Chapter 2

Uncertain Programming

Uncertain programming was founded by Liu [122] in 2009 as a type of mathematical programming involving uncertain variables. This chapter provides a general framework of uncertain programming, including expected value model, chance-constrained programming, dependent-chance programming, uncertain dynamic programming and uncertain multilevel programming. Finally, we present some uncertain programming models for project scheduling problem, vehicle routing problem, and machine scheduling problem.

2.1 Ranking Criteria

Assume that $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is an uncertain vector, $f(\boldsymbol{x}, \boldsymbol{\xi})$ is a return function, and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ are constraint functions, $j = 1, 2, \cdots, p$. Let us examine

$$\begin{cases} \max f(\boldsymbol{x}, \boldsymbol{\xi}) \\ \text{subject to:} \\ \qquad g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, \quad j = 1, 2, \cdots, p. \end{cases} \tag{2.1}$$

Mention that the model (2.1) is only a conceptual model rather than a mathematical model because there does not exist a natural ordership in an uncertain world.

Thus an important problem appearing in this area is how to rank uncertain variables. Let ξ and η be two uncertain variables. Liu [122] gave four ranking criteria.

Expected Value Criterion: We say $\xi > \eta$ if and only if $E[\xi] > E[\eta]$.

Optimistic Value Criterion: We say $\xi > \eta$ if and only if, for some predetermined confidence level $\alpha \in (0, 1]$, we have $\xi_{\sup}(\alpha) > \eta_{\sup}(\alpha)$, where $\xi_{\sup}(\alpha)$ and $\eta_{\sup}(\alpha)$ are the α-optimistic values of ξ and η, respectively.

Pessimistic Value Criterion: We say $\xi > \eta$ if and only if, for some predetermined confidence level $\alpha \in (0, 1]$, we have $\xi_{\inf}(\alpha) > \eta_{\inf}(\alpha)$, where $\xi_{\inf}(\alpha)$ and $\eta_{\inf}(\alpha)$ are the α-pessimistic values of ξ and η, respectively.

Chance Criterion: We say $\xi > \eta$ if and only if, for some predetermined levels $\overline{r}$, we have $\mathcal{M}\{\xi \ge \overline{r}\} > \mathcal{M}\{\eta \ge \overline{r}\}$.

B. Liu: Uncertainty Theory: A Branch of Mathematics, SCI 300, pp. 81–113.
springerlink.com

2.2 Expected Value Model

Assume that we believe the expected value criterion. In order to obtain a decision with maximum expected return subject to expected constraints, we have the following expected value model,

$$\begin{cases} \max E[f(\boldsymbol{x}, \boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \;\; j = 1, 2, \cdots, p \end{cases} \tag{2.2}$$

where $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is an uncertain vector, f is a return function, and g_j are constraint functions for $j = 1, 2, \cdots, p$.

Definition 2.1. *A solution x is feasible if and only if $E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0$ for $j = 1, 2, \cdots, p$. A feasible solution $\boldsymbol{x}^*$ is an optimal solution to the expected value model (2.2) if $E[f(\boldsymbol{x}^*, \boldsymbol{\xi})] \ge E[f(\boldsymbol{x}, \boldsymbol{\xi})]$ for any feasible solution $\boldsymbol{x}$.*

In practice, a decision maker may want to optimize multiple objectives. Thus we have the following expected value multiobjective programming,

$$\begin{cases} \max\, [E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \cdots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]] \\ \text{subject to:} \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \;\; j = 1, 2, \cdots, p \end{cases} \tag{2.3}$$

where $f_i(\boldsymbol{x}, \boldsymbol{\xi})$ are return functions for $i = 1, 2, \cdots, m$, and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ are constraint functions for $j = 1, 2, \cdots, p$.

