International Journal of Computational Intelligence Systems, Vol. 7, No. 4 (August 2014), 696-705

A Synthesizing Effect-Based Solution Method for Stochastic Rough Multi-objective Programming Problems

Lei Zhou[*]
School of Electrical Engineering and Automation, Tianjin University, Weijin Road No. 92, Tianjin, 300072, P. R. China
School of Economy and Management, Hebei University of Science and Technology, Yuhua Road No. 70, Shijiazhuang 050018, P. R. China
E-mail: zhoulei19@126.com

Guoshan Zhang
School of Electrical Engineering and Automation, Tianjin University, Weijin Road No. 92, Tianjin, 300072, P. R. China
E-mail: zhanggs@tju.edu.cn

Fachao Li
School of Economy and Management, Hebei University of Science and Technology, Yuhua Road No. 70, Shijiazhuang 050018, P. R. China
E-mail: lifachao@tsinghua.org.cn

Received 20 January 2013

Accepted 17 May 2013

Abstract

Multi-objective programming with uncertain information has been widely applied in modeling of industrial produce and logistic distribution problems. Usually the expectation value model and chance-constrained model as solution models are used to deal with such uncertain programming. In this paper, we consider the uncertain programming problem which contains random information and rough information and is hard to be solved. A new solution model, called stochastic rough multi-objective synthesis effect (MOSE) model, is developed to deal with a class of multi-objective programming problems with random rough coefficients. The MOSE model contains expectation value model and chance-constrained model by choosing different synthesis effect functions and can be considered as an extension of crisp multi-objective programming model. Combined with genetic algorithm, the optimal solution of the MOSE model can be obtained. It shows that the solutions of the MOSE model are better than that of other solution models. Finally, an illustrative example is provided to show the effectiveness of the proposed method.

Keywords: Multi-objective Programming; Random rough variable; Stochastic Programming; Genetic algorithm; Synthesis effect.

1. Introduction

Optimization method is widely applied in solving industrial production and logistic distribution problems. It contains single objective optimization and multi-objective optimization. If the decision maker takes only one criterion into consideration, it is a single objective optimization problem, which has been well studied for the past 50 years, and if decision maker considers more than one criterion simultaneously, it is a multi-objective optimization problem, which is widely investigated recently.

Many researchers have paid attention to the multi-objective optimization problem. Jin et al.[1] proposed evolutionary game theory, which transforms the optimization problem into game strategic problem, and obtained the optimized strategy by using adaptable dynamic game evolution process intelligently. Zhang et al.[2] proposed a scalable cooperative co-evolution and ε-dominance based multi-objective particle swarm

[*]Corresponding author

optimizers. Optimization method is usually to seek a solution over a set of possibility by certain criteria.

An uncertain multi-objective programming with random information and rough information is a branch of multi-objective optimization. Randomness and roughness as two kinds of uncertainties are usually widespread and unavoidable in practice. Processing random information and rough information is very important in production management, artificial intelligence, complex systems optimization etc. A random rough variable can describe this class of uncertainty information. It can be regarded as a family of data satisfying some laws, and there is no order relation between every two variables. Therefore, the common programming methods are not suitable for solving stochastic rough programming problems.

Liu[3] introduced the concepts of rough variable and random rough variable, stochastic rough expected value model and stochastic rough chance-constrained model, which provided a useful method to deal with the random roughness. Chance function is the distribution state of values of the random rough variables and expectation is a weighted average of all values of the random rough variables**.** Xu et al.[4, 5] studied a class of multi-objective linear programming models with random rough coefficients and gave a solution method.

The above literatures removed the uncertainty of the random rough variable by chance function or expectation of the random rough variable and then gave programming models and solution methods. These ways did not comprehensively reflect the uncertainty information. Expectation model and chance-constrained model used on stochastic rough programming didn’t reflect the information and features of the original stochastic rough programming problem fully. We hope to use numerical characteristics to describe a random rough variable clearly, in order to make solution model more reasonable.

This paper is organized as follows. Section 2 introduces some related works. Section 3 introduces some definitions of random rough variables, and shows the expectation and variance and provides a stochastic rough multi-objective programming model. Section 4 proposes a stochastic rough multi-objective synthesis effect (MOSE) model, and gives the features of the MOSE model. The solution procedure is proposed in Section 5. Section 6 provides a simple numerical example to show the rationality of the MOSE model and effectiveness of the proposed method. We give the conclusions of this paper in Section 7.

2. Related Works

Stochastic multi-objective programming problem has been proposed many years ago. There are two kinds of solution methods of the multi-objective stochastic programming problems which are expectation model and chance-constrained model[6]. Some researchers paid attention to the application of the stochastic programming[7, 8, 9, 10]. Turgut et al.[11] presented a method to generate Pareto surface for multi-objective integer programs with stochastic coefficients in the objective functions based on minimum expectation and variance criteria. Charles et al.[12] provided an algorithm that identifies redundant objective functions and redundant constraints simultaneously in multi-objective nonlinear stochastic fractional programming problems. Abdelaziz[13] surveyed various solution approaches for multi-objective stochastic programming problems where random variables are both objectives and constraints parameters. Zhang et al.[14] proposed a fuzzy-robust stochastic multi-objective programming approach, which integrates fuzzy-robust linear programming and stochastic linear programming into a general multi-objective programming framework. Zhu et al.[15] introduced stochastic semi-definite programs and chance-constrained semi-definite programs as paradigms to deal with uncertainty in applications leading to semi-definite programs. Alzalg[16] described four application models leading to stochastic second-order cone programming.

Rough set theory can deal with the problems with inexact data or imprecise information for complicated systems effectively. Recently, many researchers pay attention to rough set with randomness. Yao[17] revisited probabilistic rough set approximation operators and presented a critical review of existing studies based on rough membership functions and rough inclusion functions. Rough approximation evaluative measures and one-way and two-way inter-set dependency measures were proposed by Ziarko[18] and adopted to probabilistic rule evaluation. Yao[19] provided an analysis of three-way decision rules in the probabilistic rough set model. Ma[20] presented the parameter dependence or the continuous of the lower and upper approximations on parameters for probabilistic rough set over two universes, and discussed the uncertainty measure of the knowledge granularity and rough entropy for probabilistic rough set over two universes by the proposed concept.

