International Journal of Innovative Computing, Information and Control
Volume 9, Number 5, May 2013
ICIC International ©2013 ISSN 1349-4198
pp. 2087–2098

RESEARCH ON THE THEORY AND METHOD OF ROUGH PROGRAMMING BASED ON SYNTHESIS EFFECT

Fachao Li[1], Ying Jing[1], Yan Shi[2,3] and Chenxia Jin[1]

[1]School of Economy and Management
Hebei University of Science and Technology
No. 70, Yuhua East Road, Shijiazhuang 050018, P. R. China
lifachao@tsinghua.org.cn; 932023729@qq.com; jinchenxia2005@126.com

[2]Graduate School of Science and Technology
Tokai University
9-1-1, Toroku, Kumamoto 862-8652, Japan
yshi@ktmail.tokai-u.jp

[3]School of Management
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116024, P. R. China

Received February 2012; revised June 2012

Abstract. *Constructing different rough programming models and obtaining their solving methods are very important for forming a reasonable decision making. In this paper, for the solving problem in rough environment, we first analyze the essential characteristics of rough programming systematically, and give the general expression of rough programming. Then we construct a rough programming model based on synthesis effect by distinguishing direct effect and indirect effect (denoted by BSE-RPM for short), and discuss its constructing procedures and strategies. Finally, we analyze the effectiveness and characteristics of BSE-RPM by a case-based example. The result indicates that this model has good interpretability, can integrate the decision preference into decision making process, and it extends and enriches the existing rough programming theories and methods.*

Keywords: Decision making, Rough programming, Effect synthesis function, Equivalence relation, Rough set, Model

1. **Introduction.** Rough set theory [1] is an effective tool for data mining, and it has been the core problem in academic and application fields. Many scholars gave many discussions under different background and also obtained many important research results. [2] expanded traditional rough set theory and gave variable precision rough set model; [3] studied the application of precision parameter in the variable precision rough set model; [4] further discussed the criteria for selecting parameter in variable precision rough set model; [5] studied the knowledge reduction method through adjusting variable precision; [6] presented a novel approach for mining association rules based on rough set theory; [7] used the rough sets based on tolerance relation to discuss attribute reduction in incomplete decision systems; [8] studied the attribute reduction and optimal decision making rules acquisition for continuous value information systems by using tolerance rough set model based on similarity of different objects; [9] proposed an attribute reduction method in the Bayesian Version of variable precision rough set model; [10] proposed a way to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets; [11] studied the attribute reduction with rough sets based on general binary relations; [12] proposed a heuristic algorithm depended on mutual information for reduction of

knowledge based on rough set theory; [13] analyzed characteristics of knowledge reduction systematically based on rough set theory and proposed decision table reduction based on conditional information entropy.

All results above mainly focus on the reduction of information system, and the researches on rough programming theories and methods are less discussed. As different kinds of uncertainties (like fuzziness, randomness and roughness) greatly exist in many decision making problems, such as production management, resource allocation, complex system optimization, many scholars gave many different discussions and also obtained many important uncertain programming theories and methods.

1) For stochastic uncertainty, [14] proposed chance-constrained programming model which had used threshold value of probability as a ploy to make constraints and objectives to be certain; [15] proposed dependent-chance programming model based on maximizing the probability of satisfying the constraints; [16] studied the random assignment problem by combining with genetic algorithm, and it used risk critical value as a strategy to process stochastic information.

2) For fuzzy uncertainty, [17] discussed the solving problem for fuzzy linear programming by using interval-valued to describe fuzzy information; [18] proposed a robust multi-criteria fuzzy programming problem by using possibility measure to measure uncertainty.

3) For rough uncertainty, [19] first proposed rough programming problem, presented the concept of rough optimal solution by using approximations and distance function as measure method, and further studied the roughness of optimal solution. But the above discussions merely considered the roughness of solution, and did not consider the role of equivalence classes and their effect on decision making; [20] proposed inexact rough-interval two-stage stochastic programming model by using rough-interval to depict definite and probable variation ranges of complex parameters. But this model merely regarded intervals as a way of describing roughness, and did not systematically analyze the feature of rough programming; [21] proposed a rough set-based multiple criteria linear programming approach to solve classification and prediction problems more effectively in data mining. But rough set is only used as a prefixing part which is responsible for data preprocessing, and the structure of rough programming was not discussed in this paper; [22] classified rough programming problems into three kinds according to the features of feasible set and objective function, and discussed the solving method of rough programming with a rough feasible set and a crisp objective function. However, the essence is to transform rough programming into crisp programming by changing feasible set into its lower or upper approximations, and it did not think about the effect of equivalence classes.