Definition 2.2. *A feasible solution $\boldsymbol{x}^*$ is said to be a Pareto solution to the expected value multiobjective programming (2.3) if there is no feasible solution $\boldsymbol{x}$ such that*

$$E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] \ge E[f_i(\boldsymbol{x}^*, \boldsymbol{\xi})], \quad i = 1, 2, \cdots, m \tag{2.4}$$

and $E[f_j(\boldsymbol{x}, \boldsymbol{\xi})] > E[f_j(\boldsymbol{x}^*, \boldsymbol{\xi})]$ for at least one index j.

In order to balance multiple conflicting objectives, a decision-maker may establish a hierarchy of importance among these incompatible goals so as to satisfy as many goals as possible in the order specified. Thus we have an expected value goal programming,

$$\begin{cases} \min \displaystyle\sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ \vee 0 + v_{ij} d_i^- \vee 0) \\ \text{subject to:} \\ \qquad E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] - b_i = d_i^+, \quad i = 1, 2, \cdots, m \\ \qquad b_i - E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] = d_i^-, \quad i = 1, 2, \cdots, m \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \qquad\quad j = 1, 2, \cdots, p \end{cases} \tag{2.5}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, $d_i^+ \vee 0$ is the positive deviation from the target of goal i, $d_i^- \vee 0$ is the negative deviation from the target of goal i, f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

Theorem 2.1. *Assume* $f(\boldsymbol{x},\boldsymbol{\xi}) = h_1(\boldsymbol{x})\xi_1 + h_2(\boldsymbol{x})\xi_2 + \cdots + h_n(\boldsymbol{x})\xi_n + h_0(\boldsymbol{x})$ *where* $h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_n(\boldsymbol{x}), h_0(\boldsymbol{x})$ *are real-valued functions and* $\xi_1, \xi_2, \cdots, \xi_n$ *are independent uncertain variables. Then*

$$E[f(\boldsymbol{x},\boldsymbol{\xi})] = h_1(\boldsymbol{x})E[\xi_1] + h_2(\boldsymbol{x})E[\xi_2] + \cdots + h_n(\boldsymbol{x})E[\xi_n] + h_0(\boldsymbol{x}). \quad (2.6)$$

Proof: It follows from the linearity of expected value operator immediately.

Theorem 2.2. *Assume that* $\xi_1, \xi_2, \cdots, \xi_n$ *are independent uncertain variables and* $h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_n(\boldsymbol{x}), h_0(\boldsymbol{x})$ *are real-valued functions. Then*

$$E[h_1(\boldsymbol{x})\xi_1 + h_2(\boldsymbol{x})\xi_2 + \cdots + h_n(\boldsymbol{x})\xi_n + h_0(\boldsymbol{x})] \le 0 \quad (2.7)$$

holds if and only if

$$h_1(\boldsymbol{x})E[\xi_1] + h_2(\boldsymbol{x})E[\xi_2] + \cdots + h_n(\boldsymbol{x})E[\xi_n] + h_0(\boldsymbol{x}) \le 0. \quad (2.8)$$

Proof: It follows from Theorem 2.1 immediately.

2.3 Chance-Constrained Programming

Since the uncertain constraints $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, p$ do not define a deterministic feasible set, it is naturally desired that the uncertain constraints hold with a confidence level α. Then we have a chance constraint as follows,

$$\mathcal{M}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, p\} \ge \alpha. \quad (2.9)$$

Maximax Chance-Constrained Programming

Assume that we believe the optimistic value criterion. If we want to maximize the optimistic value to the uncertain return subject to some chance constraints, then we have the following maximax chance-constrained programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \max\limits_{\overline{f}} \overline{f} \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{f(\boldsymbol{x},\boldsymbol{\xi}) \ge \overline{f}\right\} \ge \beta \\ \qquad \mathcal{M}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \alpha_j, \quad j = 1, 2, \cdots, p \end{cases} \quad (2.10)$$

where α_j and β are specified confidence levels for $j = 1, 2, \cdots, p$, and $\max \overline{f}$ is the β-optimistic return.