Based on rough set for the knowledge partition, rough programming is an equivalent classification method to determine the approximate domain of a given problem and to determine the decision making program. Youness[21] proposed the concept of a rough programming, and defined a surely optimal solution and a possibly optimal solution. By equivalent classification method of rough set, the global rough optimal solution and local rough optimal solution were defined to solve the rough programming problem. The concept of "rough interval" was introduced in the modeling framework to represent dual-uncertain parameters by Lu et al.[22].

The above methods proposed many ways to solve multi-objective programming problem with random information or rough information.

3. Preliminaries

Randomness and roughness often coexist in many practical problems. So the established model should contain random roughness. Single stochastic programming methods, rough set methods or rough programming methods can not make decisions effectively. Stochastic rough expectation model and stochastic rough chance-constrained model can deal with this kind of uncertain programming.

In order to establish a solution model for stochastic rough programming model, we will give some basic concepts firstly. Definition 1 and 2 come from Liu[3].

Definition 1. Let ξ be a random rough vector on the rough space $(\Lambda, \Delta, A, \pi)$ (here Λ be a nonempty set, A be a σ-algebra of subsets of Λ , Δ be an element in A, and π be a nonnegative, real-valued, additive set function), and f_i : $R^n \rightarrow R$ be continuous functions, j=1, 2, …, n. Then the primitive chance of a random rough event characterized by $f_i(\xi)$, $j = 1, 2, \cdots, n$, is a function from [0, 1] to [0, 1], defined as

$$\mathrm{Ch}\{f_j(\xi) \le 0\}(\alpha) = \sup\{\beta \mid \mathrm{Tr}\{\lambda \in \Lambda \mid \Pr\{f_j(\xi(\lambda)) \le 0\} \ge \beta\} \ge \alpha\} \text{ , } j = 1, 2, \cdots, m \tag{1}$$

where α and β are predetermined confidence levels. Pr(·) is the probability function and Tr(·) is the trust function coming from Liu[3].

In the Definition 1, a concept of chance function Ch(x) is proposed. It can describe the random roughness as uncertain information considered in this paper.

When we deal with the uncertain information, the average value and the deviation degree as two important features are considered usually. Then we will introduce the concepts of expectation and variance, which will be used to describe a random rough variable.

Definition 2. Let ξ be a random rough variable defined on the rough space $(\Lambda, \Delta, A, \pi)$. Then its expectation is defined by

$$E(\xi) = \int_0^{\infty} \mathrm{Tr}\{\lambda \in \Lambda \,|\, \mathrm{E}[\xi(\lambda)] \ge r\}\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Tr}\{\lambda \in \Lambda \,|\, \mathrm{E}[\xi(\lambda)] \le r\}\mathrm{d}r \text{ ,} \tag{2}$$

and its variance is defined by

$$D(\xi) = E[\xi - E(\xi)]^2 \text{ .} \tag{3}$$

In the following, an example of random rough variable and its expectation and variance will be given. An industry will produce a seasonal product. Because the demand amount is seasonal, the profits ξ are random rough variables, the profits follow a normal distribution, but the expectation values are rough variables ρ and the variance is 1. We can present that $\xi \sim N(\rho, 1)$, ρ=([1, 2], [0, 3]).

Because the solution of uncertainty programming is not the same as normal programming, the evolutionary computation should be used to deal with the uncertainty. Genetic algorithm[23] (GA) could search for the optimal solution without regard to the specific inner connections of the problem. Generally, GA has eight basic components: genetic representation, initial population, evaluation function, reproduction selection scheme, genetic operators, generational selection scheme, stopping criteria and GA parameter setting.

4. Stochastic Rough Multi-objective Programming

In industrial production, many decision problems are multi-objective programming with random roughness. For example in the supply-demand problem, the decision maker wants the profit and the using efficiency of using the machinery to maximize. There are two objectives. At the same time, there is a class of uncertain problems with randomness and roughness simultaneously which are still paid less attention so far.

First we give the general form of stochastic rough multi-objective programming model

$$\max\, [f_1(x, \xi), f_2(x, \xi), \cdots, f_p(x, \xi)]$$
$$\text{s.t.} \begin{cases} g_j(x, \xi) \le 0, \;\; j = 1, 2, \cdots, m, \\ x \in \mathrm{M} \subseteq \mathrm{R}^n, \end{cases} \tag{4}$$

where $x=(x_1, x_2, \ldots, x_n)^T$ is the decision vector. $\xi=(\xi_1, \xi_2, \ldots, \xi_n)$ is the given random rough vector on rough space. $f_i(x, \xi)$, $g_j(x, \xi)$ are random rough variable functions, i=1, 2,…, p, j=1, 2,…, m. M is a subspace of R^n.

This model is a multi-objective programming and contains the random information and rough information in decision process. We convert it into a crisp model and apply multi-objective programming method to solve it.

Based on the expected value and chance function of random rough variables, the traditional two stochastic rough multi-objective programming solution models can easily be given as follows[3, 6].

- Expectation model

$$\max\ [E[f_1(x,\xi)],\ E[f_2(x,\xi)], \cdots,\ E[f_p(x,\xi)]]$$
$$\text{s.t.}\begin{cases} E[g_i(x,\xi)] \le 0,\ \ i = 1, 2, \cdots, m, \\ x \in \mathrm{M} \subseteq \mathrm{R}^n. \end{cases} \tag{5}$$

- Chance-constrained model

$$\max\ [\bar{f}_1(x),\ \bar{f}_2(x), \cdots,\ \bar{f}_p(x)]$$
$$\text{s.t.}\begin{cases} \mathrm{Ch}\{f_k(x,\xi) \ge \bar{f}_k(x)\}(\gamma_k) \ge \rho_k,\ k = 1, 2, \cdots, p, \\ \mathrm{Ch}\{g_j(x,\xi) \le 0\}(\eta_j) \ge \alpha_j,\ j = 1, 2, \cdots, m, \\ x \in \mathrm{M} \subseteq \mathrm{R}^n, \end{cases} \tag{6}$$

where $0 \le \rho_k, \gamma_k, \alpha_j, \eta_j \le 1$, $k = 1, 2, \cdots, p$, $j = 1, 2, \cdots, m$ are predetermined confidence levels.

