

## Research Article

# A Class of Expected Value Bilevel Programming Problems with Random Coefficients Based on Rough Approximation and Its Application to a Production-Inventory System

**Liming Yao and Jiuping Xu**

*Uncertainty Decision-Making Laboratory, Sichuan University, Chengdu 610064, China*

Correspondence should be addressed to Jiuping Xu; [xujiuping@scu.edu.cn](mailto:xujiuping@scu.edu.cn)

Received 1 February 2013; Revised 16 April 2013; Accepted 19 April 2013

Academic Editor: Ryan Loxton

Copyright © 2013 L. Yao and J. Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on the development of a bilevel optimization model with random coefficients for a production-inventory system. The expected value operator technique is used to deal with the objective function, and rough approximation is applied to convert the stochastic constraint into a crisp constraint. Then an interactive programming method and genetic algorithm are utilized to solve the crisp model. Finally, an application is given to show the efficiency of the proposed model and approaches in solving the problem.

## 1. Introduction

Production and inventory controls are a complex problem, as these two systems are affected by the number of producers and retailers, as well as the level of customer's demand. In a production-inventory system, the main problem is to determine the correct orders to meet the production quantity under some capacity constraints. Some scholars have applied bilevel programming models to formulate the decision making process of both producers and retailers. For example, in a decentralized firm, top management and the executive board, or head-quarters usually make decisions on the budget, and then each division determines a production plan in the full knowledge of the budget [1]. Many scholars [2–5] have also applied two-stage/multistage programming to the production inventory system. In fact, for a production-inventory system, bilevel programming is different from two-stage programming in the production-inventory system. Two-stage programming/multistage programming is usually used in an inventory system with two or multiple periods. However, in this paper we consider a dynamic decision making process between the producers and the retailers with only a single period in the inventory system. Therefore, a bilevel programming approach is more suitable. In a previous work, all parameters in the system were assumed to be deterministic, or in part were assumed to be stochastic or

fuzzy. For example, Sun et al. [6] presented a bilevel programming model with deterministic cost and demand to seek an optimal location for logistics distribution centers. Roghanian et al. [7] proposed a probabilistic bilevel linear multiobjective model for supply chain planning in which the market demand was assumed to be a random variable. Ji and Shao [8] developed a bilevel programming model with fuzzy demand and discounts for a newsboy problem.

Generally, there are many techniques to deal with stochastic programming problems, such as the E model (expectation optimization model), the V model (variance minimization model) and the P model (probability maximization model), proposed by Charnes and Cooper [9, 10]. If we deal with stochastic bilevel programming using the expected value operator or chance-constrained operator, this may result in a loss of information in a realistic production-inventory system. For example, if we use the expected value operator to deal with the upper limit on the number of orders, the upper limit-exceeding condition is strictly forbidden. In fact, this system is usually unworkable when faced with dramatic changes in demand, such as in holiday times. Therefore, a flexible constraint is more suited to handling realistic situations. The rough set, for example, is an efficient tool when dealing with indistinct information. Since Pawlak [11] proposed the concept of the rough set, it has rapidly developed and been applied in many fields. Pawlak and

Sowinski [12] applied the rough set approach to multiattribute decision problems. Xu and Yao [13] discussed a class of linear multiobjective programming problems with random rough coefficients and gave a crisp equivalent model. Youness [14] applied the rough set to the classification of the feasible area in mathematical programming and called it rough programming. Shi et al. [15] proposed the rough approximation based on the probability to deal with the feasible region in order to determine a flexible constraint. In this paper, we deal with a feasible set with random coefficients using probabilistic rough sets in order to determine the flexible constraint.

Bilevel programming is an NP hard problem and it is especially difficult to find numerical solutions using nonlinear bilevel programming. The existence and uniqueness of the bilevel programming solution have been discussed by many scholars [16]. They have also discussed the impact of the lower level optimal solutions on the upper level decision variable. Existing methods for solving bilevel programming problems can be divided into the following categories.

- (i) Vertex enumeration methods: its meaning that the optimal solution is one of these vertices, which are feasible space points in the multilevel problem. Interested readers can refer to these works [17–19].
- (ii) Methods based on Kuhn-Tucker conditions: these methods apply Kuhn-Tucker to deal with the following levels and convert the multilevel model into the single level. This is referred to in [20, 21].
- (iii) Fuzzy approach: the decision maker can define the membership function for the objective functions on all levels and obtain a satisfactory solution. These references [22–24] give a detailed description of this approach.
- (iv) Meta heuristics methods. These methods include genetic algorithms [25], simulated annealing [26], and hybrid tabu-ascent algorithm [27].

However, these methods must be designed according to the specific problem, and, therefore, it is often difficult to determine a usual or normal pattern. Besides, Woon et al. [28, 29] also provided the global optimization methods to solve the dynamic bilevel programming problems. This paper applies an interactive programming technique to convert the bilevel model into a single-level and uses genetic algorithm to solve the complicated nonlinear programming problem. The rest of this paper is organized as follows. In Section 2, a bilevel programming model for a production-inventory system with random demand is developed. An expected value bilevel programming model and its crisp equivalent using rough approximation are proposed, and some solution approaches are suggested in Section 3. In Section 4, an application is given to show the significance of the proposed models and algorithms. Finally, some conclusions are made in Section 5.

## 2. Mathematical Modelling

Some scholars [6, 30] have applied the bilevel programming model to research production-inventory systems, but have usually considered it in only a crisp environment. However,

realistically, because there are some uncertain factors that impact the system, this consideration is flawed. Thus, an improved bilevel programming model for realistic situations is considered in the following section.

*2.1. Key Problem Statement.* The production-inventory system at Auchan's second branch store in Chengdu is considered in this section. Since it is a new store, the production aim is to expand the business to grab market share, so cost control rather than the pursuit of profit is a key factor. However, the primary objective of the retailer is to achieve a higher gross profit and to pursue maximum net profit. In this system, on one hand, Auchan wants to pursue profit, and, on the other hand, it has to ensure the continuing support of producers. Thus, determining a reasonable inventory plan for retailers and a reasonable production plan for suppliers is important. From the retailer's view, this system is a bilevel programming problem, where the upper level has the objective of achieving maximum profit, while the lower level has the objective of controlling supplier's production costs.

A few commodities are selected to determine the plan. Generally, order costs, available storage space, and available storage costs limit inventory level; thus, customer's demand determines the order quantity. From historical data, customer's demand is often a normally distributed random variable over a week (from Wednesday to Wednesday) and peak sales are seen on the weekend. For instance, Wuhu Soya Bean Oil sells about 1050 bottles with around 600 bottles sold on the weekend, and 400 bottles sold over the 5 other weekdays. Therefore, these sales can be seen as a normally distributed random variable. Further, the order costs, holding costs and shortage costs, may be vague values which change according to the international price index. In this situation, the bilevel programming model becomes a stochastic programming problem. In this paper, we only consider a production-inventory problem with random demand; that is, only customer's demand is considered a random variable.