For the decision problem with several uncertainties, some scholars proposed and studied different rough programming problems in theory. [23] presented a concept of rough degree of membership, and proposed rough multiple objective programming by using rough degree of membership to process uncertain constraints. [24] presented a concept of rough interval and proposed an inexact rough-interval fuzzy linear programming model. [25] discussed a class of programming problem with random parameters and proposed rough chance-constrained programming model by transforming uncertain feasible set into lower and upper approximations based on probability. [26] presented concepts of random rough coefficients and fuzzy rough coefficients, and analyzed their structural characteristics. [27,28] studied a class of multi-objective programming problems with random rough coefficients. [29] studied a class of multi-objective decision making problem with fuzzy rough coefficients.

These achievements above basically reflect the research results of rough programming, while those only took a small part on stochastic programming and fuzzy programming.

Compared with the successful application of rough set theory in data mining, the researches on rough programming are at the starting stage. The shortages of these achievements are stated as follows. 1) They lack simple description form on rough programming model. 2) They only rely on lower and upper approximations as processing strategy of roughness, and did not think about the role of equivalence classes; also they lack systematic. And the two facts above must be considered in building solving model, so the relative researches are very important in theory and practice.

Based on the analysis above, in this paper, for the formal description of rough programming and model construction, we have the following work: 1) we analyze the essential characteristics of rough programming problem combined with the reality, and present its general descriptive form; 2) we put forward the concept of direct effect and indirect effect, and then propose the rough programming model based on synthesis effect (denoted by BSE-RPM for short); 3) we further analyze the basic constructing idea and procedures of BSE-RPM by using inclusion degree as measurement pattern of indirect effect; 4) we analyze the characteristics and effectiveness of BSE-RPM through a concrete medical case.

2. **Preliminaries.**

Relation (a subset of $U \times U$ is called a relation on U) is a common method describing the relationship between different things, and it is often used in decision making process. In 1984, based on an equivalence relation on U, Pawlak put forward the concept of rough set and constructed the basic frame of rough set theory. It is an effective method dealing with uncertain information in decision making process.

Definition 2.1. *(see [1]) Let U be a finite set, R be an equivalence relation on U, $[x]_R = \{y|\,(x,y) \in R\}$ be the R-equivalence class of x, and*

$$\underline{R}(X) = \{x|\, x \in U, \text{ and } [x]_R \subset X\}, \tag{1}$$

$$\overline{R}(X) = \{x|\, x \in U, \text{ and } [x]_R \cap X \neq \emptyset\}. \tag{2}$$

If $\underline{R}(X) = \overline{R}(X)$, then X is called R-exact set, otherwise X is called R-rough set. And $\underline{R}(X)$ is called R-lower approximation set of X, $\overline{R}(X)$ is called R-upper approximation set of X.

Obviously, with respect to R, the elements in $\underline{R}(X)$ surely belong to X, the elements in $U - \overline{R}(X)$ surely do not belong to X, and the elements in $\overline{R}(X) - \underline{R}(X)$ possibly belong to X. So X can be approximately described by $(\underline{R}(X), \overline{R}(X))$. For simplicity, $\mathrm{Pos}_R(X) = \underline{R}(X)$ is called R-positive region of X, $\mathrm{Neg}_R(X) = U - \overline{R}(X)$ is called R-negative domain of X, $\mathrm{Bn}_R(X) = \overline{R}(X) - \underline{R}(X)$ is called R-boundary of X, and

$$\alpha_R = \frac{|\underline{R}(X)|}{|\overline{R}(X)|} \tag{3}$$

is called R-approximate accuracy of X. Here, $X \neq \emptyset$, $|A|$ is used to express the number of elements in A.

As a metric describing the rough property of X, R-approximate accuracy of X denotes the inaccuracy of selecting decision plan based on approximation space (U, R). Equivalence class is the basic factor for decision making, so the relation between X and equivalence classes restricts the selection of decision schemes. For convenience,

$$C_R(x) = \frac{|[x]_R \cap X|}{|[x]_R|} \tag{4}$$

is called R-believable degree of x according with X. It is easy to see that $C_R(x)$ reflects the degree of X including equivalence class $[x]_R$, and describes the roughness of x with respect to X in approximation space (U, R) from partial views.

Rough set has many good operational properties which are discussed in [30].

3. **Analysis on Essential Characteristic of Rough Programming.** The core of programming problem is to seek a decision scheme to make the objective attain optimum in a certain region. And its general form is as follows:

$$\begin{cases} \max f(x), \\ \text{s.t. } x \in X. \end{cases} \tag{5}$$

Here, X is a subset of U (called feasible region), and $f(x)$ is a function with quantitative characteristics on U (called objective function).

Based on the different characteristics of feasible region and objective function, (5) can be thought as certain programming problem (X is a crisp subset and $f(x)$ is a real-valued function on U) and uncertain programming problem (X or $f(x)$ has uncertainties). Especially, if X is a fuzzy set or $f(x)$ has fuzziness, (5) is called a fuzzy programming problem; if X or $f(x)$ has randomness, (5) is called a stochastic programming problem.