As solution model, expectation model uses the expectation of random rough variable instead of the random rough variable, chance-constrained model uses chance function to describe the degree of objectives and constraints satisfaction, and they can convert the model (4) into a crisp model. We can see that the each solution model considered one feature of stochastic rough multi-objective programming model. In order to make the solution model precise enough, we need to consider the features of the constraints satisfaction degree and objective function together.

When the variation is large, the expectation could not describe the random rough variable effectively. So we could not get the optimal solution of the stochastic rough multi-objective programming by model (5). Compared with model (5), the model (6) can control the quality of the decision beforehand, but it still can't realize the solution if the distribution of $f_i(x, \xi)$ and $g_j(x, \xi)$ are complex or difficult to be described.

4.1. Stochastic rough multi-objective synthesis effect (MOSE) model

Because $f_i(x, \xi)$ contains random rough vector ξ with uncertainty, we can not give the order of different $f_i(x, \xi)$. The expectation and variance are the two main features describing random rough variables. Many literatures describe the uncertainty of $f_i(x, \xi)$ by $E(f_i(x, \xi))$ and $D(f_i(x, \xi))$. $E(f_i(x, \xi))$ is the principal index to describe the size feature of $f_i(x, \xi)$ and $D(f_i(x, \xi))$ is the reliable degree of $E(f_i(x, \xi))$ representing $f_i(x, \xi)$, $i = 1, 2, \cdots, p$.

When we deal with the multi-objective programming model, the usual method is weighted average of the objectives. Following this way and considering the $E(f_i(x, \xi))$ and $D(f_j(x, \xi))$, we give a function $K(x, \xi)$= $[\sum_{j=1}^{p} \omega_i E(f_i)][1 + k\sum_{j=1}^{p} l_i D(f_i)]^{-1}$ to describe ($f_1(x, \xi)$, $f_2(x, \xi)$, …, $f_p(x, \xi)$), $0 \le \omega_i$, $l_i \le 1$ and $\sum_{i=1}^{p} \omega_i = 1$, $\sum_{i=1}^{p} l_i = 1$, ω_i and l_i are the weight of $E(f_i(x, \xi))$ and $D(f_i(x, \xi))$. With different ω_i and l_i, the important degrees of different objective functions are shown in u=$K(x, \xi)$.

Stochastic rough multi-objective programming is an uncertain decision, and the results can't satisfy the related constraints absolutely. Therefore, it is more suitable synthetically considering the constraint satisfaction and the size of objectives. To establish a general solution model under this idea, we can synthesize the objective and constraint satisfaction through some strategies (called the synthesis effect function), then discuss the programming based on the synthesis effect value. The following will give the axiomatic system for multi-attribute synthesis effect functions.

Principle 1. If the satisfaction degree of constraints $g_j(x, \xi)$ is the same, the greater the concentrated quantification value of objectives $K(x, \xi)$ is, the better the effect is.

Principle 2. If $K(x, \xi)$ is the same, the greater the satisfaction degree of constraints $g_j(x, \xi)$ is, the better the effect is.

Principle 3. If the constraints $g_j(x, \xi)$ are absolutely satisfied, the decision only depends on the value of objectives $K(x, \xi)$.

Principle 4. If at least a constraint $g_j(x, \xi)$ is dissatisfied, we can't make a decision.

Let Θ be the range of u, $S(u, v)$= $S(u, v_1, v_2,..., v_m)$ is a map on $\Theta\times[0, 1]^m \to (-\infty, +\infty)$. If the following conditions 1-4 hold we call $S(u, v)$ satisfying synthesis effect function on Θ .

Condition 1. For any given $v_i \in [0, 1]$, $S(u, v)$ is monotone increasing in each u , $i = 1, 2, \cdots, m$.

Condition 2. For any given $u \in \Theta$, $S(u, v)$ is monotone increasing in v_i , $i = 1, 2, \cdots, m$.

Condition 3. $S(u, 1, 1,…, 1)$ is strictly monotone increasing in u.

Condition 4. When $\prod_{j=1}^{m} v_j = 0$, $S(u, v_1, v_2,..., v_m)$=0 for any $u \in \Theta$.

If we regard the synthesis strategy of objective and degree of constraints satisfaction as a function $S(u, v)$= $S(u, v_1, v_2,..., v_m)$, u=$K(x, \xi)$, v_i is the satisfaction degree of the i th constraint with the conversion interval [0, 1], the above four conditions hold.

According to the above principles, we have the following conclusions:

1) For any given $c \in [0, +\infty), k, k_j \in (0, +\infty)$, $S(u, v_1, v_2,..., v_m)$= $(u+c)^k \Pi_{j=1}^{m} v_j^{k_j}$ is a synthesis effect function on $(-\infty, +\infty)$.

2) For any given $k, k_j \in (0, +\infty)$, $S(u, v_1, v_2,..., v_m)$ $= \exp(ku)\Pi_{j=1}^{m} v_j^{k_j}$ is a synthesis effect function on $(-\infty, +\infty)$.

3) For any given $a, b, a_j \in (0, \infty)$, $S(u, v_1, v_2,..., v_m)$ $= au + b\sum_{j=1}^{m} a_j v_j$ is a synthesis effect function on $(-\infty, +\infty)$.

If we consider that $\beta_j(x) = \mathrm{Ch}\{g_j(x,\xi)) \le 0\}(\eta_j)$, $j = 1, 2, \cdots, m$, $u = K(x, \xi)$ and $v = (\beta_1(x), \beta_2(x), \cdots,$ $\beta_m(x))$. $S(u, v) = S(K(x, \xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$ is the objective function. Then model (4) can be converted into the following model (7):

$$\begin{cases}\max S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x)),\\ \text{s. t. } x \in \mathrm{M} \subseteq \mathrm{R}.\end{cases} \tag{7}$$

Model (7) is called MOSE model of model (4).

4.2. Features of MOSE Model

MOSE model is an effective model to deal with stochastic rough multi-objective problem. In this part some important features of MOSE model will be introduced.