Assume that there are  $m$  plants which manufacture  $n$  items. The decision maker orders items from  $m$  plants. The enterprise aims to maximize overall profit and minimize overall costs including order costs and inventory costs. The notations are as follows:

- $n$ : number of items
- $m$ : number of plants
- $x_i$ : inventory holding of item  $i$
- $y_{ij}$ : production amount of item  $i$  in plant  $j$
- $c_{ij}$ : production cost per item  $i$  in plant  $j$
- $D_i$ : demand per period for the  $i$ th item
- $t_i$ : unit selling price of item  $i$
- $h_i$ : unit holding cost for the  $i$ th item
- $p_i$ : unit shortage cost for the  $i$ th item
- $k_i$ : unit holding area for the  $i$ th item
- $R$ : all available common resources
- $R_j$ : special resource control of plant  $j$
- $K$ : available storage space
- $B$ : available total budgetary costs

$CS(D_i, x_i)$ : total costs including shortage costs and storage costs for item  $i$

$P(D_i, x_i)$ : profit for item  $i$ .

For the  $i$ th item, some assumptions are introduced as follows.

- (i) The time horizon is infinite.
- (ii) The production rate is instantaneous.
- (iii) The lead time is zero.
- (iv) There is no fixed ordering cost and there is no initial inventory.
- (v) The demand  $D_i$  of the  $i$ th item in a cycle is a normally distributed random variable with a mean  $\mu_i$  and a variance  $\sigma_i^2$ .
- (vi) The shortage cost depends on shortage level, that is, for the  $i$ th item, the shortage cost is  $p_i \cdot \max\{0, D_i - x_i\}$ .
- (vii) The holding cost is dependent on the inventory, that is, for the  $i$ th item, the holding cost is  $h_i \cdot \max\{0, x_i - D_i\}$ .

**2.2. Model Formulation.** As shown in Figure 1, the whole decision process includes two sections for some market-oriented commodities. As the upper level, the retailer has to consider the optimal inventory level to satisfy customer's demand and achieve the maximal profit. However, it is scarcely possible to accurately estimate the customer's demand, especially in some holidays. Hence, the market information usually impacts the retailer's decision on how much the inventory level should be and further impacts her/his order quantity. As the lower level, the producer has to make the production plan according to the retailers' feedback due to the fierce competition. The optimal production quantity satisfying the retailer's demand and some resource limitations should be considered. To quickly grab the market share, the producer usually considers the minimal production as the first objective. Above all, the retailer and producer alternatively interact with each other by transferring the inventory and production information. Therefore, a bilevel model, in which the production is considered the lower level and the retailer is considered the lower level, is mathematically formulated to find the optimal production and inventory strategy.

**2.2.1. Lower Level.** Production should satisfy the following constraints and achieve the following objective.

- (i) All producers share a common resource:

$$\sum_{i=1}^n \sum_{j=1}^m y_{ij} \leq R. \tag{1}$$

- (ii) Some special resources may be controlled:

$$\sum_{i=1}^n y_{ij} \leq R_j, \quad j = 1, 2, \dots, m. \tag{2}$$

- (iii) Since the initial inventory amount is 0, the total quantity of ordered items should be more than or equal to the inventory holding for item  $i$ :

$$\sum_{j=1}^m y_{ij} \geq x_i, \quad i = 1, 2, \dots, n. \tag{3}$$

- (iv) The objective of production is to minimize production costs which is formulated as follows:

$$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} y_{ij}. \tag{4}$$

**2.2.2. Upper Level.** The retailer should satisfy the following constraints and achieve the following objective.

- (i) Inventory level is limited by overall capacity:

$$\sum_{i=1}^n k_i x_i \leq K. \tag{5}$$

- (ii) Inventory holdings should be more than or equal to 0:

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \tag{6}$$

- (iii) When the demand is  $D_i$  for the  $i$ th item, the real amount of sold items is

$$\min \{D_i, x_i\} = \begin{cases} D_i, & \text{if } D_i < x_i \\ x_i, & \text{if } D_i \geq x_i. \end{cases} \tag{7}$$

The overall cost  $CS(D_i, x_i)$  ( $i = 1, 2, \dots, n$ ) (including shortage cost and storage cost) of every item is calculated as follows:

$$CS(D_i, x_i) = p_i \max\{0, D_i - x_i\} + h_i \max\{0, x_i - D_i\}. \tag{8}$$

Because of the random demand  $D_i$ , the decision maker usually expect the average total budgetary level to be less than the predetermined level, so the following constraint is obtained:

$$E \left[ \sum_{i=1}^n CS(D_i, x_i) \right] \leq B. \tag{9}$$

- (iv) Similarly, the retailer's objective is to maximize average overall profit which consists of expected sales income minus expected overall costs, so it follows that

$$\max E \left[ \sum_{i=1}^n P(D_i, x_i) \right] \tag{10}$$

$$= \sum_{i=1}^n (t_i E[\min\{D_i, x_i\}] - E[CS(D_i, x_i)]).$$

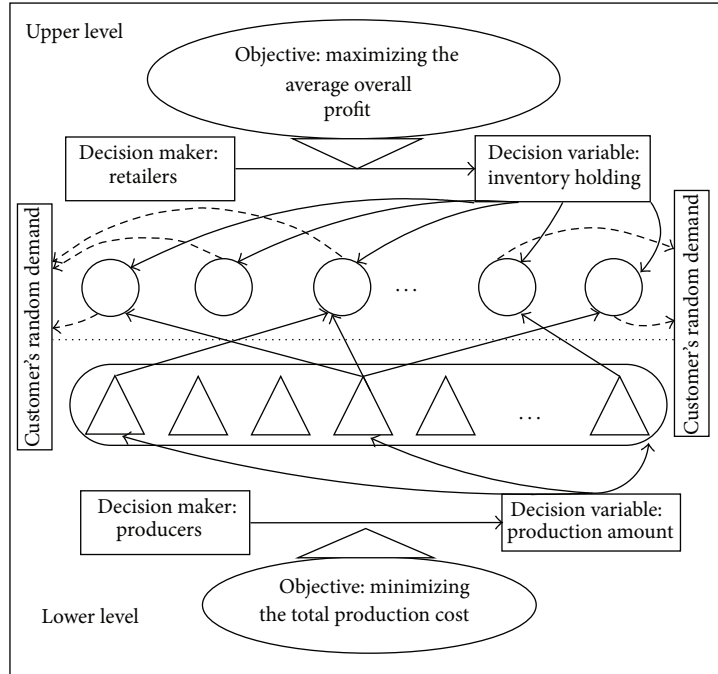


FIGURE 1: Structure of a production-inventory system.

Above all, the problem is defined as a bilevel optimization model as follows:

$$\begin{aligned}
 & \max_{x_i} \sum_{i=1}^n (t_i E[\min\{D_i, x_i\}] - E[CS(D_i, x_i)]) \\
 & \text{s.t.} \left\{ \begin{array}{l} \sum_{i=1}^n k_i x_i \leq K \\ E\left[\sum_{i=1}^n CS(D_i, x_i)\right] \leq B \\ x_i \geq 0, \quad i = 1, 2, \dots, n \\ \min_{y_{ij}} \sum_{i=1}^n \sum_{j=1}^m c_{ij} y_{ij} \\ \left\{ \begin{array}{l} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \leq R \\ \sum_{i=1}^n y_{ij} \leq R_j \\ \sum_{j=1}^m y_{ij} \geq x_i \\ y_{ij} \geq 0, \quad j = 1, 2, \dots, m. \end{array} \right. \end{array} \right. \quad (11)
 \end{aligned}$$

### 3. Approximate the Constraint Using a Rough Set

In problem (11), although the objective function is changed into a crisp function, the constraints still include random

coefficients. If we still deal with these using an expected value operator, it may result in a loss of information in a realistic production-inventory system. Here, we apply the rough approximation technique to deal with the stochastic programming problems. Some basic definitions and properties of the rough set are outlined in the following.