In real decision making process, we not only consider (5), but also care about the relation between elements of U. For instance:

Case 1. In order to develop the new market well, a company wants to choose a person in charge of developing the business there. The prime work involves many things in all fields, so when choosing a person in charge, we must take specialty advantages, social relationship, and working ability into consideration. If we regard X as the set of candidates, U as the set of people who have some special relationship R with the staff in this company, $f(x)$ as comprehensive measurement of specialty advantages and working ability of candidates, $[x]_R$ as the set of people who have relationship R with x (means potential available resources of x), this problem is a decision making problem with form (5) related to R.

Case 2. In order to strengthen market competitiveness, enterprise wants to choose an employee who is responsible for technical innovation. Development of this item involves substantive cooperation. So when choosing the employee, we must take specialty advantages, working ability, coordinate ability, physical condition and team composition into consideration, and we also need to guarantee project success and protect core technology. If we regard X as the set of candidates, U as the set of people who have close cooperative research relationship R with the staff in this enterprise, $f(x)$ as comprehensive measurement of specialty advantages, working ability, coordinate ability, physical condition of candidates, $[x]_R$ as the set of people who have relationship R with x (means potential available resources of x), this problem is a decision making problem with form (5) related to R.

Case 3. In order to improve scientific research level of staff, a scientific research institution plans to choose a high-performance calculation software. And there are 5 candidates. The research target and direction of staff are different, so when choosing a product, we must take its performance and the compatibility with other calculation software into consideration. If we regard x as the set of products prepared for choosing, U as the set of related softwares, $f(x)$ as performance measurement of products, $[x]_R$ as the set of softwares which have relationship R with x (means potential application range of x), this problem is a decision making problem with form (5) related to R.

The decision making problems with the above characteristics exist widely in the fields of equipment renewal, resource management, complex system optimization and so on. And

$[x]_R$ has different effect on different decision making problems. For instance, the effect of $[x]_R$ is bigger, it is better in Case 1, the degree of X including $[x]_R$ is bigger, it is better in Case 2 (that is, the more staff of $[x]_R$ belong to our company, the more the company will have security). And the number of elements of $[x]_R$ is more, it is better in Case 3 (that is, the more compatibility, the higher utilization). All the decision making process above involves all elements which have relationship R with X (R-upper approximation of X). Owing to that R-upper (lower) approximation is foundation of rough set theory, this programming is called rough programming and can be formulated as follows:

$$\begin{cases} \max f(x), \\ \text{s.t. } x \in (X, R). \end{cases} \tag{6}$$

Here, U is the universe and $X \subseteq U$; $f(x)$ is a function with some quantitative characteristics on U, and it is a quantitative index which reflects scheme x is good or not (called ***direct effect*** of scheme x); R is a relationship on U; $x \in (X, R)$ means x and X are interrelated with respect to R.

By the value of $f(x)$, (6) can be divided into certain rough programming (CRP) and uncertain rough programming (URP). For URP: 1) If $f(x)$ is a rough-valued (rough variable on real number field $\mathbb{R}$) function, then it is called a rough rough programming; 2) If $f(x)$ is a random-valued (random variable on real number field $\mathbb{R}$) function, then it is called a random rough programming; 3) If $f(x)$ is a fuzzy-valued (fuzzy number on real number field $\mathbb{R}$) function, then it is called a fuzzy rough programming. Easily to know, using the characteristic of objective and constraints to distinguish the properties of programming can be helpful to use related theory and method to build solving model. So (6) has wide generality and interpretability.

For convenience, in what follows, we suppose that X is a crisp subset of U, $f(x)$ is a nonnegative real-valued function on U and R is an equivalence relation on U.

The set $[x]_R$ which has relationship R with x always has indirect effect on the performance of x (called ***indirect effect*** of scheme x). So when we make a decision, we should consider direct effect and indirect effect at the same time. If the indirect effect of equivalence class is interpreted as a mapping G from $U/R = \{[x]_R | \, x \in U\}$ to $[0, 1]$ (called ***indirect effect function***), the synthesis problem of direct effect and indirect effect can be abstracted as a mapping $S(u, v)$ from $[0, \infty) \times [0, 1]$ to $[0, \infty)$ (where u denotes $f(x)$ and v denotes $G([x]_R)$), $S(u, v)$ is called ***effect synthesis function***, and it satisfies: 1) for any given u or v, $S(u, v)$ is non-decreasing; 2) $S(u, 1)$ is strictly monotone increasing), then model (6) can be transformed into model (7):

$$\begin{cases} \max S(f(x), G([x]_R)), \\ \text{s.t. } x \in \overline{R}(X). \end{cases} \tag{7}$$

It is easy to see when $U/R = \{U\}$ (means all elements of U are the same type or the feature of classes are not considered) and $G(U) = 1$, the decision results of (7) and (5) are same for any $S(u, v)$, this indicates model (7) is an extension of classical programming model (5). Considering that the measure models of indirect effect and effect synthesis operator are the core of model (7), so we call (7) ***rough programming model based on synthesis effect*** (denoted by BSE-RPM for short) in the following. It is easy to prove that

$$S_1(u, v) = u(1 + kv^\alpha), \;\; 0 \le k \le 1, \; 0 \le \alpha < \infty, \tag{8}$$

$$S_2(u, v) = u \ln(1 + kv), \;\; 0 \le k < \infty, \tag{9}$$

$$S_3(u, v) = ue^{kv}, \;\; 0 \le k < \infty \tag{10}$$

are all effect synthesis functions.