In practice, it is possible that there exist multiple objectives. We thus have the following maximax chance-constrained multiobjective programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \left[\max\limits_{\overline{f}_1} \overline{f}_1, \max\limits_{\overline{f}_2} \overline{f}_2, \cdots, \max\limits_{\overline{f}_m} \overline{f}_m \right] \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{ f_i(\boldsymbol{x}, \boldsymbol{\xi}) \ge \overline{f}_i \right\} \ge \beta_i, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\left\{ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \right\} \ge \alpha_j, \quad j = 1, 2, \cdots, p \end{cases} \tag{2.11}$$

where β_i are predetermined confidence levels for $i = 1, 2, \cdots, m$, and $\max \overline{f}_i$ are the β-optimistic values to the return functions $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, $i = 1, 2, \cdots, m$, respectively.

If the priority structure and target levels are set by the decision-maker, then we have a minimin chance-constrained goal programming,

$$\begin{cases} \min\limits_{\boldsymbol{x}} \sum\limits_{j=1}^{l} P_j \sum\limits_{i=1}^{m} \left(u_{ij} \left(\min\limits_{d_i^+} d_i^+ \vee 0 \right) + v_{ij} \left(\min\limits_{d_i^-} d_i^- \vee 0 \right) \right) \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{ f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+ \right\} \ge \beta_i^+, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\left\{ b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^- \right\} \ge \beta_i^-, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\left\{ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \right\} \ge \alpha_j, \qquad\quad j = 1, 2, \cdots, p \end{cases} \tag{2.12}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, $\min d_i^+ \vee 0$ is the β_i^+-optimistic positive deviation from the target of goal i, $\min d_i^- \vee 0$ is the β_i^--optimistic negative deviation from the target of goal i, b_i is the target value according to goal i, and l is the number of priorities.

Minimax Chance-Constrained Programming

Assume that we believe the pessimistic value criterion. If we want to maximize the pessimistic value subject to some chance constraints, then we have the following minimax chance-constrained programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \min\limits_{\overline{f}} \overline{f} \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{ f(\boldsymbol{x}, \boldsymbol{\xi}) \le \overline{f} \right\} \ge \beta \\ \qquad \mathcal{M}\left\{ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \right\} \ge \alpha_j, \quad j = 1, 2, \cdots, p \end{cases} \tag{2.13}$$

where α_j and β are specified confidence levels for $j = 1, 2, \cdots, p$, and $\min \overline{f}$ is the β-pessimistic return.

If there are multiple objectives, then we have the following minimax chance-constrained multiobjective programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \left[\min\limits_{\overline{f}_1} \overline{f}_1, \ \min\limits_{\overline{f}_2} \overline{f}_2, \ \cdots, \ \min\limits_{\overline{f}_m} \overline{f}_m \right] \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{ f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le \overline{f}_i \right\} \ge \beta_i, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\{ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \} \ge \alpha_j, \quad j = 1, 2, \cdots, p \end{cases} \tag{2.14}$$

where $\min \overline{f}_i$ are the β_i-pessimistic values to the return functions $f_i(\boldsymbol{x}, \boldsymbol{\xi})$, $i = 1, 2, \cdots, m$, respectively.

We can also formulate an uncertain decision system as a minimax chance-constrained goal programming according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min\limits_{\boldsymbol{x}} \sum\limits_{j=1}^{l} P_j \sum\limits_{i=1}^{m} \left[u_{ij} \left(\max\limits_{d_i^+} d_i^+ \vee 0 \right) + v_{ij} \left(\max\limits_{d_i^-} d_i^- \vee 0 \right) \right] \\ \text{subject to:} \\ \qquad \mathcal{M}\left\{ f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \ge d_i^+ \right\} \ge \beta_i^+, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\left\{ b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \ge d_i^- \right\} \ge \beta_i^-, \quad i = 1, 2, \cdots, m \\ \qquad \mathcal{M}\{ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \} \ge \alpha_j, \qquad\qquad j = 1, 2, \cdots, p \end{cases} \tag{2.15}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, $\max d_i^+ \vee 0$ is the β_i^+-pessimistic positive deviation from the target of goal i, $\max d_i^- \vee 0$ is the β_i^--pessimistic negative deviation from the target of goal i, b_i is the target value according to goal i, and l is the number of priorities.

