problem. Some experiments have proved the functions and the effectiveness of the proposed solution algorithm.

In the traditional BLP model, Kuhn-Tucker conditions (Bard and Falk 1982; Bialas and Karwan 1982; Hansen, Jaumard, and Savard 1992) are used to transform a BLP model to a single level one. The basic idea of the transformation proposed in this paper is different from the Kuhn-Tucher condition based transformation. In the solution algorithm of traditional BLP problems, the follower’s problem is transformed to constraints, while in the solution algorithm proposed for RSBLP problems, the objective functions of the leader and the follower are combined to one. Besides, the most important issue is that the two transformations solve different models of bilevel problems, as one is a RSBLD problem and another is a bilevel linear programming.

Further study will include the development of approaches for multi-objectives or multi-followers RSBLD problems. A comprehensive bilevel decision support system is being developed to implement the proposed techniques for supporting real decision makers to solve their bilevel decision problems effectively.

ACKNOWLEDGMENT

The work presented in this paper was supported by Australian Research Council (ARC) under discovery grant DP0557154.

REFERENCES

AGRAWAL, R., T. IMIELINSKI, and A. SWAMI. 1993. Mining association rules between sets of items in large databases. *In* the Proceeding of The ACM SIGMOD Conference on Management of Data, pp. 207-216.

AGRAWAL, R., and R. SRIKANT. 1994. Fast algorithms for mining association rules. *In* the Proceeding of 20th International Conference of Very Large Databases, pp. 487-499.

AIYOSHI, E., and K. SHIMIZU. 1981. Hierarchical decentralized systems and its new solution By A Barrier Method. IEEE Transactions on Systems, Man, and Cybernetics, **11**:444-449.

ANANDALINGAM, G., and T. FRIESZ. 1992. Hierarchical optimization: An introduction. Annals of Operations Research, **34**:1-11.

BARD, J. F., and J. E. FALK. 1982. An explicit solution to the multi-level programming problem. Computers & Operations Research, **9**:77-100.

BARD, J. F.. 1984. An investigation of the linear three level programming problem. IEEE Transactions on Systems, Man, and Cybernetics, **14**:711-71.

BARD, J. F., and J. T. MOORE. 1992. An algorithm for the discrete bilevel programming problem. Naval Research Logistics, **39**:419-435.

BARD, J. F. 1998. Practical Bilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers, USA.

BIALAS, W., and M. H. KARWAN. 1978. Multilevel linear programming. Technical Report 78-1, State University of New York at Buffalo, Operations Research Program.

BIALAS, W. F., and M. H. KARWAN. 1982. On two-level optimization. IEEE Transactions on Automatic Control, **AC-26**: 211-214.

BIALAS, W., and M. H. KARWAN. 1984. Two-level linear programming. Management Science, **30**:1004-1020.

CALVETE, H., I., C. GALÉ, and P. M. MATEO. 2008. A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research, **188**(1):14-28.

CANDLER, W., and R. TOWNSLEY. 1982. A linear two-level programming problem. Computers and Operations Research, **9**:59-76.

CHEN, Y., M. FLORIAN, and S. WU. 1992. A descent dual approach for linear bilevel programs. Technical Report CRT-866, Centre de Recherche sur les Transports.

DEMPE, S. 1987. A simple algorithm for the linear bilevel programming problem. Optimization, **18**:373-385.

HANSEN, P., B. JAUMARD, and G. SAVARD. 1992. An extended branch-and-bound rules for linear bilevel programming. SIAM Journal on Scientific and Statistical Computing, **13**:1194-1217.

HAN, J., and M. Kamber. 2001. Data mining concepts and techniques. Morgan Kaufmann Publishers.

LAI, Y. J. 1996. Hierarchical optimization: a satisfactory solution. Fuzzy Sets and Systems, 77:321-335.

LAN, K. M., U. P. WEN, H. S. SHIH and E. S. LEE. 2007. A hybrid neural network approach to bilevel programming problems. Applied Mathematics Letters, **20**(8):880-884.

LIANG, L., and S. H. SHENG. 1992. The stability analysis of bilevel decision and its application. Decision And Decision Support System, **2**:63-70.

Quinlan, J. R. 1983. Learning efficient classification procedures and their application to chess end-games. In J.S. Michalski, J.G. Carbonell, & T.M. Mirchell, *Machine Learning: An Artificial Intelligence Approach* (pp. 463-482), *1*. Morgan Kaufmann, Palo Alto, CA.