Theorem 1. *If model (4) is a crisp programming,* $(\inf K(x,\xi), \sup K(x,\xi)) \subset \Theta$, $S(u, v_1, v_2\ \ , \cdots, v_m)$ *is a synthesis effect function on* Θ. *When the number of elements in* $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$ *is more than 1, and* $K(x, \xi)$ *is not constant function on* $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$, *model (4) and model (7) have the same optimal solution.*

Proof. As the constraint of a crisp programming only has two states: satisfied or dissatisfied, $\beta_i(x)$ only has two values: 0 or 1. That is,

$\beta_j(x) = \mathrm{Ch}\{g_j(x,\xi) \le 0\}(1) = 1$ for $g_j(x,\xi) \le 0$,

$\beta_j(x) = \mathrm{Ch}\{g_j(x,\xi) \le 0\}(0) = 0$ for $g_j(x,\xi) > 0$.

1) If $x^*$is the optimal solution of model (4), $\beta_j(x)$= Ch{$g_j(x^*, \xi)$≤0}(1)=1, j=1, 2, …, m. In the following, we will prove S($K(x, \xi)$, $\beta_1(x)$, $\beta_2(x)$, …, $\beta_m(x)$) ≤ S($K(x^*, \xi)$, $\beta_1(x^*)$, $\beta_2(x^*)$, …, $\beta_m(x^*)$) through two cases.

$$S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$$
$$= S(K(x,\xi), 1, 1, \cdots, 1) \le S(K(x^*,\xi), 1, 1, \cdots, 1)$$
$$= S(K(x^*,\xi)\ \beta_1(x^*), \beta_2(x^*), \cdots, \beta_m(x^*)).$$

For any $x \in \bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$, by $K(x, \xi) \le K(x^*, \xi)$, $\beta_j(x)$=Ch{$g_j(x, \xi)$≤0}(1)=1, $D(K(x, \xi))$=0, and the monotone increasing of $K(x, \xi)$ and $S(u, 1, 1, …, 1)$, we have

$$S(K(x,\xi)\ \ \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$$
$$= S(K(x,\xi), 1, 1, \cdots, 1) \le S(K(x^*,\xi), 1, 1, \cdots, 1)$$
$$= S(K(x^*,\xi), \beta_1(x^*), \beta_2(x^*), \cdots, \beta_m(x^*)).$$

For any $x \in \mathrm{M} - \bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$, at least there is a $j \in \{1, 2, \cdots, m\}$ such that $\beta_j(x)$=Ch{$g_j(x, \xi)$≤0}(0)=0, so $\prod_{j=1}^{m} \beta_j(x) = 0$. Combining with the properties of $S(u, v_1, v_2, \cdots, v_m)$, we have

$$S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$$
$$= S(K(x,\xi), 0, 0, \cdots, 0) \le S(K(x^*,\xi), 1, 1, \cdots, 1)$$
$$= S(K(x^*,\xi), \beta_1(x^*), \beta_2(x^*), \cdots, \beta_m(x^*)).$$

2) If $x^*$is the optimal solution of model (7), then $S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x)) \le S(K(x^*,\xi), \beta_1(x^*),$ $\beta_2(x^*), \cdots, \beta_m(x^*))$ for any $x \in \mathrm{M}$.

Since $S(u, 1, 1, …, 1)$ is strictly monotone increasing, we only prove that $x^* \in \bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$, then we can obtain that $x^*$is the optimal solution of model (4). Actually, if $x^* \notin \bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$, then there must exist $j \in \{1, 2, \cdots, m\}$ such that $\beta_j(x^*) = \mathrm{Ch}(g_j(x^*,\xi) \le 0)(0) = 0$. Then we have that $S(K(x^*,\xi), \beta_1(x^*), \beta_2(x^*),$ $\cdots, \beta_m(x^*)) \le S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$ for any $x \in M$, that is $S(K(x^*,\xi), \beta_1(x^*), \beta_2(x^*), \cdots, \beta_m(x^*))$ $= S(K(x,\xi), \beta_1(x), \beta_2(x),\ \cdots, \beta_m(x))$, for any $x \in M$. This implies that $S(K(x,\xi), \beta_1(x), \beta_2(x), \cdots, \beta_m(x))$ is a constant function or $-\infty$ on $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$. By this and the number of elements in $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$ is more than 1 and $S(u, 1, 1, …, 1)$ is strictly monotone increasing, we know that $S(K(x, \xi), 1, 1, …, 1)$ is a constant function on $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$. This contradicts to the conditions. □

In fact, most crisp programming problems satisfy that the number of elements in $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$ is more than 1 and $f_i(x, \xi)$ is not constant function on $\bigcap_{j=1}^{m}\{w \mid g_j(w,\xi) \le 0\}$. So model (7) can be considered as an extension of ordinary programming problem. But if $f_i(x, \xi)$ or $g_j(x, \xi)$ contains random roughness, the optimal solution determined by different synthesis effect functions is generally not the same. Sometimes these differences are big, it can be popularly interpreted that these differences are caused by different decisions.

Besides model (7) is the extension of expectation model and chance-constrained model. If we select different synthesis effect function, model (7) can also be converted into the above two models. For example, using $E(f_i(x, \xi))$ to concentralizedly describe the size of $f_i(x, \xi)$, i=1, 2, …, p, and $\beta_j(x)=\delta(E(-g_j(x, \xi)))$ is the satisfaction degree of $g_j(x, \xi)\leq 0$, model (7) is the expectation model (5) for $S(u, v_1, v_2,..., v_m)= \exp(u)\cdot\theta(\prod_{j=1}^{m}\delta(v_j-\alpha_j))$. Here, $\delta(t)$=0 for t<0, and $\delta(t)$=1 for $t\geq 0$; and $\theta(0)=-\infty$, $\theta(1)$=1. Using $\overline{f}_i(x)$ to concentralizedly describe the size of $f_i(x, \xi)$, Ch$\{g_j(x, K(x, \xi))\leq 0\}(\eta)$ and the satisfaction degree of $g_j(x, \xi)\leq 0$, model (7) is the chance-constrained model (6) for $S(u, v_1, v_2,..., v_m)= \exp(u)\cdot\theta(\prod_{j=1}^{m}\delta(v_j-\alpha_j))$. Here, $\mathrm{Ch}\{f(x,\xi)\geq\overline{f}(x)\}(\gamma)\geq\rho$, $0\leq\gamma, \rho\leq 1$, and $\delta(t)$=0 for t<0, and $\delta(t)$=1 for $t\geq 0$; and $\theta(0)=-\infty$, $\theta(1)$=1.