Theorem 2.3. *Assume that $\xi_1, \xi_2, \cdots, \xi_n$ are independent uncertain variables with uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively, and $h_1(\boldsymbol{x})$, $h_2(\boldsymbol{x}), \cdots, h_n(\boldsymbol{x}), h_0(\boldsymbol{x})$ are real-valued functions. Then*

$$\mathcal{M}\left\{ \sum_{i=1}^{n} h_i(\boldsymbol{x})\xi_i \le h_0(\boldsymbol{x}) \right\} \ge \alpha \tag{2.16}$$

holds if and only if

$$\sum_{i=1}^{n} h_i^+(\boldsymbol{x})\Phi_i^{-1}(\alpha) - \sum_{i=1}^{n} h_i^-(\boldsymbol{x})\Phi_i^{-1}(1-\alpha) \le h_0(\boldsymbol{x}) \tag{2.17}$$

where

$$h_i^+(\boldsymbol{x}) = \begin{cases} h_i(\boldsymbol{x}), & \text{if } h_i(\boldsymbol{x}) > 0 \\ 0, & \text{if } h_i(\boldsymbol{x}) \le 0, \end{cases} \tag{2.18}$$

$$h_i^-(\boldsymbol{x}) = \begin{cases} 0, & \text{if } h_i(\boldsymbol{x}) \ge 0 \\ -h_i(\boldsymbol{x}), & \text{if } h_i(\boldsymbol{x}) < 0 \end{cases} \tag{2.19}$$

for $i = 1, 2, \cdots, n$. *Especially, if* $h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_n(\boldsymbol{x})$ *are all nonnegative, then (2.17) becomes*

$$\sum_{i=1}^{n} h_i(\boldsymbol{x})\Phi_i^{-1}(\alpha) \le h_0(\boldsymbol{x}); \tag{2.20}$$

if $h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_n(\boldsymbol{x})$ *are all nonpositive, then (2.17) becomes*

$$\sum_{i=1}^{n} h_i(\boldsymbol{x})\Phi_i^{-1}(1-\alpha) \le h_0(\boldsymbol{x}). \tag{2.21}$$

Proof: For each i, if $h_i(x) > 0$, then $h_i(\boldsymbol{x})\xi_i$ is an uncertain variable whose uncertainty distribution is described by

$$\Psi_i^{-1}(\alpha) = h_i^+(\boldsymbol{x})\Phi_i^{-1}(\alpha), \quad 0 < \alpha < 1.$$

If $h_i(x) < 0$, then $h_i(\boldsymbol{x})\xi_i$ is an uncertain variable whose uncertainty distribution is described by

$$\Psi_i^{-1}(\alpha) = -h_i^-(\boldsymbol{x})\Phi_i^{-1}(1-\alpha), \quad 0 < \alpha < 1.$$

It follows from the operational law that the uncertainty distribution of the sum $h_1(\boldsymbol{x})\xi_1 + h_2(\boldsymbol{x})\xi_2 + \cdots + h_n(\boldsymbol{x})\xi_n$ is described by

$$\Psi^{-1}(\alpha) = \Psi_1^{-1}(\alpha) + \Psi_2^{-1}(\alpha) + \cdots + \Psi_n^{-1}(\alpha), \quad 0 < \alpha < 1.$$

From which we may derive the result immediately.