*Definition 1* (Pawlak [11]). Let  $U$  be a universe, and  $X$  a set representing a concept. Then its lower approximation is defined by

$$\underline{X} = \{x \in U \mid R^{-1}(x) \subset X\}, \quad (12)$$

and the upper approximation is defined by

$$\overline{X} = \bigcup_{x \in X} R(x), \quad (13)$$

where  $R$  is the equivalence relationship on  $U$ . Obviously, we have  $\underline{X} \subseteq X \subseteq \overline{X}$ .

Slowinski and Vanderpooten [31] extended the equivalence relation to a more general case and proposed a binary similarity relation that does not encompass symmetry and transitivity but reflexivity. Different from the equivalence relation, the similarity relation does not generate partitions on  $U$ ; for example, the similarity relation defined on  $R$  as “ $x$  is similar to  $y$  if and only if  $|x - y| \leq 1$ .”

*Example 2.* Consider the random event  $I = \{(x, y) \mid (x - \xi)^2 + y^2 \leq 1\}$ , where  $\xi \sim U(0, 1.5)$  is a uniformly distributed variable. If  $\xi_i$  takes the value in  $A_i$  with a probability  $p_i$ ,

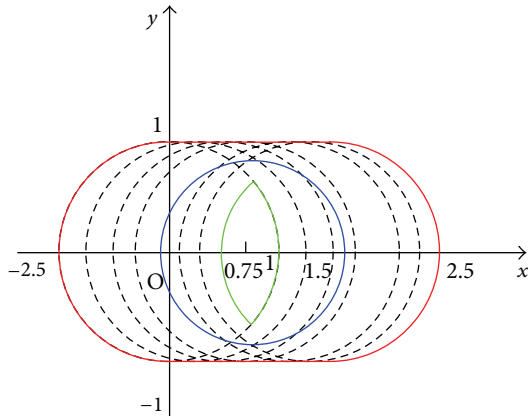


FIGURE 2: Random set approximated using the rough set.

let us define the similarity relationship  $R$  on the universe  $X$  satisfying

$$\bar{I}_R = \bigcup_{a \in [0, 1.5]} \{(x, y) \mid (x - a)^2 + y^2 \leq 1\}, \quad (14)$$

where  $\bar{I}_R$  is the upper approximation of the feasible region  $I$ , and its lower approximation is defined as follows:

$$\underline{I}_R = \bigcap_{a \in [0, 1.5]} \{(x, y) \mid (x - a)^2 + y^2 \leq 1\}. \quad (15)$$

Let  $I_E = \{(x, y) \mid (x - 3/4)^2 + y^2 \leq \sqrt{13}/4\}$  represent the expected feasible region dealt with by expected operator. As shown in Figure 2, the red circle and its interior represent the upper approximation  $\bar{I}_R$ , the green circle and its interior represent the lower approximation  $\underline{I}_R$ , and the blue circle and its interior represent the expected feasible region. Obviously, when dealing with the feasible region using expected operator, some information is lost. Therefore, it is not appropriate to apply this technique to deal with the feasible region, especially, when a decision needs to be made quickly, so a flexible method is more suitable.

In order to determine the upper and lower approximation degree describing set  $X$ , the concept of approximation *accuracy* was proposed by Pawlak [11]:

$$\alpha_R(X) = \frac{|\underline{X}|}{|\bar{X}|}, \quad (16)$$

where  $X \neq \Phi$  and  $|X|$  expresses the cardinal number of set  $X$  when  $X$  is a finite set; otherwise, it expresses the Lebesgue measure.

Another ratio defines *quality* of  $X$  using the attributes of  $R$  according to Pawlak [11]:

$$\gamma_R(X) = \frac{|\underline{X}|}{|X|}. \quad (17)$$

The quality  $\gamma_R(X)$  represents the relative frequency of the objects correctly classified using the attributes of  $R$ .

Many scholars consider the detailed statistical information of the overlap of an equivalence class and define the probabilistic approximation operators as follows [32]:

$$\begin{aligned} \underline{I}_\alpha &= \{x \in X \mid \Pr\{I \mid [x]\} \geq \alpha\} \\ \bar{I}_\beta &= \{x \in X \mid \Pr\{I \mid [x]\} > \beta\}, \end{aligned} \quad (18)$$

where  $0 \leq \beta < \alpha \leq 1$ , and  $[x]$  expresses the equivalent class of  $x$ . We usually take the condition  $0.5 < \alpha \leq 1$  and  $\beta = 1 - \alpha$  to determine the dual properties. After being dealt with like this, the unclear set  $I$  is converted into two crisp sets. According to Slowinski and Vanderpooten [31], we can extend these to the expected approximation under the following similarity relationship  $R_\epsilon$  for the function  $f(x, \xi)$  with random coefficient  $\xi$ :

$$xR_\epsilon y \iff E[|f(y, \xi) - f(x, \xi)|] \leq \epsilon, \quad (19)$$

where  $x, y \in X$ , and  $\epsilon$  is the deviation that the decision maker permits. In fact, the following relationship  $E[|f(x, \xi) - f(x, \xi)|] = 0 \leq \epsilon$  must hold, so  $R_\epsilon$  has reflexivity. It is apparent that it has symmetry, so  $R_\epsilon$  can be said to have a similarity relationship. The determination of the programming problem solution for the converted feasible set requires two steps. The first step is a requirement that the lower and upper approximation sets should be close enough to the initial feasible set. This means that we must guarantee the accuracy of the approximation to enable extension at some levels. The second step is that the programming problem in the lower and upper approximation sets is directly solved. This is guaranteed by the property  $\underline{I} \subseteq \bar{I}$ . Youness [14] applied rough approximation to deal with a crisp feasible set, so we can deal with the random feasible set using the expected approximation operator. Similarly, after ensuring that the required accuracy is achieved, only the programming problem in the upper approximation can be solved.

Let  $I = \{x \mid \sum_{i=1}^n CS(D_i, x_i) \leq B, x \geq 0\}$ ,  $\alpha(I)$  be the confidence level, that is, the accuracy of the approximation and  $\theta (0 \leq \theta \leq 1)$  be the accuracy the decision maker requires. Taking  $\alpha(I) = \theta$ , we can determine the value of the deviation  $\epsilon$  according to the confidence level  $\theta$ . As the accuracy  $\theta$  increases, the deviation  $\epsilon$  decreases, and the feasible region created by the upper approximation is close to the one determined using the expected value operator. This is especially true if  $\theta = 1$  as we get  $\underline{I} = I = \bar{I}$ . Therefore, if decision makers make a plan which includes upcoming holidays in the production-inventory system, they could decrease the confidence level  $\theta$  in order to cope with increasing demand.