From above we can see that the MOSE model contains expectation model and chance-constrained model. So we can transform the model (4) to MOSE model to solve it. The solution of MOSE model contains the solution of expectation model and chance-constrained model.

In industrial production and logistic distribution there are many stochastic rough multi-objective programming models can be converted into crisp convex programming models with proper synthesizing effect functions. For example in a logistic distribution problem, we consider the distribution cost and distribution service level as two objectives, the distribution road and transportation capacity as constraints. Because demand amount is seasonal which contains roughness and the distribution cost contains randomness, the profits are random rough variables. This is a problem which contains both roughness and randomness. If the two considered objective functions are linear functions and constraints are linear too and the selected synthesizing effect function is a linear function the MOSE model is a convex programming.

If the MOSE model is a convex programming, we can prove that the local optimized solution is the global optimized solution. There are many methods used in convex programming including Fibonacci method, 0.618 method, the fastest descent method, Newton method, restricted function method etc. But when we deal with some complex problems, the MOSE model is not a convex programming, and the above methods can not be used. Genetic algorithm is an effective method to solve non-convex programming problems. In part 6 we will solve the MOSE model by genetic algorithm and an example of the industrial production problem will be given.

5. Solution Procedure

In what follows, we will give the solution procedure to solve stochastic rough multi-objective problem. Here we will establish MOSE model by synthesis strategy of objective and satisfaction degree of constraints. In some cases, it is very hard to convert the random rough constrains into their deterministic equivalents for given confidence levels. Liu[3] has discussed how to make use of random rough simulation to compute chance functions. We use the technique of random rough simulation to handle random rough objective functions and check the random rough constraints.

Step 1 Input the initial programming model. Based on the practical problems, a multi-objective programming model with stochastic rough coefficients is established. So the stochastic rough multi-objective programming model can be given.

Step 2 Concentrated quantification value $u=K(x, \xi) =(\sum_{j=1}^{p}\omega_j s_j)[1+k(\sum_{j=1}^{p}l_j t_j)^a]^{-1}$. Select a class of synthesizing effect functions such as $S(u, v)=uv^b$. With different coefficients k, a, b we can get different concentrated quantification operators and synthesizing effect functions.

Step 3 Convert the stochastic rough multi-objective programming model to MOSE model. Through proper $K(x, \xi)$ and synthesizing effect functions we can convert the stochastic rough multi-objective programming model to MOSE model. MOSE model is a multi-objective programming model without uncertainty.

Step 4 Solve MOSE model by genetic algorithm:

1) Input the parameters population size $N_{\text{pop-size}}$, crossover probability p_c, mutation probability p_m.

2) Initialize $N_{\text{pop-size}}$ chromosomes and check feasibility of chromosomes by random rough simulations technology[4].

3) Update the chromosomes by crossover and mutation operations and random rough simulation is used to check the feasibility of offspring.

4) Select the chromosomes by spinning the roulette wheel.

5) Make the crossover operation and the mutation operation.

6) Repeat the 2) to the 5), until completes the cycle-index.

7) Get the best chromosome as the optimal solution, and obtain the optimum value.

Step 5 Output the solution.

6. Case-based Example

Next an example will be given about industry produce process. An industry will produce three kinds of seasonal products. Because the demand amount is seasonal, the profits are random rough variables. The profits follow a normal distribution, but the expected values are rough variables. When producing each product, the using efficiency of the machinery is a random rough variable, but the coefficients are different. The products are no less than 20, and the gross amount is no less than 100. The other coefficients can be seen in *Table 1*. Here (c_1, c_2, c_3)=(1, 2, 4), (c_4, c_5, c_6)=(1.2, 0.8, 1.5), ρ_i are rough variables, $i = 1, 2, \cdots, 9$, and $\xi_1 \sim N(\rho_1, 1)$, ρ_1=([1, 2], [0, 3]); $\xi_2 \sim N(\rho_2, 4)$, ρ_2=([2, 3], [1, 4]); $\xi_3 \sim N(\rho_3, 1)$, ρ_3=([3, 4], [2, 5]); $\xi_4 \sim N(\rho_4, 4)$, ρ_4=([0, 1], [0, 3]); $\xi_5 \sim N(\rho_5, 1)$, ρ_5=([1, 2], [0, 3]); $\xi_6 \sim N(\rho_6, 1)$, ρ_6=([2, 3], [0, 3]); $\xi_7 \sim N(\rho_7, 4)$, ρ_7=([0, 1], [0, 3]); $\xi_8 \sim N(\rho_8, 1)$, ρ_8=([1, 2], [0, 3]); $\xi_9 \sim N(\rho_9, 1)$, ρ_9=([2, 3], [0, 3]).

Table 1. The resource demand in a production process

Product	1	2	3	Possible use amount
Worker amount	1	3	2	500
Storage capacity Profit	$c_1\xi_4$	$c_2\xi_5$	$c_3\xi_6$	800
Use efficiency	ξ_1	ξ_2	ξ_3	
	$c_4\xi_7$	$c_5\xi_8$	$c_6\xi_9$	

The problem is how many products to produce to make the total of storage capacity profit and use efficiency maximum.