Remark 3.1. *When $f(x)$ has some uncertainties, we can centralizedly quantify it into a real value $T(f(x))$ by some strategies. For example, when $f(x)$ has random uncertainty, we can describe it by its expectation $E(f(x))$ and variance $D(f(x))$, and further combine some suitable synthesis operator $Z(u,v)$ to integrate $E(f(x))$ and $D(f(x))$ into a value $T(f(x)) = Z(E(f(x)), D(f(x)))$. Then we can get the solution model of URP by using $T(f(x))$ instead of $f(x)$ in (7). So (7) is a solving model with strong operability.*

Remark 3.2. *If $S(u,v) = uv$, then: 1) when $G([x]_R) = 1$ for $[x]_R \subset X$, and $G([x]_R) = 0$ for $[x]_R \not\subset X$, (7) is the solution model based on lower approximation in [22]; 2) when $G([x]_R) = 1$ for $[x]_R \bigcap X \neq \emptyset$, and $G([x]_R) = 0$ for $[x]_R \bigcap X = \emptyset$, (7) is the solution model based on upper approximation in [22].*

Remark 3.3. *According to the Remark 4.2 in [31], we can get the rough chance-constrained programming model in [25] by model (7) under appropriate effect synthesis operator.*

Theorem 3.1. *Let X be a non-single-point set, and x^* is the optimal solution to (5). Then for any effect synthesis function $S(u,v)$, x^* is also the optimal solution to (6) if and only if $G([x^*]_R) = \max\{G([x]_R) | x \in X\}$.*

Proof: If $G([x^*]_R) = \max\{G([x]_R) \mid x \in X\}$, then $f(x) \leq f(x^*)$. Considering the monotonicity of $S(u,v)$, we know that $S(f(x), G([x]_R)) \leq S(f(x^*), G([x^*]_R))$ for any $x \in X$. So x^* is the optimal solution to (6).

Conversely, if $G([x^*]_R) \neq \max\{G([x]_R) \mid x \in X\}$, then there must exist an effect synthesis function $S(u,v)$ and $x_0 \in X$ such that $S(f(x^*), G([x^*]_R)) < S(f(x_0), G([x_0]_R))$. So x^* is not the optimal solution to (6) (in fact, by X is non-single-point set, we can know that there must exist $x_0 \in X$ such that $G([x_0]_R) > G([x^*]_R)$. For $S(u,v) = u + kv$, $k \in [0, \infty)$, when $k > (f(x^*) - f(x_0))/(G([x_0]_R) - G([x^*]_R))$, we have $S(f(x^*), G([x^*]_R)) < S(f(x_0), G([x_0]_R)))$, this is in contradiction with condition.

The analysis above shows that rough programming can be regarded as a synthesis of common programming and a relationship on U. With different synthesis operators, rough programming can be transformed into different common ones.

4. Effect Metric Model Based on Inclusion Degree.

In Section 3, combine with several typical decision making problems, we give the general form of rough programming model, and establish rough programming model based on synthesis effect. The indirect effect of equivalence classes in different problems has different meanings and expressions, and even this difference is big (such as Case 2 and Case 3 in Section 3). So the key to solve rough programming problems is to construct metric pattern of indirect effect under different background. In this section, we will take that whether the equivalence class meets decision making requirements or not (or the degree of equivalence class meeting decision making requirements) as basic metric criterion of indirect effect, and further discuss the construction of rough programming model based on synthesis effect.

$D(A \subset B) = |A \cap B|/|A|$, with good visualization and strong operability, is an effective method measuring the degree of B including A, so $D([x]_R \subset X)$ describe the degree of $[x]_R$ according with X (that is, the R-believable degree $C_R(x)$ in Section 2). And its value reflects the potential value of x. And for the Case 2 in Section 3, $C_R(x)$ can be considered as the degree of team members belonging to the company. $C_R(x) = 1$ means team members belong to the company totally. That the value of $C_R(x)$ is bigger is good for protecting core technology. If we use $C_R(x)$ to measure indirect effect of $[x]_R$, model (7) can be transformed into (11):

$$\begin{cases} \max S(f(x), C_R(x)), \\ \text{s.t. } x \in \overline{R}(X). \end{cases} \tag{11}$$