If the demand is subject to a normal distribution, we get the following theorem.

**Theorem 3.** Assume that  $D_i$  is the normally distributed random variable with mean vector  $\mu_i$  and positive definite variance  $\sigma_i^2$ , written as  $D_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ , and they are independent of



each other. The similarity relationship  $R_\varepsilon$  is therefore defined as follows:

$$\varsigma R_\varepsilon \zeta \iff E \left[ \left| \sum_{i=1}^n \text{CS}(D_i, \varsigma_i) - \sum_{i=1}^n \text{CS}(D_i, \zeta_i) \right| \right] \leq \varepsilon, \quad (20)$$

where  $\varsigma$  and  $\zeta$  are any two vectors in  $X^n$ , and  $\varepsilon$  is the deviation. Assume that  $\theta$  ( $0 \leq \theta \leq 1$ ) is the confidence level, and then problem (11) can be converted into the following model under the similarity relationship  $R_\varepsilon$ :

$$\begin{aligned} & \max H(x) \\ & = \frac{1}{2} \sum_{i=1}^n \left\{ (t_i - p_i - h_i) \right. \\ & \quad \times \left( 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-(x_i - \mu_i)^2 / 2\sigma_i^2} \right] \right) \\ & \quad \left. + [t_i(\mu_i + x_i) + (\mu_i - x_i)(h_i - p_i)] \right\} \\ & \text{s.t.} \left\{ \begin{array}{l} \sum_{i=1}^n k_i x_i \leq K \\ \frac{1}{2} \sum_{i=1}^n \left[ (p_i + h_i) \left( 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-(x_i - \mu_i)^2 / 2\sigma_i^2} \right] \right) \right] \\ \leq B - \frac{1}{2} \sum_{i=1}^n (h_i - p_i)(x_i - \mu_i) + (\varepsilon)_0 \\ x_i \geq 0, \quad i = 1, 2, \dots, n \\ \min_{y_{ij}} \sum_{i=1}^n \sum_{j=1}^m c_{ij} y_{ij} \\ \left\{ \begin{array}{l} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \leq R \\ \sum_{i=1}^n y_{ij} \leq R_j \\ \sum_{j=1}^m y_{ij} \geq x_i \\ y_{ij} \geq 0, \quad j = 1, 2, \dots, m. \end{array} \right. \end{array} \right. \end{aligned} \quad (21)$$

*Proof.* Firstly, cope with the objective function using the expected value operator. Since  $D_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$  is a normally distributed random variable with a mean vector  $\mu_i$  and a positive definite variance  $\sigma_i^2$ , it follows that  $\min\{D_i, x_i\}$  and  $\text{CS}(D_i, x_i)$  are both random variables. For the function  $\min\{D_i, x_i\}$ , it follows that

$$\begin{aligned} E[\min\{D_i, x_i\}] &= E \left[ \frac{1}{2} (D_i + x_i - |D_i - x_i|) \right] \\ &= \frac{1}{2} (E[D_i] - E[|D_i - x_i|] + x_i) \quad (22) \\ &= \frac{1}{2} (E[|D_i - x_i|] + \mu_i + x_i). \end{aligned}$$

It follows from the definition of the random variable that for  $x_i \geq 0$

$$\begin{aligned} E[|D_i - x_i|] &= \int_{-\infty}^{+\infty} \frac{|t - x_i|}{\sqrt{2\pi}\sigma} e^{-(t - \mu_i)^2 / 2\sigma^2} dt \\ &= \int_{-\infty}^{x_i} \frac{x_i - t}{\sqrt{2\pi}\sigma} e^{-(t - \mu_i)^2 / 2\sigma^2} dt + \int_{x_i}^{+\infty} \frac{t - x_i}{\sqrt{2\pi}\sigma} e^{-(t - \mu_i)^2 / 2\sigma^2} dt \\ &= 1 + 2 \int_{-\infty}^{x_i} \frac{x_i - t}{\sqrt{2\pi}\sigma} e^{-(t - \mu_i)^2 / 2\sigma^2} dt \\ &= 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-(x_i - \mu_i)^2 / 2\sigma_i^2} \right], \quad (23) \end{aligned}$$

where  $\Phi$  is the standard normally distributed function. Similarly, we have

$$\begin{aligned} E[\text{CS}(D_i, x_i)] &= p_i E[\max\{0, D_i - x_i\}] + h_i E[\max\{0, x_i - D_i\}] \\ &= \frac{1}{2} (p_i + h_i) E[|D_i - x_i|] + \frac{1}{2} (h_i - p_i)(x_i - \mu_i). \quad (24) \end{aligned}$$

From the above, the objective function in the upper level can be converted into

$$\begin{aligned} & \max H(x) \\ & = \frac{1}{2} \sum_{i=1}^n \left\{ (t_i - p_i - h_i) \right. \\ & \quad \times \left( 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-(x_i - \mu_i)^2 / 2\sigma_i^2} \right] \right) \\ & \quad \left. + [t_i(\mu_i + x_i) + (\mu_i - x_i)(h_i - p_i)] \right\}. \quad (25) \end{aligned}$$

Secondly, we deal with the constraint  $E[\sum_{i=1}^n \text{CS}(D_i, x_i)] \leq B$  using rough approximation. Let  $I = \{x \mid \sum_{i=1}^n \text{CS}(D_i, x_i) \leq B, x_i \geq 0, i = 1, 2, \dots, n\}$ , so then it follows from the similarity relationship  $R_\varepsilon$  that,

$$\begin{aligned} \underline{I} &= \left\{ x \mid \sum_{i=1}^n E[\text{CS}(D_i, x_i)] \leq B - \varepsilon, x \geq 0 \right\} \\ \bar{I} &= \left\{ x \mid \sum_{i=1}^n E[\text{CS}(D_i, x_i)] \leq B + \varepsilon, x \geq 0 \right\}. \quad (26) \end{aligned}$$

It is known that  $|\cdot|$  expresses the cardinality of a set in the finite universe. For the infinite universe, we use this to express the Lebesgue measure. Then we have

$$\begin{aligned} |\underline{I}| &= \int \int \cdots \int_{\underline{I}} dx_1 dx_2 \cdots dx_n, \\ |\bar{I}| &= \int \int \cdots \int_{\bar{I}} dx_1 dx_2 \cdots dx_n. \end{aligned} \tag{27}$$

From the confidence level  $\theta$ , there must be a value  $(\varepsilon)_0$  such that  $\alpha(I) = \theta$ , so, for risk takers, the constraint can be converted into

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^n (p_i + h_i) \\ &\times \left( 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-(x_i - \mu_i)^2 / 2\sigma^2} \right] \right) \\ &\leq B - \frac{1}{2} \sum_{i=1}^n (h_i - p_i) (x_i - \mu_i) + (\varepsilon)_0. \end{aligned} \tag{28}$$

Taking these into problem (11) and we can get the transformed model. This completes the proof.  $\square$

### 4. Solution Approach

For bilevel programming (21), there are two key problems: (i) Does an optimal solution exist? (ii) What is the global optimal solution if the lower level has multiple optimal solutions? Firstly, some scholars [20, 21] apply the Kuhn-Tucker conditions to judge whether a global optimal solution exists and then converts the bilevel model into a single level. If the optimal solution for problem (21) does not exist, there is no equilibrium between the producers and the retailers. From the view of the retailer, they can just make a decision without considering the cost control of the producer since the retailer is mightier than the producers. If the optimal solution for problem (21) does not exist, this means that the producers and the retailers cannot reach an agreement to achieve their respective objectives. Secondly, if the lower level has multiple optimal solutions, this means that the producers have multiple production strategies. All Pareto optimal strategies are taken from the lower level to the upper level, and then a Pareto optimal strategy can be found by comparing the retailers' profits. Finding an optimal solution for the complete system is discussed in the following section.