Model of this problem is

$$\begin{aligned}&\max\ f_1 = \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3\\&\max\ f_2 = c_4\xi_7 x_1 + c_5\xi_8 x_2 + c_6\xi_9 x_3\\&\text{s.t.}\begin{cases}x_1 + x_2 + x_3 \ge 100,\\ x_1 + 3x_2 + 2x_3 \le 500,\\ c_1\xi_4 x_1 + c_2\xi_5 x_2 + c_3\xi_6 x_3 \le 800,\\ x_1 \ge 20,\ x_2 \ge 20,\ x_3 \ge 20.\end{cases}\end{aligned}\tag{8}$$

A. Expectation value model

$$\begin{aligned}&\max\ (1.5\omega_1 + \omega_2)x_1 + (2.5\omega_1 + 1.5\omega_2)x_2 + (3.5\omega_1 + 2\omega_2)x_3\\&\text{s.t.}\begin{cases}x_1 + x_2 + x_3 \ge 100,\\ x_1 + 3x_2 + 2x_3 \le 500,\\ x_1 + 6x_2 + 4x_3 \le 800,\\ x_1 \ge 20,\ x_2 \ge 20,\ x_3 \ge 20,\end{cases}\end{aligned}\tag{9}$$

where the weight of first objective is ω_1 and the weight of second objective is ω_2.

B. Chance-constrained model

$$\begin{aligned}&\max\ \overline{f}_1, \overline{f}_2\\&\text{s.t.}\begin{cases}x_1 + x_2 + x_3 \ge 100,\\ x_1 + 3x_2 + 2x_3 \le 500,\\ \mathrm{Ch}\{\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 \ge \overline{f}_1(x)\}(0.9) \ge 0.9,\\ \mathrm{Ch}\{c_4\xi_7 x_1 + c_5\xi_8 x_2 + c_6\xi_9 x \ge \overline{f}_2(x)\}(0.9) \ge 0.9,\\ \mathrm{Ch}\{\xi_4 x_1 + 4\xi_5 x_2 + 2\xi_6 x_3 \le 800\}(0.9) \ge 0.9,\\ x_1 \ge 20,\ x_2 \ge 20,\ x_3 \ge 20,\end{cases}\end{aligned}\tag{10}$$

where the confidence levels are all 0.9.

The expectation model is very easy to be understood, and the solution procedure is very easy too. But the chance-constrained model is not easy to calculate. We must solve it by genetic algorithm. The random rough simulation is used to check the feasibility of offspring. The calculation complexity is higher than expectation model.

Using the method of Xu[4], we can convert the chance-constrained model into the following model (11):

$$\begin{aligned}&\min\ \lambda\\&\text{s.t.}\begin{cases}\text{-}(2.4x_1 + 1.4x_2 + 2.2x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2})\\ \ge -850.83 + 250.92(\overline{\rho}_1 - \lambda),\\ \text{-}(2.88x_1 + 1.92x_2 + 3.6x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2})\\ \ge -1126.83 + 375.63(\overline{\rho}_2 - \lambda),\\ x_1 + x_2 + x_3 \ge 100,\\ x_1 + 3x_2 + 2x_3 \le 500,\\ 2.88x_1 + 1.92x_2 + 3.6x_3 + 1.28\sqrt{x_1^2 + 4x_2^2 + x_3^2} \le 800,\\ x_1 \ge 20,\ x_2 \ge 20,\ x_3 \ge 20,\\ 0 \le \lambda \le 1,\end{cases}\end{aligned}\tag{11}$$

where $\overline{\rho}_1 = \overline{\rho}_2 = 1$. The weight of first objective is ω_1. The weight of second objective is ω_2.

C. MOSE Model

Let $S(u, v)=uv^b$, $u=K(x, \xi) = (\sum_{j=1}^{p} \omega_j s_j)[1 + k(\sum_{j=1}^{p} l_j t_j)^a]^{-1}$, a, b, $k \ge 0$, and the confidence levels is 0.9, then we can convert model (8) into following crisp programming model (12).

$$\max[\omega_1 E(f_1)+\omega_2 E(f_2)][1+k[l_1 D(f_1)+l_2 D(f_2)]^a]^{-1}$$
$$[\mathrm{Ch}\{\xi_4 x_1+4\xi_5 x_2+2\xi_6 x_3 \le 800\}(0.9)]^b \qquad (12)$$
$$\text{s.t.}\begin{cases} x_1+x_2+x_3 \ge 100, \\ x_1+3x_2+2x_3 \le 500, \\ x_1 \ge 20,\ x_2 \ge 20,\ x_3 \ge 20. \end{cases}$$

Using genetic algorithm (its parameters setting are: binary code mode; mutation probability is 0.001; crossover probability is 1; population size is 80; evolutionary generations is 1000; random rough simulation is used to check the feasibility of offspring), we can get Table 2 and Table 3.

Table 2. The results for every method (ω_1=0.3, ω_2=0.7, l_1=l_2=1)

Solving model		Optimal solution	Expectation of objectives	Variation of objectives	Constraint satisfaction
Expectation value model		(200, 20, 120)	$E(f_1)=770$, $E(f_2)=624$	$D(f_1)=55200$, $D(f_2)=1184656$	0.55
Chance-constrained model		(75.6, 21.2, 40.2)	$E(f_1)=307.1$, $E(f_2)=236.8$	$D(f_1)=8230$, $D(f_2)=168526$	0.92
Chance-constrained equivalent model		(20, 60, 20)	$E(f_1)=250$, $E(f_2)=156$	$D(f_1)=8000$, $D(f_2)=14724$	0.90
MOSE model	*k*=1, *a*=1, *b*=1	(78.3, 66, 20)	$E(f_1)=352.3$, $E(f_2)=233.2$	$D(f_1)=15243$, $D(f_2)=180258$	0.72
	k=1, *a*=1, *b*=10	(88.7, 22.4, 26.3)	$E(f_1)=281.1$, $E(f_2)=212.2$	$D(f_1)=9563$, $D(f_2)=228467$	0.81
	k=1, *a*=5, *b*=1	(60.4, 57.3, 20)	$E(f_1)=303.9$, $E(f_2)=201.2$	$D(f_1)=10615$, $D(f_2)=108068$	0.89
	k=1, *a*=5, *b*=10	(27.6, 20, 64.7)	$E(f_1)=317.9$, $E(f_2)=251.2$	$D(f_1)=5748$, $D(f_2)=31613$	0.96
	k=10, *a*=1, *b*=1	(90.2, 28.4, 20)	$E(f_1)=276.3$, $E(f_2)=202.3$	$D(f_1)=10149$, $D(f_2)=235734$	0.89
	k=10, *a*=1, *b*=10	(96.6, 20, 20)	$E(f_1)=264.9$, $E(f_2)=199.9$	$D(f_1)=10532$, $D(f_2)=269905$	0.94
	k=10, *a*=5, *b*=1	(77.2, 24.9, 30.6)	$E(f_1)=285.2$, $E(f_2)=214.3$	$D(f_1)=8136$, $D(f_2)=174147$	0.92
	k=10, *a*=5, *b*=10	(55.2, 20.2, 20)	$E(f_1)=203.3$, $E(f_2)=150.5$	$D(f_1)=4263$, $D(f_2)=88916$	0.99