Theorem 2.4. *Assume that* $x_1, x_2, \cdots, x_n$ *are nonnegative decision variables, and* $\xi_1, \xi_2, \cdots, \xi_n, \xi$ *are independently linear uncertain variables* $\mathcal{L}(a_1, b_1)$, $\mathcal{L}(a_2, b_2), \cdots, \mathcal{L}(a_n, b_n), \mathcal{L}(a, b)$, *respectively. Then for any confidence level* $\alpha \in (0, 1)$, *the chance constraint*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \le \xi\right\} \ge \alpha \tag{2.22}$$

holds if and only if

$$\sum_{i=1}^{n}((1-\alpha)a_i + \alpha b_i)x_i \le \alpha a + (1-\alpha)b. \tag{2.23}$$

Proof: Assume that the uncertain variables $\xi_1, \xi_2, \cdots, \xi_n, \xi$ have uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n, \Phi$, respectively. Then

$$\Phi_i^{-1}(\alpha) = (1-\alpha)a_i + \alpha b_i, \quad i = 1, 2, \cdots, n,$$

$$\Phi^{-1}(1-\alpha) = \alpha a + (1-\alpha)b.$$

Thus the result follows from Theorem 2.3 immediately.

Theorem 2.5. *Assume that $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, and $\xi_1, \xi_2, \cdots, \xi_n, \xi$ are independently zigzag uncertain variables $\mathcal{Z}(a_1, b_1, c_1)$, $\mathcal{Z}(a_2, b_2, c_2), \cdots, \mathcal{Z}(a_n, b_n, c_n), \mathcal{Z}(a, b, c)$, respectively. Then for any confidence level $\alpha \geq 0.5$, the chance constraint*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \leq \xi\right\} \geq \alpha \tag{2.24}$$

holds if and only if

$$\sum_{i=1}^{n}((2-2\alpha)b_i + (2\alpha-1)c_i)x_i \leq \alpha(2\alpha-1)a + (2-2\alpha)b. \tag{2.25}$$

Proof: Assume that the uncertain variables $\xi_1, \xi_2, \cdots, \xi_n, \xi$ have uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n, \Phi$, respectively. Then

$$\Phi_i^{-1}(\alpha) = (2-2\alpha)b_i + (2\alpha-1)c_i, \quad i = 1, 2, \cdots, n,$$

$$\Phi^{-1}(1-\alpha) = (2\alpha-1)a + (2-2\alpha)b.$$

Thus the result follows from Theorem 2.3 immediately.

Theorem 2.6. *Assume that $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, and $\xi_1, \xi_2, \cdots, \xi_n, \xi$ are independently normal uncertain variables $\mathcal{N}(e_1, \sigma_1)$, $\mathcal{N}(e_2, \sigma_2), \cdots, \mathcal{N}(e_n, \sigma_n), \mathcal{N}(e, \sigma)$, respectively. Then for any confidence level $\alpha \in (0, 1)$, the chance constraint*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \leq \xi\right\} \geq \alpha \tag{2.26}$$

holds if and only if

$$\sum_{i=1}^{n}\left(e_i + \frac{\sigma_i\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}\right)x_i \leq e - \frac{\sigma\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}. \tag{2.27}$$

Proof: Assume that the uncertain variables $\xi_1, \xi_2, \cdots, \xi_n, \xi$ have uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n, \Phi$, respectively. Then

$$\Phi_i^{-1}(\alpha) = e_i + \frac{\sigma_i\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}, \quad i = 1, 2, \cdots, n,$$

$$\Phi^{-1}(1-\alpha) = e - \frac{\sigma\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}.$$

Thus the result follows from Theorem 2.3 immediately.