It is worth noting that the optimal solution to (11) may not be in X. If we need optimal decision must be in X, we could use $G([x]_R) = C_R(x)$ for $x \in X$, and $G([x]_R) = 0$ for $x \notin X$ to measure the indirect effect of $[x]_R$, and use $S^*(u, v)$ as the effect synthesis function in (7) (here, $S^*(u, v) = S(u, v)$ for $v \neq 0$, and $S^*(u, v) = 0$ for $v = 0$), and we can get (12):

$$\begin{cases} \max S(f(x), C_R(x)), \\ \text{s.t. } x \in X. \end{cases} \tag{12}$$

Especially, if $S(u, v) = u$ for $v \geq \beta$, and $S(u, v) = 0$ for $v < \beta$, then model (12) can be shown as:

$$\begin{cases} \max f(x), \\ \text{s.t. } x \in X, \text{ and } C_R(x) \geq \beta. \end{cases} \tag{13}$$

Here, $\beta \in [0, 1]$ means the lowest R-believable degree of decision objects. If $S(u, v) = u(1 + 0.8v^2)$, then model (12) can be shown as:

$$\begin{cases} \max f(x)(1 + 0.8(C_R(x))^2), \\ \text{s.t. } x \in X. \end{cases} \tag{14}$$

The analysis above shows that rough programming can be transformed into specific programming problem by choosing metric pattern of indirect effect $G([x]_R)$ and synthesis effect operator $S(u, v)$. $G([x]_R)$ can be considered as a supplement for scheme x and it is an evidence reflecting credibility of $f(x)$. $S(u, v)$ can be considered as a further diagnosis and correction of scheme x, different $S(u, v)$ reflects different processing strategies. For instance, if $S(u, v) = u(1 + kv^\alpha)$, kv^α means acting factor of indirect effect, k and α are parameters reflecting the importance of indirect effect, and their characteristics are stated as follows: 1) k is a parameter reflecting importance of indirect effect from the whole. That k is bigger means decision makers pay much more attention to indirect effect, and $k = 0$ means indirect effect is not considered; 2) α is a parameter reflecting the importance of indirect effect concretely (the visual explanation of kv^α is shown in Figure 1). When $0 \leq \alpha < 1$ and α is smaller means decision makers pay less attention to indirect effect (especially, when $\alpha = 0$, indirect effect have no new effect on decision result). When $\alpha \geq 1$ and α is bigger means decision makers pay more attention to the decision scheme which has bigger indirect effect (namely, the decision value of decision scheme will change only if the indirect effect is bigger. Especially, when $\alpha \to \infty$, decision value of decision scheme will change only if indirect effect is 1). When $S(u, v) = u(1 + kv^\alpha)$, if α is too big or too small, it will not reflect the role of indirect effect effectively. In practice, we should choose the value of k and α by considering specific decision preference. In general, $k \in [1, 5]$ and $\alpha \in [0.5, 2]$.

From what have discussed above, we can construct rough programming model based on synthesis effect according to the following steps:

Step 1 Determine the metric pattern of indirect effect, that is, determine a mapping G from U/R to $[0, 1]$, by combining with the influence that the relationship R has on decision making.

Step 2 Choose the synthesis operator of direct effect and indirect effect, then we can obtain the decision making model of rough programming according to (7).

5. **An Application of BSE-RPM in Making Scheme of Taking Drug.** In this section, we will further analyze the feature and performance of BSE-RPM by combining with a problem about making scheme of taking drug.