Table 3. The results for every method (ω_1=ω_2=1, l_1=0.3, l_2=0.7)

Solving model		Optimal solution	Expectation of objectives	Variation of objectives	Constraint satisfaction
Expectation value model		(200, 20, 120)	$E(f_1)=770$, $E(f_2)=624$	$D(f_1)=55200$, $D(f_2)=1184656$	0.55
Chance-constrained model		(75.6, 21.2, 40.2)	$E(f_1)=307.1$, $E(f_2)=236.8$	$D(f_1)=8230$, $D(f_2)=168526$	0.92
Chance-constrained equivalent model		(20, 60, 20)	$E(f_1)=250$, $E(f_2)=156$	$D(f_1)=8000$, $D(f_2)=14724$	0.90
MOSE model	*k*=1, *a*=1, *b*=1	(37.7, 98.2, 20)	$E(f_1)=372.1$, $E(f_2)=223.1$	$D(f_1)=21108$, $D(f_2)=48005$	0.89
	k=1, *a*=1, *b*=10	(65.1, 43.5, 38.7)	$E(f_1)=341.9$, $E(f_2)=246.4$	$D(f_1)=9520$, $D(f_2)=126636$	0.95
	k=1, *a*=5, *b*=1	(50.2, 46.1, 29)	$E(f_1)=292.1$, $E(f_2)=202.6$	$D(f_1)=7611$, $D(f_2)=75830$	0.91
	k=1, *a*=5, *b*=10	(44.6, 32, 55.4)	$E(f_1)=304.8$, $E(f_2)=258.1$	$D(f_1)=7106$, $D(f_2)=64849$	0.94
	k=10, *a*=1, *b*=1	(50.3, 20.4, 28.9)	$E(f_1)=227.6$, $E(f_2)=171.5$	$D(f_1)=4198$, $D(f_2)=75012$	0.88
	k=10, *a*=1, *b*=10	(86.5, 20, 30)	$E(f_1)=284.8$, $E(f_2)=217.8$	$D(f_1)=9182$, $D(f_2)=217770$	0.90
	k=10, *a*=5, *b*=1	(56.2, 30.9, 22.6)	$E(f_1)=240.7$, $E(f_2)=172.3$	$D(f_1)=5579$, $D(f_2)=92723$	0.92
	k=10, *a*=5, *b*=10	(44.2, 20.9, 55)	$E(f_1)=311.1$, $E(f_2)=243.1$	$D(f_1)=5852$, $D(f_2)=63351$	0.98

Obviously, a, b, k, ω_1, ω_2, l_1, l_2 are the parameters describing the uncertainty consciousness. We use the synthesis effect function $S(K(x, \xi), \beta)=[\omega_1 E(f_1)+\omega_2 E(f_2)][1+k[l_1 D(f_1)+l_2 D(f_2)]^a]^{-1}\beta^b$, a, b, $k\ge 0$, where $\omega_1+\omega_2=1$, $l_1+l_2=1$. From the two tables we can see that if $b>a$, $b>k$, the decision making consciousness tends to constraint satisfaction as big as possible. If $b<a$, $b<k$, the decision making consciousness tends to variation of objective as small as possible. When $\omega_1>\omega_2$ and $l_1=l_2$, the decision making consciousness tends to the first objective, or it tends to the second objective. When $l_1>l_2$ and $\omega_1=\omega_2$, the decision making consciousness tends to the second objective, or it tends to the first objective. Therefore the MOSE model is closer to stochastic rough programming model. The above analysis and computation results indicate: 1) For the same stochastic rough programming problem, the variations of the decision results are smaller than that of expectation model and the satisfaction of constraints by using MOSE model are bigger than that of expectation model, which shows that the decision reliability of MOSE model is greater than the expectation model. 2) Using MOSE model when k is small, the objective function expectations are generally better than chance-constrained model. It indicates that using some MOSE models decision results are better than the chance-constrained model. 3) With different MOSE models, the decision results are different, and even the different is great. Therefore, MOSE model can effectively integrate uncertainty process consciousness into decision process. For example if $k=a$, the bigger b is, the higher the chance constrained satisfaction of the solution is. 4) The solutions of MOSE model in some decision making consciousnesses are superior to the chance constrained equivalent model proposed in Xu[4]. For example when $\omega_1=0.3$, $\omega_2=0.7$, $l_1=l_2=1$, $k=1$, $a=5$, $b=10$, the solution of the MOSE model is (27.6, 20, 64.7). Expectation of objectives 317.9, 251.2 are bigger than expectation of objective of chance-constrained equivalent model which is 250, 156. Variation of first objective 5748 is smaller than variation of objective of chance-constrained equivalent model which is 8000. Constraint satisfaction 0.96 is bigger than constraint satisfaction of chance-constrained equivalent model which is 0.9. Only variation of second objective 31613 is not smaller than variation of objective of chance-constrained equivalent model which is 14724. So when $\omega_1=\omega_2=1$, $l_1=0.3$, $l_2=0.7$, $k=1$, $a=5$, $b=10$, the solution of MOSE model is better than the solution of chance-constrained equivalent model, except of variation of second objective. We can also see that when $\omega_1=\omega_2=1$, $l_1=0.3$, $l_2=0.7$, $k=10$, $a=5$, $b=10$, the solution of MOSE model is better than the solution of chance-constrained equivalent model, except of variation of second objective.

Table 2 and *Table 3* also show when we choose other proper coefficients we can also get the good solution as above not only $k=1$, $a=5$, $b=10$ in *Table 2* and $k=10$, $a=5$, $b=10$ in *Table 3*.