Two steps can be implemented to solve the bilevel crisp programming problem (21). Firstly, we apply an interactive programming technique to convert it into a single level. Secondly, the genetic algorithm is applied to compute the nonlinear programming problem generated by the first step.

**4.1. Interactive Programming Technique.** The interactive programming technique proposed in [22, 23, 33] is usually applied to solve bilevel programming problems. Problem (21) is taken as an example to illustrate this method. We take the uncertain objective function to evaluate the decision maker's

imprecise considerations. For the each level's objective function in problem (21), the decision maker has fuzzy goals such as "the goal should be more than or equal to a certain value."

Let  $H_1(x, y) = H(x)$ ,  $H_2(x, y) = \sum_{i=1}^n \sum_{j=1}^m c_{ij} y_{ij}$  in problem (21), where  $x = (x_1, x_2, \dots, x_n)$ ,  $y = (y_1, y_2, \dots, y_m)$ . We can, respectively, denote the maximum and minimum values of each objective functions as follow:

$$\begin{aligned} H_1^1 &= \max_{x \in X, y \in Y} H_1(x, y), & H_2^1 &= \max_{x \in X, y \in Y} H_2(x, y), \\ H_1^0 &= \min_{x \in X, y \in Y} H_1(x, y), & H_2^0 &= \min_{x \in X, y \in Y} H_2(x, y). \end{aligned} \tag{29}$$

The functions  $\mu_i(H_i(x, y))$  change between  $H_i^0$  and  $H_i^1$ ,  $i = 1, 2$ . Take the linear function to characterize the objective at each level, which is defined as follows

$$\begin{aligned} \mu_1(H_1(x, y)) &= \begin{cases} 1, & H_1(x, y) \geq H_1^1 \\ \frac{H_1(x, y) - H_1^0}{H_1^1 - H_1^0}, & H_1^0 \leq H_1(x, y) < H_1^1 \\ 0, & H_1(x, y) < H_1^0, \end{cases} \\ \mu_2(H_2(x, y)) &= \begin{cases} 0, & H_2(x, y) \geq H_2^1 \\ \frac{H_2^1 - H_2(x, y)}{H_2^1 - H_2^0}, & H_2^0 \leq H_2(x, y) < H_2^1 \\ 1, & H_2(x, y) < H_2^0. \end{cases} \end{aligned} \tag{30}$$

The retailer level is specified with a maximal satisfactory level  $\varepsilon \in [0, 1]$ . Then the production level maximizes the function such that  $\mu_1(H_1(x, y)) \leq \varepsilon$ . This means that the production level should solve the following problem:

$$\begin{aligned} \max \quad & \mu_2(H_2(x, y)) \\ \text{s.t.} \quad & \begin{cases} \mu_1(H_1(x, y)) \leq \varepsilon \\ \sum_{i=1}^n \sum_{j=1}^m y_{ij} \leq R \\ \sum_{i=1}^n y_{ij} \leq R_j \\ \sum_{j=1}^m y_{ij} \geq x_i \\ y_{ij} \geq 0. \end{cases} \end{aligned} \tag{31}$$

To obtain a global satisfactory optimal solution for both levels, the retailer level needs to be comprised of the production level at the satisfaction of the retailer level. Therefore, the satisfactory degree for both levels is defined as

$$\lambda = \min \{ \mu_1(H_1(x, y)), \mu_2(H_2(x, y)) \}, \tag{32}$$

and the problem (31) is converted into

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \begin{cases} \mu_1(H_1(x, y)) \geq \lambda \\ \mu_2(H_2(x, y)) \geq \lambda \\ \sum_{i=1}^n \sum_{j=1}^m y_{ij} \leq R \\ \sum_{i=1}^n y_{ij} \leq R_j \\ \sum_{j=1}^m y_{ij} \geq x_i \\ \sum_{i=1}^n k_i x_i \leq K \\ \frac{1}{2} \sum_{i=1}^n \left[ (p_i + h_i) \left( 1 + \sqrt{\frac{2}{\pi}} \left[ (x_i + \mu_i) \Phi \left( \frac{x_i - \mu_i}{\sigma_i} \right) + \sigma_i e^{-\frac{(x_i - \mu_i)^2}{2\sigma^2}} \right] \right) \right] \\ \leq B - \frac{1}{2} \sum_{i=1}^n (h_i - p_i)(x_i - \mu_i) + (\varepsilon)_0 \\ x_i \geq 0 \\ y_{ij} \geq 0. \end{cases} \end{aligned} \quad (33)$$

By solving problem (33), we determine the overall satisfactory solution for both levels. Next, the efficiency of the solutions to the problem (33) is proved.

**Theorem 4.** *If problem (33) has optimal solutions, solutions to problem (33) must also be the solutions to problem (21).*

*Proof.* We consider  $x = (x_1, x_2, \dots, x_n)$ ,  $y = (y_1, y_2, \dots, y_n)$ ,  $\lambda$  as decision making variables, where  $y_i = (y_{i1}, y_{i2}, \dots, y_{im})$ ,  $i = 1, 2, \dots, n$ . Let  $X^* = (x^*, y^*, \lambda^*)$  be an optimal solution for problem (33).  $X^*$  is subject to all constraints. Assume that  $X^*$  is not the optimal for problem (21). In the second level, there is a solution  $y' = (y'_1, y'_2, \dots, y'_n)$  such that

$$H_2(x, y') < H_2(x, y^*). \quad (34)$$

Then

$$\begin{aligned} & \frac{H_2^1 - H_2(x, y')}{H_2^1 - H_2^0} \\ & = \mu_2(H_2(x, y')) > \mu_2(H_2(x, y^*)) = \frac{H_2^1 - H_2(x, y^*)}{H_2^1 - H_2^0}. \end{aligned} \quad (35)$$

Similarly, in the first level, we know that optimal  $x^*$  is subject to the constraint  $\sum_{j=1}^m y_{ij}^* \geq x^*$  in the second level. If  $x^*$  is not the optimal solution for problem (21), there is  $x'$  such that  $H_1(x', y') > H_1(x^*, y^*)$ , and then

$$\begin{aligned} & \frac{H_1(x', y') - H_1^0}{H_1^1 - H_1^0} \\ & = \mu_1(H_1(x', y')) > \mu_1(H_1(x^*, y^*)) = \frac{H_1^1 - H_1(x^*, y^*)}{H_1^1 - H_1^0}. \end{aligned} \quad (36)$$

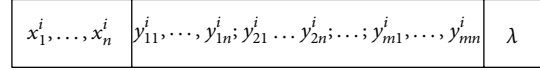


FIGURE 3: Structure of one chromosome.