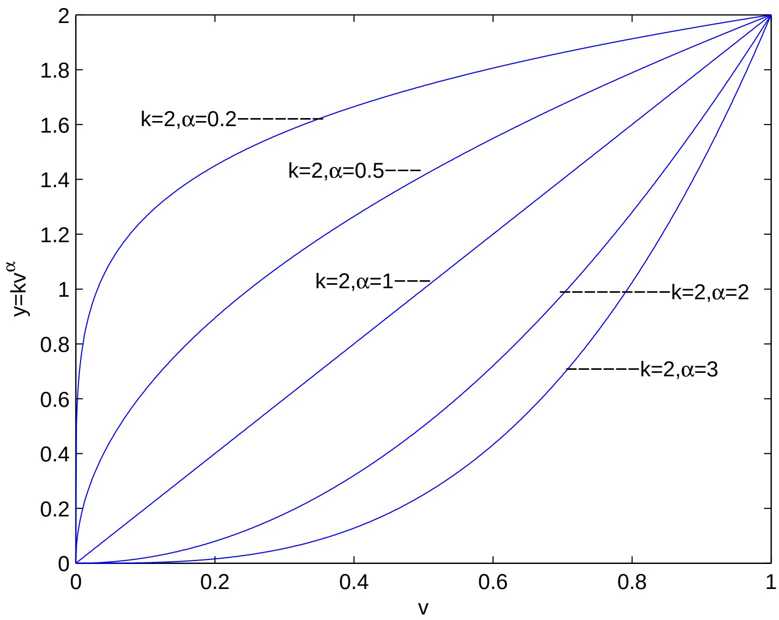

FIGURE 1. The explanation for acting factor of indirect effect

5.1. **Case description.** The drug D was developed to treat cancer of the liver. It could kill cancer cells and inhibit the growth and reproduction of them, but also injure normal human cells at the same time. Normal human cells which ensure normal function of human body have small proportion in total cells (such as leukocytes, lymphocyte, bone marrow cells and platelets), but when they are injured, may decrease immune function, hematopoietic function and liver function. If it is going too bad, may lead to heart failure. The clinical test showed that: 1) Using the drug for one course implies to kill 30% of cancer cells; 2) Using the drug for two courses implies to kill 50% of cancer cells and 0.1% of normal cells; 3) Using the drug for three courses implies to kill 70% of cancer cells and 0.3% of normal cells; 4) Using the drug for four courses implies to kill 90% of cancer cells and 0.5% of normal cells; 5) Using the drug for five courses implies to kill 100% of cancer cells and 0.8% of normal cells. Try to select therapeutic scheme.

5.2. **Analysis of the solution process.** The main purpose of this therapeutic scheme is to kill cancer cells. So if $f(x)$ means the satisfaction degree of killing cancer cells x and we only consider killing cancer cells, making the therapeutic scheme can be generalized into the following programming problem:

$$\begin{cases} \max f(x), \\ \text{s.t. } x \in X. \end{cases} \tag{15}$$

Here, U means all of cells including cancer cells and normal human cells, X means all of liver cancer cells, and $f(x)$ is shown in Table 1 (in Table 1, X_i is the cancer cells killed in ith course):

TABLE 1. The value of $f(x)$

x	$x \in X_1$	$x \in X_2$	$x \in X_3$	$x \in X_4$	$x \in X_5$
$f(x)$	0.3	0.5	0.7	0.9	1.0

However, it is worth noting that when cancer cells are killed, the normal human cells are also killed at the same time from the second course. When the damage rate of normal cells is low, body can adjust and maintain the normal function. However, higher mortality rate of the normal cells may lead to the failure of body functions. So when we design treatment programs, we must consider killing cancer cells and normal cells. Define an equivalence relation R on U as:

$$R = \left\{ (x, y) \middle| \begin{array}{l} x \text{ and } y \text{ satisfy one of following conditions: 1) cells killed} \\ \text{in the same course; 2) cells were not killed after treatment} \end{array} \right\}. \tag{16}$$

The defined equivalence relation R makes a partition to U as $E_i = \{x | \, x \text{ is killed in the } i\text{th course}\}$, $i = 1, 2, 3, 4, 5$, $E_6 = \{x | \, x \text{ is not killed after treatment}\}$. Then the essence of making therapeutic scheme is to solve the following rough programming problem:

$$\left\{ \begin{array}{l} \max f(x), \\ \text{s.t. } x \in (X, R). \end{array} \right. \tag{17}$$

During the treatment, the less the quantity of normal human cells killed is, the better the effect is, so we measure indirect effect of equivalence class E_i by the approaching degree between total normal cells be killed from the first to the ith course Y_i and the damage limit (when damage is larger than 1% may lead to heart failure), that is, $G(E_i) = 1 - Y_i$. And the indirect effect values of each equivalence class were shown in Table 2.

Based on the discussion in Section 4, if taking $S(u, v) = u(1 + kv^{\alpha})$ as the synthesis operator of direct effect and indirect effect, then we can get the following common programming problem (18):

$$\left\{ \begin{array}{l} \max f(x)(1 + k(G([x]_R))^{\alpha}), \\ \text{s.t. } x \in X. \end{array} \right. \tag{18}$$

By this we can select the therapeutic scheme according to the specific situation of patients. The decision results of different parameters are shown in Table 3.

From Table 3, we can see that the optimization result is closely related to the selection of synthesis operator. For this case and the effect synthesis model $S(u, v) = u(1 + kv^{\alpha})$, we can get the decision results as follows: 1) when $k = 2.5$ and $\alpha = 2.0$, the optimal solution $x^* \in E_3$, this implies that the patient should continuously take drug D for three courses; 2) when $k = 2.5$ and $\alpha = 1.0$, the optimal solution $x^* \in E_4$, this implies that the patient should continuously take drug D for four courses; 3) when $k = 0.5$ and $\alpha = 0.5$, the optimal solution $x^* \in E_5$, this implies that the patient should continuously take drug D for five courses; 4) when $k = 0.5$ and $\alpha = 1.0$, the optimal solution $x^* \in E_4$, this implies that the patient should continuously take drug D for four courses; 5) the change of α has a obvious influence on decision result, while the change of k has weak influence.