As the above example when $(c_4, c_5, c_6)=(1.2, 10, 1.5)$ we recalculate the solution in different solution models.

1) Using chance-constrained equivalent mode the solution is (60, 20, 20), $E(f_1)=210$, $E(f_2)=156$, $D(f_1)=4800$, $D(f_2)=144580$, and the degree of constraint satisfaction is 0.55.

2) Using MOSE model with $S(K(x, \xi), \beta)=[\omega_1 E(f_1)+\omega_2 E(f_2)][1+k[l_1 D(f_1)+ l_2 D(f_2)]^a]^{-1}\beta^b$, when $\omega_1=0.3$, $\omega_2=0.7$, $l_1=l_2=1$, $k=1$, $a=5$, $b=10$, the solution is (33.5, 20.4, 44.7), $E(f_1)=268.2$, $E(f_2)=207.8$, $D(f_1)=4430$, $D(f_2)=79056$, and the degree of constraint satisfaction is 0.96.

Obviously the expectations of objectives in MOSE model are bigger than expectations of objectives in chance-constrained equivalent model. The variances of objectives in MOSE model are smaller than variances of objectives in chance-constrained equivalent model. The degree of constraint satisfaction in MOSE is bigger than the degree of constraint satisfaction in chance-constrained equivalent model. So the solution in MOSE model is better than the solution in chance-constrained equivalent model.

In this example, only one kind of synthesis effect functions $S(u, v)$ is used. If we use other kinds of synthesis effect functions $S(u, v)$ to construct MOSE model, we can get the more results. Different approximately satisfactory solutions can be obtained by changing the coefficients a, b, k, ω_1, ω_2, l_1, l_2.

7. Conclusion

In this paper we present a formulation of multi-objective programming with random rough coefficients. MOSE model is developed as a crisp equivalent solution model. MOSE model solves the stochastic rough programming problem and develops the traditional two stochastic rough multi-objective programming solution models. Using the advantage of solving non-convex programming of GA, the random rough simulation based on GA is applied to get the optimal solution to some common problems. We analyze the advantages of different solutions in different decisions. The MOSE model is both more feasible and efficient than traditional solution methods for handling some complex problems.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61074088, 71071049).

References

1. X. Z. Jin, X. Lei and J. Du, Evolutionary Game Theory in Multi-objective Optimization Problem, *International Journal of Computational Intelligence Systems*. 3 (2010) 74-87.
2. X. W. Zhang and H. Liu, A scalable coevolutionary multi-objective particle swarm optimizer, *International Journal of Computational Intelligence Systems*. 3(5) (2010) 590-600.
3. B. D. Liu, *Theory and Practice of Uncertain Programming,* 2nd edn. (Physica Verlag, Berlin, 2009).
4. J. P. Xu and L. M. Yao, A class of multiobjective linear programming models with random rough coefficients, *Mathematical and Computer Modelling*. 49(1-2) (2009) 189-206.
5. J. P. Xu and L. M. Yao, A class of expected value multi-objective programming problems with random rough coefficients, *Mathematical and Computer Modelling*. 50(1-2) (2009) 141-158.
6. B. D. Liu, *Uncertain Theory: An Introduction to its Axiomatic Foundations,* 2nd edn. (Springer-Verlag, Berlin, 2007).
7. A. Azaron, K. N. Brown, S. A. Tarim and M. Modarres, A multi-objective stochastic programming approach for supply chain design considering risk, *Int. J. Production Economics*. 116(1) (2008) 129-138.
8. F. B. Abdelaziz, B. Aouni and R. E. Fayedh, Multi-objective stochastic programming for portfolio selection, *European Journal of Operational Research*. 177(3) (2007) 1811-1823.
9. C. W. Cao, X. S. Gu and Z. Xin, Chance constrained programming models for refinery short-term crude oil scheduling problem, *Applied Mathematical Modelling*. 33(3) (2009) 1696-1707.
10. M. Ramezani, M. Bashiri and R. Tavakkoli-Moghaddam, A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level, *Applied Mathematical Modelling*. 37(1-2) (2013) 328-344.
11. O. Turgut and A. E. Murat, Generating pareto surface for multi objective integer programming problems with stochastic objective coefficients, *Procedia Computer Science*. 6 (2011) 46-51.
12. V. Charles, A. Udhayakumar and V. R. Uthariaraj, An approach to find redundant objective function(s) and redundant constraint(s) in multi-objective nonlinear stochastic fractional programming problems, *European Journal of Operational Research*. 201(2) (2010) 390-398.
13. F. B. Abdelaziz, Solution approaches for the multiobjective stochastic programming, *European Journal of Operational Research*. 216(1) (2012) 1-16.
14. X. D. Zhang, G. H. Huang, C.W. Chen, Z. F. Liu and Q. G. Lin, A fuzzy-robust stochastic multiobjective programming approach for petroleum waste management planning, *Applied Mathematical Modelling*. 34(10) (2010) 2778-2788.
15. Y. T. Zhu and K. A. Ariyawansa, A preliminary set of applications leading to stochastic semidefinite programs and chance-constrained semidefinite programs, *Applied Mathematical Modelling*. 35(5) (2011) 2425-2442.
16. B. M. Alzalg, Stochastic second-order cone programming: Applications models, *Applied Mathematical Modelling*. 36(10) (2012) 5122-5134.
17. Y. Y. Yao, Probabilistic rough set approximations, *International Journal of Approximate Reasoning*. 49(2) (2008) 255-271.
18. W. Ziarko, Probabilistic approach to rough sets, *International Journal of Approximate Reasoning*. 49(2) (2008) 272-284.
19. Y. Y. Yao, Three-way decisions with probabilistic rough sets, *Information Sciences*. 180(3) (2010) 341-353.
20. W. M. Ma, Probabilistic rough set over two universes and rough entropy, *International Journal of Approximate Reasoning*. 53(4) (2012) 608-619.
21. E. A. Youness, Characterizing solutions of rough programming problems, *European Journal of Operational Research*. 168(3) (2006) 1019-1029.
22. 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*. 35(9) (2011) 4330-4340.
23. H. Holland, *Adaption in Natural and Artifical Systems*, (University of Michigan, Ann Arbor, 1975).