From (32), (35), and (36), it can be concluded that the optimal solution  $\lambda'$  for variables  $x'$ ,  $y'$  must satisfy  $\lambda' < \lambda^*$ . This shows a conflict that  $\lambda^*$  is the optimal solution to problem (33). This completes the proof.  $\square$

**4.2. Genetic Algorithm.** Since the early 1960s, many researchers have had an increasing interest in efficient methods for solving complicated problems. There are many evolutionary computation methods which have been well developed. They include the following algorithms: genetic algorithms (designed by Holland [34]), evolutionary strategies (designed by Fogel [35]), and genetic programming (designed by Koza [36]). As a searching method, the genetic algorithm is the most powerful. For many years, significant research on the genetic algorithms has been discussed and summarized by several authors: Fonseca and Fleming [37], Holland [34], Michalewicz [38], Fogel [35], Goldberg [39], and Gen and Cheng [40]. In the following section, we apply the genetic algorithm to solve problem (33). This can be summarized as follows.

*Step 1 (representation).*

- (i) For this problem, we consider a solution  $X = (x, y, \lambda)$  to the problem as a chromosome, where  $x = (x_1, x_2, \dots, x_n)$ ,  $y = (y_{11}, \dots, y_{1n}; y_{21}, \dots, y_{2n}; \dots; y_{m1}, \dots, y_{mn})$ .
- (ii) Randomly choose one chromosome in the feasible region. Repeat the process  $N_{\text{pop-size}}$  times, and we get  $N_{\text{pop-size}}$  initial feasible chromosomes  $X^1, X^2, \dots, X^{N_{\text{pop-size}}}$  (see Figure 3).

*Step 2 (evaluation and selection).* In problem (33), the evaluation function can be given as follows:

$$\text{eval}(x, y, \lambda) = f(x, y, \lambda) = \lambda. \quad (37)$$

Then we apply the roulette wheel method to develop the selection process.

Each time we select a single chromosome for a new population in the following way. Compute the total probability  $q_i$ :

$$q_i = \sum_{j=1}^i \text{eval}^j, \quad i = 1, 2, \dots, N_{\text{pop-size}}. \quad (38)$$

Generate a random number  $r$  in  $[0, 1]$  and select the  $i$ th chromosome  $x^i$  such that  $q_{i-1} < r \leq q_i$ ,  $1 \leq i \leq N_{\text{pop-size}}$ . Repeat the above process  $N_{\text{pop-size}}$  times and we obtain  $N_{\text{pop-size}}$  copies of chromosomes.

*Step 3 (genetic operator).*

- (i) Crossover operation: to fully search all the possible feasible solutions in the feasible region, we cut



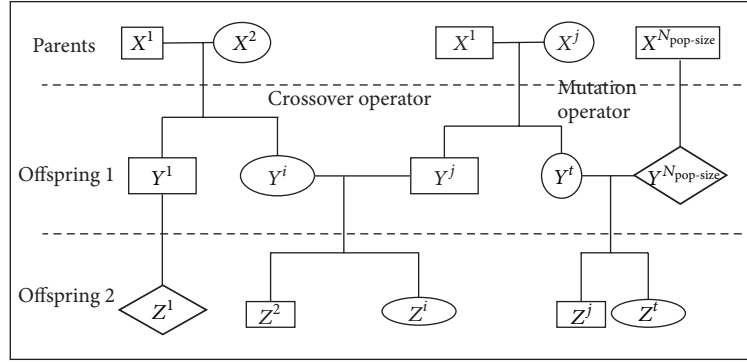


FIGURE 4: GA Illustration.

the chromosome into three sections and, respectively, carry out the crossover operation. Suppose the probability of the crossover operation in each section is, respectively,  $P_\alpha, P_\beta, P_\gamma$ . Randomly generate three numbers  $\alpha, \beta, \gamma$  from  $(0, 1)$ , and the chromosome  $X^i$  is selected as a parent if  $\alpha < P_\alpha, \beta < P_\beta$  and  $\gamma < P_\gamma$ . Repeat this  $N_{pop-size}$  times and get  $\min\{P_\alpha, P_\beta, P_\gamma\} \cdot N_{pop-size}$  chromosomes to operate the crossover. The crossover operator is produced as follows (see Figure 4):

$$\begin{aligned}
 Y^i &= (1 - \alpha) \begin{Bmatrix} x^i \\ 0 \\ 0 \end{Bmatrix} + \alpha \begin{Bmatrix} x^j \\ 0 \\ 0 \end{Bmatrix} + (1 - \beta) \begin{Bmatrix} 0 \\ y^i \\ 0 \end{Bmatrix} + \beta \begin{Bmatrix} 0 \\ y^j \\ 0 \end{Bmatrix} \\
 &+ (1 - \gamma) \begin{Bmatrix} 0 \\ 0 \\ \lambda^i \end{Bmatrix} + \gamma \begin{Bmatrix} 0 \\ 0 \\ \lambda^j \end{Bmatrix} \\
 Y^j &= \alpha \begin{Bmatrix} x^i \\ 0 \\ 0 \end{Bmatrix} + (1 - \alpha) \begin{Bmatrix} x^j \\ 0 \\ 0 \end{Bmatrix} + \beta \begin{Bmatrix} 0 \\ y^i \\ 0 \end{Bmatrix} + (1 - \beta) \begin{Bmatrix} 0 \\ y^j \\ 0 \end{Bmatrix} \\
 &+ \gamma \begin{Bmatrix} 0 \\ 0 \\ \lambda^i \end{Bmatrix} + (1 - \gamma) \begin{Bmatrix} 0 \\ 0 \\ \lambda^j \end{Bmatrix}.
 \end{aligned} \tag{39}$$

Check if the new chromosomes are feasible. If so, they are selected as the offspring to replace the parents.

- (ii) Mutation operation:  $P_m$  is the probability of the mutation operation. Generate a random number  $m$  from  $(0, 1)$ , and the chromosome  $x^j$  is selected as a parent to undergo the mutation operation provided that  $m < P_m$ . Repeat this  $N_{pop-size}$  times and get  $P_m \cdot N_{pop-size}$  chromosomes to operate the mutation. Take the chromosome  $X^i$  as a parent. Suppose that  $M$  is a sufficiently large positive number and randomly

select a mutation direction  $\mathbf{d} \in \mathbf{R}^n$ . Operate the following mutation operator (see Figure 4):

$$Y^i = \begin{Bmatrix} x^i \\ y^i \\ \lambda^i \end{Bmatrix} + M \cdot \begin{Bmatrix} x^i \\ y^i \\ \lambda^i \end{Bmatrix}. \tag{40}$$

Replace  $X^i$  with  $X^i + M \cdot \mathbf{d}$  if  $X^i + M \cdot \mathbf{d}$  is feasible. Otherwise adjust  $M$  until  $X^i + M \cdot \mathbf{d}$  is feasible.