Combining the discussion in Section 4, if we regard k and α as the parameters reflecting decision preference, then all the above discussions show that BSE-RPM can integrate decision preference into decision making process effectively by a quantitative way. And they can be embodied by the following two forms: 1) if the patient's enginery state is good, we should reduce the importance of indirect effect, and select relatively smaller α

TABLE 2. The indirect effect values of every equivalence class

$[x]_R$	E_1	E_2	E_3	E_4	E_5
The total number of normal cells are killed (%)	0	0.1	0.3	0.5	0.8
$G([x]_R)$	1	0.9	0.7	0.5	0.2

Table 3. The decision results of different parameters

x			$x \in E_1$	$x \in E_2$	$x \in E_3$	$x \in E_4$	$x \in E_5$
$f(x)$			0.3	0.5	0.7	0.9	1.0
$G([x]_R)$			1.0	0.9	0.7	0.5	0.2
$S(f(x), G([x]_R))$	$k = 0.5$	$\alpha = 0.5$	0.4500	0.7372	0.9928	1.2182	1.2236
		$\alpha = 1.0$	0.4500	0.7250	0.9450	1.1250	1.1000
		$\alpha = 1.5$	0.4500	0.7135	0.9050	1.0591	1.0447
		$\alpha = 2.0$	0.4500	0.7025	0.8715	1.0125	1.0200
	$k = 1.0$	$\alpha = 0.5$	0.6000	0.9743	1.2857	1.5363	1.4472
		$\alpha = 1.0$	0.6000	0.9500	1.1900	1.3500	1.2000
		$\alpha = 1.5$	0.6000	0.9269	1.1099	1.2182	1.0894
		$\alpha = 2.0$	0.6000	0.9050	1.0430	1.1250	1.0400
	$k = 1.5$	$\alpha = 0.5$	0.7500	1.2115	1.5785	1.8546	1.6708
		$\alpha = 1.0$	0.7500	1.1750	1.4350	1.5750	1.3000
		$\alpha = 1.5$	0.7500	1.1403	1.3149	1.3773	1.1342
		$\alpha = 2.0$	0.7500	1.1075	1.2145	1.2375	1.0600
	$k = 2.0$	$\alpha = 0.5$	0.9000	1.4487	1.8713	2.1727	1.8944
		$\alpha = 1.0$	0.9000	1.4000	1.6800	1.8000	1.4000
		$\alpha = 1.5$	0.9000	1.3538	1.5199	1.5364	1.1789
		$\alpha = 2.0$	0.9000	1.3100	1.3860	1.3500	1.0800
	$k = 2.5$	$\alpha = 0.5$	1.0500	1.6859	2.1641	2.4910	2.1180
		$\alpha = 1.0$	1.0500	1.6250	1.9250	2.0250	1.5000
		$\alpha = 1.5$	1.0500	1.5673	1.7249	1.6955	1.2236
		$\alpha = 2.0$	1.0500	1.5125	1.5575	1.4625	1.1000

(such as $\alpha \in [0.4, 0.8]$) and appropriate k (such as $k \in [1, 2]$); 2) if the patient's enginery state is not very well, we should improve the importance of indirect effect, and select relatively larger α (such as $\alpha \in [1, 2]$) and appropriate k (such as $k \in [2, 3]$).

6. **Conclusion.** In this paper, combining with concrete background, we analyze the essential characteristics of rough programming, give the general expression of rough programming, and establish a rough programming model based on synthesis effect (BSE-RPM). Then we discuss the constructing strategies of BSE-RPM, and analyze the performance of BSE-RPM by a case-based example. As shown from theoretical analysis and experimental results, the study in this paper has the following advantages compared with the present researches. 1) The expression of rough programming has good structural properties and inclusiveness; these can help us to establish suitable solving method by combining with the properties of the problem. 2) We first consider the role of equivalence classes in decision making, which provides a useful tool for processing the roughness with different ideas. 3) BSE-RPM has good interpretability and structural characteristics, it not only contains the present methods, but also intergrates decision preference into the decision making process effectively. So the discussions in this paper enrich the existing programming theories, and it can be widely used in many fields such as resource management, system decision and artificial intelligence.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (71071049, 71210107002) and the Natural Science Foundation of Hebei Province (F2011208056).

REFERENCES

[1] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences*, vol.11, no.5, pp.341-356, 1982.

[2] W. Ziarko, Variable precision rough set model, *Journal of Computer and System Sciences*, vol.46, no.1, pp.39-59, 1993.

[3] C. T. Su and J. H. Hsu, Precision parameter in the variable precision rough sets model: An application, *Omega: International Journal of Management Science*, vol.34, no.2, pp.149-157, 2006.

[4] M. Beynon, Reducts within the variable precision rough sets model: A further investigation, *European Journal of Operational Research*, vol.134, no.3, pp.592-605, 2001.

[5] J. S. Mi, W. Z. Wu and W. X. Zhang, Approaches to knowledge reduction based on variable precision rough set model, *Information Sciences*, vol.159, no.3, pp.255-272, 2004.