Theorem 2.7. *Assume* $x_1, x_2, \cdots, x_n$ *are nonnegative decision variables, and* $\xi_1, \xi_2, \cdots, \xi_n, \xi$ *are independently lognormal uncertain variables* $\mathcal{LOGN}(e_1, \sigma_1)$, $\mathcal{LOGN}(e_2, \sigma_2), \cdots, \mathcal{LOGN}(e_n, \sigma_n), \mathcal{LOGN}(e, \sigma)$, *respectively. Then for any confidence level* $\alpha \in (0, 1)$, *the chance constraint*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \le \xi\right\} \ge \alpha \tag{2.28}$$

holds if and only if

$$\sum_{i=1}^{n} \exp(e_i) \left(\frac{\alpha}{1-\alpha}\right)^{\sqrt{3}\sigma_i/\pi} x_i \le \exp(e) \left(\frac{1-\alpha}{\alpha}\right)^{\sqrt{3}\sigma/\pi}. \tag{2.29}$$

Proof: Assume that the uncertain variables $\xi_1, \xi_2, \cdots, \xi_n, \xi$ have uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n, \Phi$, respectively. Then

$$\Phi_i^{-1}(\alpha) = \exp(e_i) \left(\frac{\alpha}{1-\alpha}\right)^{\sqrt{3}\sigma_i/\pi}, \quad i = 1, 2, \cdots, n,$$

$$\Phi^{-1}(1-\alpha) = \exp(e) \left(\frac{1-\alpha}{\alpha}\right)^{\sqrt{3}\sigma/\pi}.$$

Thus the result follows from Theorem 2.3 immediately.

2.4 Dependent-Chance Programming

In practice, there usually exist multiple tasks in a complex uncertain decision system. Sometimes, the decision-maker believes the chance criterion and wishes to maximize the chance of meeting these tasks. In order to model this type of uncertain decision system, Liu [122] provided the third type of uncertain programming, called *dependent-chance programming*, in which the underlying philosophy is based on selecting the decision with maximal chance to meet the task. Dependent-chance programming breaks the concept of feasible set and replaces it with uncertain environment.

Definition 2.3. *By an uncertain environment we mean the uncertain constraints represented by*

$$g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, \quad j = 1, 2, \cdots, p \tag{2.30}$$

where $\boldsymbol{x}$ *is a decision vector, and* $\boldsymbol{\xi}$ *is an uncertain vector.*

Definition 2.4. *By a task we mean an uncertain inequality (or a system of uncertain inequalities) represented by*

$$h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \tag{2.31}$$

where $\boldsymbol{x}$ *is a decision vector, and* $\boldsymbol{\xi}$ *is an uncertain vector.*

Definition 2.5. *The chance function of task $\mathcal{E}$ characterized by (2.31) is defined as the uncertain measure that the task $\mathcal{E}$ is met, i.e.,*

$$f(\boldsymbol{x}) = \mathcal{M}\{h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \tag{2.32}$$

subject to the uncertain environment (2.30).

How do we compute the chance function in an uncertain environment? In order to answer this question, we first give some basic definitions. Let $r(x_1, x_2, \cdots, x_n)$ be an n-dimensional function. The ith decision variable x_i is said to be degenerate if

$$r(x_1, \cdots, x_{i-1}, x_i', x_{i+1}, \cdots, x_n) = r(x_1, \cdots, x_{i-1}, x_i'', x_{i+1}, \cdots, x_n)$$

for any x_i' and x_i''; otherwise it is nondegenerate. For example,

$$r(x_1, x_2, x_3, x_4, x_5) = (x_1 + x_3)/x_4$$

is a 5-dimensional function. The variables x_1, x_3, x_4 are nondegenerate, but x_2 and x_5 are degenerate.