PAWLAK, Z. 1991. Rough Sets Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston.

WANG, G. M., X. J. WANG, Z. P. WAN and Y. B. LV. 2007. A globally convergent algorithm for a class of bilevel nonlinear programming problem. Applied Mathematics and Computation, **188**(1):166-172.

WANG, G. Y. 2001. Rough Set Theory and Knowledge Acquisition. Press of Xi'an University of Communications.

WANG, G. Y., Z. ZHENG, and Y. ZHANG. 2002. RIDAS-a rough set based intelligent data analysis system. *In* the Proceeding of the first international conference on machine learning and cybernetics, pp. 646-649.

WHITE, D., and G. ANANDALINGAM. 1993. A penalty function approach for solving bi-Level linear programs. Journal of Global Optimization, **3**:397-419.

SAKAWA, M., I. NISHIZAKI, and Y. UEMURA. 2000a. Interactive fuzzy programming for multilevel linear programming problems with fuzzy parameters. Fuzzy Sets and Systems, **109**:3-19.

SAKAWA, M., I. NISHIZAKI, and Y. UEMURA. 2000b. Interactive fuzzy programming for two-level linear fractional programming problems with fuzzy parameters. Fuzzy Sets and Systems, **115**: 93-103.

SAKAWA, M., and K. YAUCHI. 2000. Interactive decision making for multiobjective nonconvex programming problems with fuzzy numbers through coevolutionary genetic algorithms. Fuzzy Sets and Systems, **114**:151-165.

SAKAWA, M., and I. NISHIZAK. 2001a. Interactive fuzzy programming for two-level linear fractional programming problems. Fuzzy Sets and Systems, **119**:31-40.

SAKAWA, M., and I. NISHIZAK. 2001b. Interactive fuzzy programming for decentralized two-level linear programming problems. Fuzzy Sets and Systems, **125**:301-315.

SAKAWA, M., and I. NISHIZAK. 2002. Interactive fuzzy programming for two-level nonconvex programming problems with fuzzy parameters through genetic algorithms. Fuzzy Sets and Systems, **127**:185-197.

SAVARD, G. 1989. Contributions á la programmation mathématique á deux niveaux. PhD thesis. Université de Montréal, École Polytechnique.

SHI, H. S., Y. J. LAI, and E. S. LEE. 1996. Fuzzy approach for multi-level programming problems. Computers & Operations Research, **23**:73-91.

STACKELBERG, H. V. 1952. The Theory of Market Economy, Oxford University Press, Oxford.

TARSKI, A. 1956. The concept of truth in formalized language, logic semantics, metamathematics. Clarendon Press, Oxford.

YAO, Y. Y., and J. T. YAO. 2002. Granular computing as a basis for consistent classification problems. Communications of Institute of Information and Computing Machinery (special issue of PAKDD'02 Workshop on Toward the Foundation of Data Mining), **5**:101-106.

ZIARKO, W., N. CERCONE, N. and X. HU. 1996. Rule discovery from databases with decision matrices. Lecture Notes In Computer Science, **1079**:653-662.

ZHANG, G., and J. LU. 2005. The definition of optimal solution and an extended Kuhn-Tucker approach for fuzzy linear bilevel programming. IEEE Computational Intelligence Bulletin, **2**(5):1-7.

ZHANG, G., and J. LU. 2006. Model and approach of fuzzy bilevel decision making for logistics planning problem. Journal of Enterprise Information Management, **20**:178-197.

ZHANG, G., J. LU and T. DILLON. 2007a. Fuzzy linear bilevel optimization: Solution concepts, approaches and applications. in Fuzzy Logic – A Spectrum of Theoretical and Practical Issues, P. P. Wang, D. Ruan, and E. E. Kerre, Eds. Berlin: Springer, pp. 351-379.

ZHANG, G., J. LU, and T. DILLON. 2007b. Solution concepts and an approximation Kuhu-Tucker approach for fuzzy multi-objective linear bilevel programming. in Pareto Optimality, Game Theory and Equilibria, Chinchuluun, Migdalas, Pardalos and Pitsoulis, Eds. Berlin: Springer, pp. 307-330.

ZHENG, Z., and G. Y. WANG. 2004. RRIA: a rough set and rule tree based incremental knowledge acquisition algorithm. Fundamenta Informaticae, **59**:299-313.