### 5. Practical Application

In this section, an example of the inventory system of some goods in the Auchan supermarket in Chengdu is introduced. Auchan is a global company with more than 186,000 staff worldwide and nearly 1,200 stores and wholesale clubs across 12 countries. By August 31, 2009, the company had 211 oversize stores, 300 supermarkets, and 600 convenience stores. Internationally, the company operates units in Argentina (12), Brazil (294), Canada (278), China (63), Costa Rica (133), Germany (85), Guatemala (122), Honduras (37), Japan (391), Mexico (828), Nicaragua (36), Puerto Rico (54), El Salvador (59), South Korea (16), and the United Kingdom (323).

To improve its sales, Auchan stores investigate all factors which might impact the market. The inventory system is an important factor impacting the costs and profits of a store, as the efficiency of an inventory system determines the retailer's profit. In this section, 5 items provided by 2 producers have been selected. Table 1 shows the sales data for the 5 items in one period. Some parameters are certain in the period and they are easily incorporated into the computational method. As is known, the sales amount (demand) of all items changes every day. However, sales in another period can be predicted using historical data. Ordinarily, demand  $D_i$  follows a stochastic distribution, for example, the normal distribution at a certain time. However, if there is a holiday in the next period, the sales volume sharply increases, so will not satisfy demand if the decision maker makes orders using the former method. Hence, decision makers have to lower some limitations to face the increase in sales volume. For every group (Pickles, Oils, Sauces), three representatives which sell well are picked up and their parameters (including

TABLE 1: Parameters of the random demand  $D_i$ .

$\lambda_i$	Item 1	Item 2	Item 3	Item 4	Item 5
$\mu_i$	50	70	100	40	60
$\sigma_i^2$	4	6	9	2	4

TABLE 2: Other parameters in the production-inventory system.

Product	Item 1	Item 2	Item 3	Item 4	Item 5
Unit selling price $t_i$	100	80	60	85	100
Unit storage cost $h_i$	10	6	12	8	10
Unit shortage cost $p_i$	40	35	30	45	50
Unit storage area $k_i$	5	8	6	5	8
Production cost in plant $j$	15/12	8/10	4/4	10/9	14/15

TABLE 3: Optimal solutions under different confidence levels.

Variable	Item 1	Item 2	Item 3	Item 4	Item 5
$\theta = 0.85$	$F_U = 35148.36, F_L = 5667.43$				
$x$	182.5	0.54	1.62	0.84	0.073
$y_{1j}$	210.7	0	0.58	0.84	6.29
$y_{2j}$	0	0.56	0	0	480.51
$\theta = 0.9$	$F_U = 41916.68, F_L = 7140.244$				
$x$	190.51	0.15	0.87	0.38	0.13
$y_{1j}$	307.81	0	0.481	0.38	4.48
$y_{2j}$	0	0.61	0	0	489.71
$\theta = 0.95$	$F_U = 49872.31, F_L = 8451.456$				
$x$	198.05	0.058	0.42	0.33	0.081
$y_{1j}$	395.29	0	0.409	0.33	2.58
$y_{2j}$	0	0.42	0	0	495.37
$\theta = 1$	$F_U = 50132.57, F_L = 7725.394$				
$x$	199.95	0.032	0.35	0.12	0.057
$y_{1j}$	394.39	0	0.345	0.12	2.231
$y_{2j}$	0	0.348	0	0	498.45

Note:  $F_U$  and  $F_L$  denote objective values of the upper level and lower level, respectively.  $y_{1j}$  and  $y_{2j}$  denote the production quantity by producer 1 and produce 2, respectively.

shortage cost, set up cost, holding cost, and production cost) are listed in Table 1. Other parameters are in Table 2. The common resource is 1000. The special resource controls for plants 1 and 2 are 400 and 500, respectively. Available storage space is 1000 and available total budgetary cost is 20000, and the confidence level is 0.85.

From problem (21) in Theorem 3, the crisp model is obtained. Then an interactive programming technique and a GA are applied to solve this problem. After 5000 cycles, the optimal solution is obtained for the above problems as shown in Table 3. The decision maker can adjust the satisfaction level according to her/his different purposes to obtain a more optimal strategy.

**5.1. Sensitivity Analysis.** The decision maker can adjust the parameter to obtain different solutions. From deduction, it is found that accuracy is the key factor impacting the results.

If the accuracy  $\theta$  decreases, the feasible set expands, so then a better optimal solution can be determined. From Table 3, it can be seen that, with an increase in  $\theta$ , the objective function at the upper level increases and the objective function at the lower level decreases, which is consistent with theoretical analysis. This means that if the next holiday is long, the sales volume sharply increases, so the decision maker must decrease the accuracy to increase the retailer's profit and reduce the producers' cost.

Furthermore, if the demand of a certain item sharply increases in the next period, for example, oil consumption during Spring Festival holidays, the weight of the oil must be increased. From the second column in Table 3, it is known that the objective values and order amounts of the second kind of items increase as their weights increase. This leads to the maximum probability that the inventory cost of oil is less than the predetermined value; that is, although the oil order quantities increase, the inventory does not increase because of the large demand.

**5.2. Comparison Analysis.** Other research also introduces other approaches to solve bilevel programming with crisp parameters. For example, some scholars [41] have introduced the Stackelberg-Nash equilibrium for multilevel programming and have designed a genetic algorithm to obtain the optimal solution. This can be solved using the above algorithm to obtain the optimal solution as shown in Table 4. From this, it can be seen that the optimal solutions in the two tables are similar, and thus it can be concluded that the two methods are aimed at different problems. The interactive programming technique is efficient and convenient to those problems which contain fewer variables and constraints and more objectives, while the hybrid intelligent algorithm is efficient and convenient for those problems which have more variables and fewer objectives. Therefore, the decision maker is able to choose an efficient method according to their the real problems.

The expected operator is also usually used to deal with a constraint with random coefficients [9]. This problem is solved by using the constraint  $\sum_{i=1}^n E[CS(D_i, x_i)] \leq B$ , that is, at a confidence level  $\theta = 1$ , and by using a fuzzy programming technique, both of which are listed in the first row in Table 3. As can be seen, the objective function at the upper level is smaller and the objective function at the lower level is larger than those obtained using rough approximation. The reason is that, the constraint  $\sum_{i=1}^n E[CS(D_i, x_i)] \leq B$  requires that the order amount strictly satisfies the above condition under average demand. This results in a lack of orders when facing the next holiday under the former inventory level.

## 6. Conclusions

In this paper, we have discussed a bilevel programming model with random coefficients and its application to production-inventory systems. We also converted it into a crisp model using an expected value operator and rough approximation. Then an interactive programming technique and a GA have been applied to solve this bilevel programming problem.

TABLE 4: Optimal solution by hybrid intelligent algorithm.

Variable														
$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$y_{11}$	$y_{21}$	$y_{31}$	$y_{41}$	$y_{51}$	$y_{12}$	$y_{22}$	$y_{32}$	$y_{42}$	$y_{52}$
197.96	0.050	0.42	0.12	0.055	399.05	0.035	0.51	0.124	3.05	0	0.351	0.032	0	499.62

Finally, an application was exhibited to show the efficiency of the proposed model and algorithm.