[6] C. B. He and Q. M. Chen, Approach for mining association rules based on rough set, *Journal of Computer Applications*, vol.30, no.1, pp.25-28, 2010.

[7] Z. Q. Meng and Z. Z. Shi, A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets, *Information Sciences*, vol.179, no.16, pp.2774-2793, 2009.

[8] Y. Y. Guan, H. K. Wang, Y. Wang and F. Yang, Attribute reduction and optimal decision rules acquisition for continuous valued information systems, *Information Sciences*, vol.179, no.17, pp.2974-2984, 2009.

[9] D. Slezak and W. Ziarko, Attribute reduction in the Bayesian Version of variable precision rough set model, *Electronic Notes in Theoretical Computer Science*, vol.82, no.4, pp.263-273, 2003.

[10] D. G. Chen, C. Z. Wang and Q. H. Hu, A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets, *Information Sciences*, vol.177, no.17, pp.3500-3518, 2007.

[11] C. Z. Wang, C. X. Wu and D. G. Chen, A systematic study on attribute reduction with rough sets based on general binary relations, *Information Sciences*, vol.178, no.9, pp.2237-2261, 2008.

[12] D. Q. Miao and G. R. Hu, A heuristic algorithm for reduction of knowledge, *Journal of Computer Research and Development*, vol.36, no.6, pp.681-684, 1999.

[13] G. Y. Wang, H. Yu and D. C. Yang, Decision table reduction based on conditional information entropy, *Chinese Journal of Computers*, vol.25, no.7, pp.759-766, 2002.

[14] A. Charnes and W. W. Cooper, Chance-constrained programming, *Management Science*, vol.6, no.1, pp.73-79, 1959.

[15] B. D. Liu, Dependent-chance programming: A class of stochastic optimization, *Computers and Mathematics with Applications*, vol.34, no.12, pp.89-104, 1997.

[16] F. Li, C. X. Jin and Y. Shi, Study on solution models and methods for random assignment problem, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.359-368, 2009.

[17] T.-S. Shih, J.-S. Su and J.-S. Yao, Fuzzy linear programming based on interval-valued fuzzy sets, *International Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2081-2090, 2009.

[18] T. Hasuike and H. Ishii, A robust multi-criteria fuzzy programming problem considering possibility measure, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4907-4914, 2009.

[19] A. Ebrahim, Characterizing solutions of rough programming problems, *European Journal of Operational Research*, vol.168, no.3, pp.1019-1029, 2006.

[20] H. W. Lu, G. H. Huang and L. He, Inexact rough-interval two-stage stochastic programming for conjunctive water allocation problems, *Journal of Environmental Management*, vol.91, no.1, pp.261-269, 2009.

[21] Z. W. Zhang, Y. Shi and G. X. Gao, A rough set-based multiple criteria linear programming approach for the medical diagnosis and prognosis, *Expert Systems with Applications*, vol.36, no.5, pp.8932-8937, 2009.

[22] M. S. Osman, E. F. Lashein, E. A. Youness and T. E. M. Atteya, Mathematical programming in rough environment, *Optimization*, vol.60, no.5, pp.603-611, 2011.

[23] Z. M. Tao and J. P. Xu, A class of rough multiple objective programming and its application to solid transportation problem, *Information Sciences*, vol.188, no.2, pp.215-235, 2012.

[24] H. W. Lu, G. H. Huang and L. He, An inexact rough-interval fuzzy linear programming method for generating conjunctive water-allocation strategies to agricultural irrigation systems, *Applied Mathematical Modelling*, vol.35, no.9, pp.4330-4340, 2011.

[25] Y. Shi, L. M. Yao and J. P. Xu, A probability maximization model based on rough approximation and its application to the inventory problem, *International Journal of Approximate Reasoning*, vol.52, no.2, pp.261-281, 2011.
[26] B. D. Liu, *Theory and Practice of Uncertain Programming*, Physica Verlag, New York, 2002.
[27] J. P. Xu and L. M. Yao, A class of multiobjective linear programming models with random rough coefficients, *Mathematical and Computer Modelling*, vol.49, no.1-2, pp.189-206, 2009.
[28] J. P. Xu and L. M. Yao, A class of expected value multi-objective programming problems with random rough coefficients, *Mathematical and Computer Modelling*, vol.50, no.1-2, pp.141-158, 2009.
[29] J. P. Xu and L. H. Zhao, A class of fuzzy rough expected value multi-objective decision making model and its application to inventory problems, *Computers and Mathematics with Applications*, vol.56, no.8, pp.2107-2119, 2008.
[30] W. X. Zhang and W. Z. Wu, *Theory and Method of Rough Sets*, Science Press, Beijing, 2001.
[31] C. Jin, F. Li, Y. Shi and L. Zhou, Study on stochastic programming methods based on synthesis effect, *International Journal of Innovative Computing, Information and Control*, vol.8, no.10(A), pp.6837-6848, 2012.