ZHENG, Z., J. LU, G. ZHANG, and Q. HE. 2009. Rule Sets based Bilevel Decision Model and Algorithm. Expert Systems With Applications, **36**:.28-26.

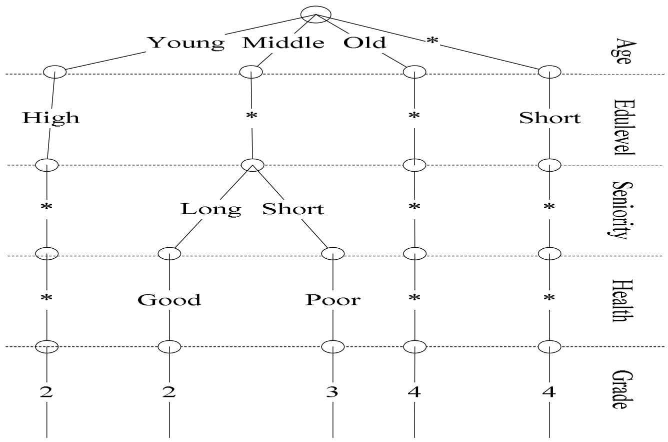

FIGURE 1. An example of rule tree

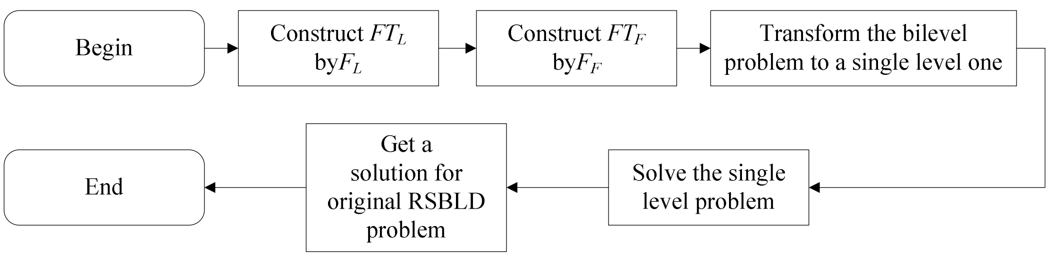

FIGURE 2. Flow chart of Alg. 1

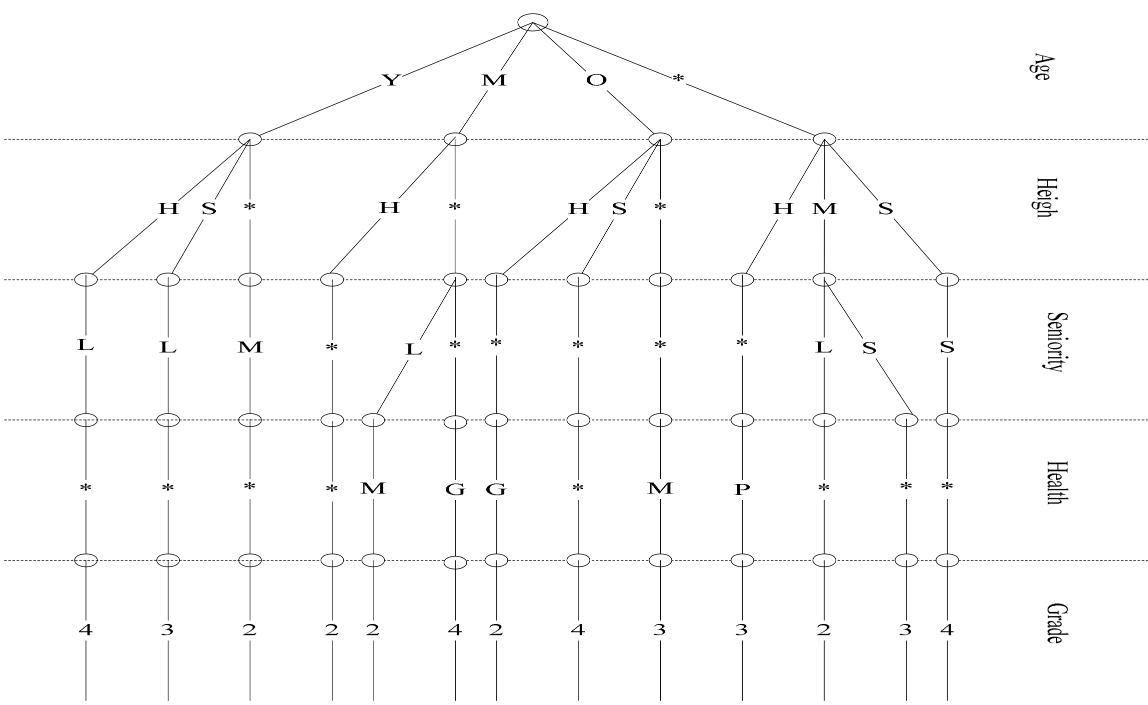

FIGURE 3. Rule tree of the leader's objective rule set

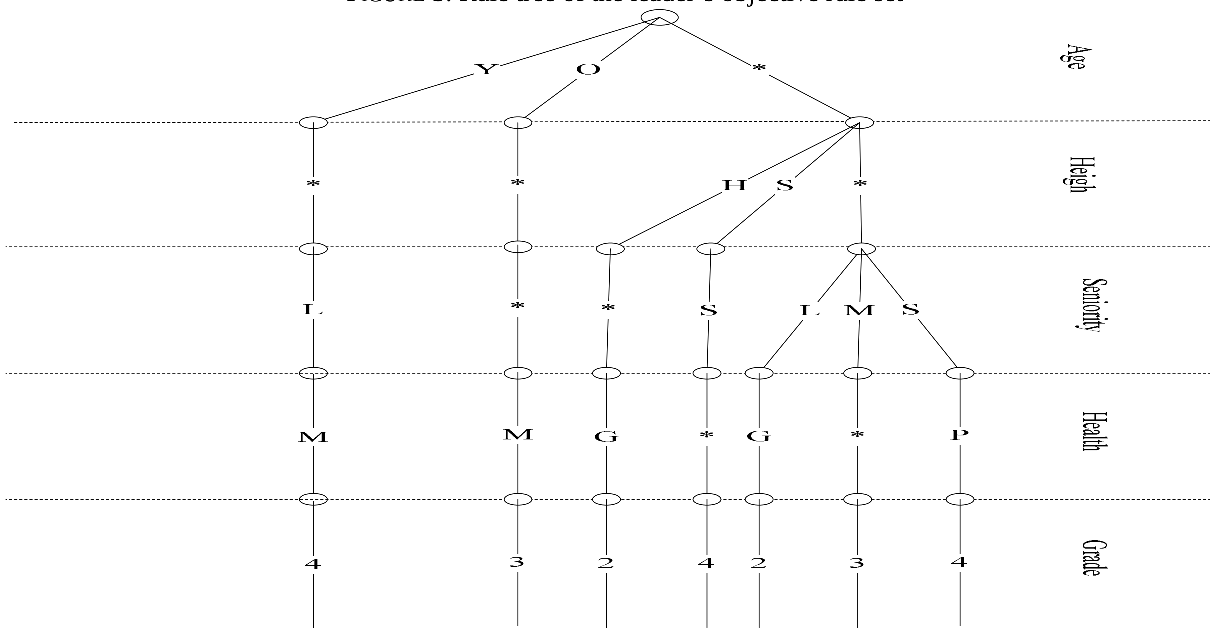

FIGURE 4. Rule tree of the follower's objective rule set