Although the model constructed in this paper should be helpful for solving some real problems, it is only dealt with using the expected value and rough approximation. If the decision maker has different purposes, such as minimizing budgetary costs or maximizing the probability that all costs are lower than the budget, the chance constraint or dependent chance techniques can be applied. In further research to be under taken, a more complete detailed analysis will be given.

## Acknowledgments

The work is supported by the Key Program of National Natural Science Foundation of China (Grant no. 70831005), the "985" Program of Sichuan University (Innovative Research Base for Economic Development and Management), the Philosophy and Social Sciences Planning Project of Sichuan Province (Grant no. SC12BJ05), and the Initial Funding for Young Teachers of Sichuan University (Grant no. 2013SCU11014).

## References

- [1] O. E. Emam, "A fuzzy approach for bi-level integer non-linear programming problem," *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 62–71, 2006.
- [2] G. B. Dantzig and A. Madansky, "On the solution of two-stage linear programs under uncertainty," in *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, Statistical Laboratory, University of California, June 1960.
- [3] C. Liu, Y. Fan, and F. Ordóñez, "A two-stage stochastic programming model for transportation network protection," *Computers & Operations Research*, vol. 36, no. 5, pp. 1582–1590, 2009.
- [4] J. Y. Jung, G. Blau, J. F. Pekny, G. V. Reklaitis, and D. Eversdyk, "Integrated safety stock management for multi-stage supply chains under production capacity constraints," *Computers and Chemical Engineering*, vol. 32, no. 11, pp. 2570–2581, 2008.
- [5] T. Paksoy and C.-T. Chang, "Revised multi-choice goal programming for multi-period, multi-stage inventory controlled supply chain model with popup stores in Guerrilla marketing," *Applied Mathematical Modelling*, vol. 34, no. 11, pp. 3586–3598, 2010.
- [6] H. Sun, Z. Gao, and J. Wu, "A bi-level programming model and solution algorithm for the location of logistics distribution centers," *Applied Mathematical Modelling*, vol. 32, no. 4, pp. 610–616, 2008.
- [7] E. Roghanian, S. J. Sadjadi, and M. B. Aryanezhad, "A probabilistic bi-level linear multi-objective programming problem to supply chain planning," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 786–800, 2007.
- [8] X. Ji and Z. Shao, "Model and algorithm for bilevel newsboy problem with fuzzy demands and discounts," *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 163–174, 2006.
- [9] A. Charnes and W. W. Cooper, "Deterministic equivalents for optimizing and satisficing under chance constraints," *Operations Research*, vol. 11, pp. 18–39, 1963.
- [10] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, NY, USA, 1997.
- [11] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [12] Z. Pawlak and R. Sowiński, "Rough set approach to multi-attribute decision analysis," *European Journal of Operational Research*, vol. 72, no. 3, pp. 443–459, 1994.
- [13] J. Xu and L. 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.
- [14] E. A. Youness, "Characterizing solutions of rough programming problems," *European Journal of Operational Research*, vol. 168, no. 3, pp. 1019–1029, 2006.
- [15] Y. Shi, L. Yao, and J. 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. 261–280, 2011.
- [16] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: a bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, 1994.
- [17] J. F. Bard and J. T. Moore, "A branch and bound algorithm for the bilevel programming problem," *Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 281–292, 1990.
- [18] O. Ben-Ayed and C. E. Blair, "Computational difficulties of Bi-level Linear Programming," *Operations Research*, vol. 38, no. 3, pp. 556–560, 1990.
- [19] W. F. Bialas and M. H. Karwan, "Two-level linear programming," *Management Science*, vol. 30, no. 8, pp. 1004–1020, 1984.
- [20] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *The Journal of the Operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.
- [21] S. Sinha and S. B. Sinha, "KKT transformation approach for multi-objective multi-level linear programming problems," *European Journal of Operational Research*, vol. 143, no. 1, pp. 19–31, 2002.
- [22] M. Sakawa, I. Nishizaki, and Y. Uemura, "Interactive fuzzy programming for multilevel linear programming problems," *Computers & Mathematics with Applications*, vol. 36, no. 2, pp. 71–86, 1998.
- [23] M. Sakawa, "Interactive fuzzy goal programming for nonlinear programming problems and its applications to water quality management," *Control and Cybernetics*, vol. 13, pp. 217–228, 1984.
- [24] S. Pramanik and T. K. Roy, "Fuzzy goal programming approach to multilevel programming problems," *European Journal of Operational Research*, vol. 176, no. 2, pp. 1151–1166, 2007.

- [25] S. R. Hejazi, A. Memariani, G. Jahanshahloo, and M. M. Sepehri, "Linear bilevel programming solution by genetic algorithm," *Computers and Operations Research*, vol. 29, no. 13, pp. 1913–1925, 2002.
- [26] K. H. Sahin and A. R. Ciric, "A dual temperature simulated annealing approach for solving bilevel programming problems," *Computers and Chemical Engineering*, vol. 23, no. 1, pp. 11–25, 1998.
- [27] M. Gendreau, P. Marcotte, and G. Savard, "A hybrid tabu-ascent algorithm for the linear bilevel programming problem," *Journal of Global Optimization*, vol. 8, no. 3, pp. 217–233, 1996.
- [28] S. F. Woon, V. Rehbock, and R. C. Loxton, "Global optimization method for continuous-time sensor scheduling," *Nonlinear Dynamics and Systems Theory*, vol. 10, no. 2, pp. 175–188, 2010.
- [29] S. F. Woon, V. Rehbock, and R. Loxton, "Towards global solutions of optimal discrete-valued control problems," *Optimal Control Applications & Methods*, vol. 33, no. 5, pp. 576–594, 2012.
- [30] D. D. Wu, "BiLevel programming data envelopment analysis with constrained resource," *European Journal of Operational Research*, vol. 207, no. 2, pp. 856–864, 2010.
- [31] R. Slowinski and D. Vanderpooten, "A generalized definition of rough approximations based on similarity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 2, pp. 331–336, 2000.
- [32] Y. Yao, "Probabilistic rough set approximations," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 255–271, 2008.
- [33] M. Sakawa, I. Nishizaki, and Y. Uemura, "Interactive fuzzy programming for multi-level linear programming problems with fuzzy parameters," *Fuzzy Sets and Systems*, vol. 109, no. 1, pp. 3–19, 2000.
- [34] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan, Ann Arbor, Mich, USA, 1975.
- [35] D. B. Fogel, *Evolution Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, USA, 1995.
- [36] J. R. Koza, *Genetic Programming*, The MIT Press, Cambridge, Mass, USA, 1992.
- [37] C. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, pp. 1–16, 1995.
- [38] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, NY, USA, 1994.
- [39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, NY, USA, 1989.
- [40] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, New York, NY, USA, 1997.
- [41] J. Gao and B. Liu, "Fuzzy multilevel programming with a hybrid intelligent algorithm," *Computers & Mathematics with Applications*, vol. 49, no. 9-10, pp. 1539–1548, 2005.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