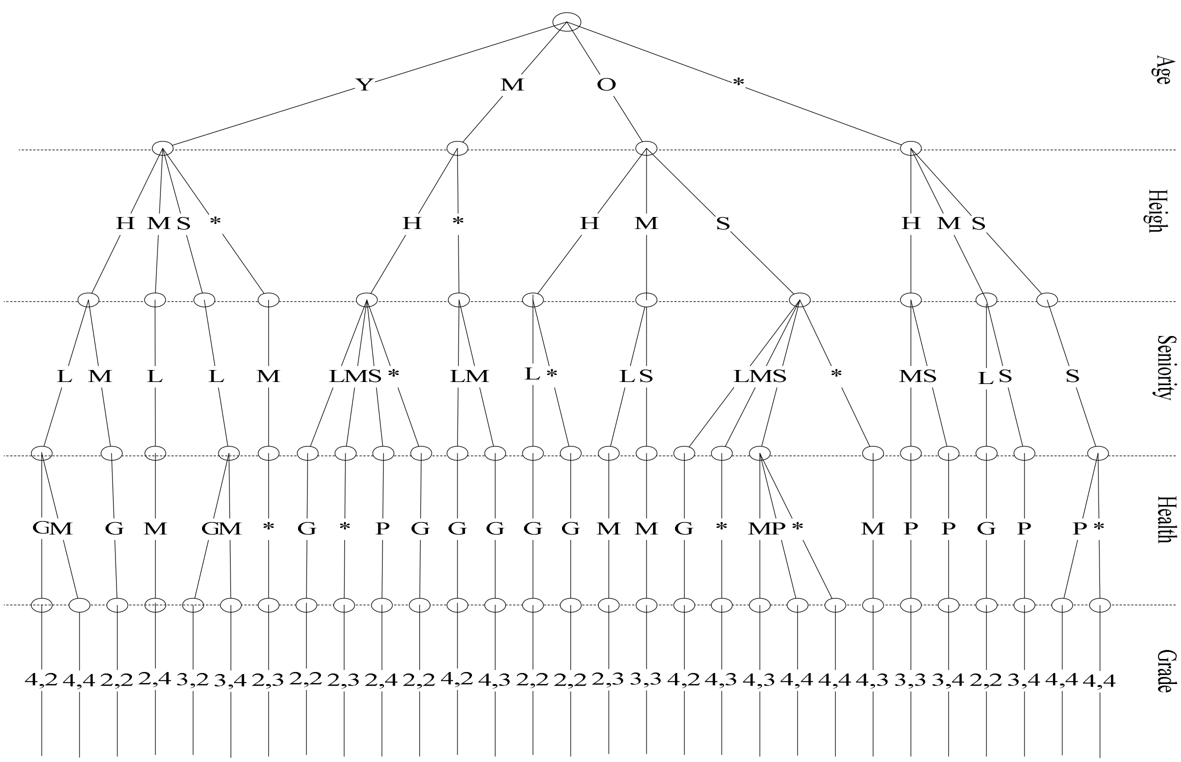

FIGURE 5. Transformation result of the objective rule trees

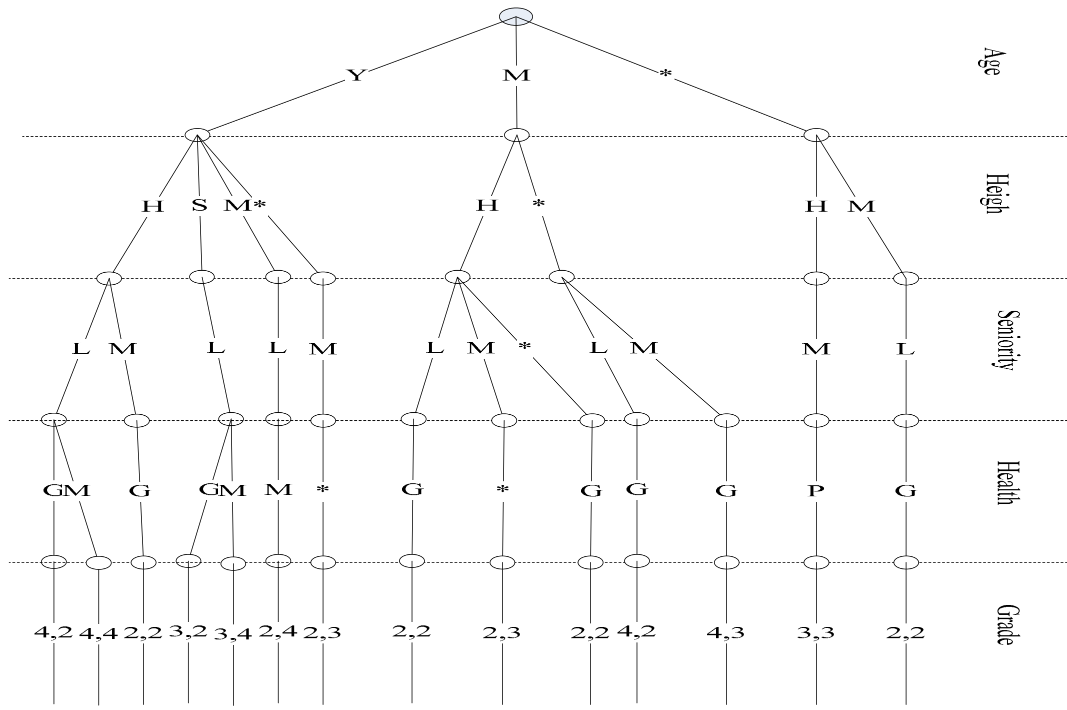

FIGURE 6. Combined objective rule trees after pruning by the constraint rules