Definition 2.6. *Let $\mathcal{E}$ be a task $h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$. The support of the task $\mathcal{E}$, denoted by $\mathcal{E}^*$, is defined as the set consisting of all nondegenerate decision variables of $h(\boldsymbol{x}, \boldsymbol{\xi})$.*

Definition 2.7. *The jth constraint $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$ is called an active constraint of task $\mathcal{E}$ if the set of nondegenerate decision variables of $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ and the support $\mathcal{E}^*$ have nonempty intersection; otherwise it is inactive.*

Definition 2.8. *Let $\mathcal{E}$ be a task $h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$ in the uncertain environment $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, $j = 1, 2, \cdots, p$. The dependent support of task $\mathcal{E}$, denoted by $\mathcal{E}^{**}$, is defined as the set consisting of all nondegenerate decision variables of $h(\boldsymbol{x}, \boldsymbol{\xi})$ and $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ in the active constraints of task $\mathcal{E}$.*

Remark 2.1: It is obvious that $\mathcal{E}^* \subset \mathcal{E}^{**}$ holds.

Definition 2.9. *The jth constraint $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$ is called a dependent constraint of task $\mathcal{E}$ if the set of nondegenerate decision variables of $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ and the dependent support $\mathcal{E}^{**}$ have nonempty intersection; otherwise it is independent.*

Remark 2.2: An active constraint must be a dependent constraint.

Definition 2.10. *Let $\mathcal{E}$ be a task $h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$ in the uncertain environment $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, $j = 1, 2, \cdots, p$. For each decision $\boldsymbol{x}$ and realization $\boldsymbol{\xi}$, the task $\mathcal{E}$ is said to be consistent in the uncertain environment if the following two conditions hold: (i) $h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$; and (ii) $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, $j \in J$, where J is the index set of all dependent constraints.*

In order to maximize the chance of some task in an uncertain environment, a dependent-chance programming may be formulated as follows,

$$\begin{cases} \max \mathcal{M}\{h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \\ \text{subject to:} \\ \qquad g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{2.33}$$

where $\boldsymbol{x}$ is an n-dimensional decision vector, $\boldsymbol{\xi}$ is an uncertain vector, the task $\mathcal{E}$ is characterized by $h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, and the uncertain environment is described by the uncertain constraints $g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, $j = 1, 2, \cdots, p$. The model (2.33) is equivalent to

$$\max \mathcal{M}\{h(\boldsymbol{x}, \boldsymbol{\xi}) \le 0,\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j \in J\} \tag{2.34}$$

where J is the index set of all dependent constraints.

If there are multiple tasks in an uncertain environment, then we have the following dependent-chance multiobjective programming,

$$\begin{cases} \max\, [\mathcal{M}\{h_1(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\}, \cdots, \mathcal{M}\{h_m(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\}] \\ \text{subject to:} \\ \qquad g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{2.35}$$

where tasks $\mathcal{E}_i$ are characterized by $h_i(\boldsymbol{x}, \boldsymbol{\xi}) \le 0$, $i = 1, 2, \cdots, m$, respectively. The model (2.35) is equivalent to

$$\begin{cases} \max \mathcal{M}\{h_1(\boldsymbol{x}, \boldsymbol{\xi}) \le 0,\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j \in J_1\} \\ \max \mathcal{M}\{h_2(\boldsymbol{x}, \boldsymbol{\xi}) \le 0,\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j \in J_2\} \\ \qquad \cdots \\ \max \mathcal{M}\{h_m(\boldsymbol{x}, \boldsymbol{\xi}) \le 0,\ g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j \in J_m\} \end{cases} \tag{2.36}$$

where J_i are the index sets of all dependent constraints of tasks $\mathcal{E}_i$, $i = 1, 2, \cdots, m$, respectively.

Dependent-chance goal programming is employed to formulate uncertain decision systems according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ \vee 0 + v_{ij} d_i^- \vee 0) \\ \text{subject to:} \\ \qquad \mathcal{M}\{h_i(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} - b_i = d_i^+, \quad i = 1, 2, \cdots, m \\ \qquad b_i - \mathcal{M}\{h_i(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} = d_i^-, \quad i = 1, 2, \cdots, m \\ \qquad g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, \qquad\qquad\qquad j = 1, 2, \cdots, p \end{cases}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, $d_i^+ \vee 0$ is the positive deviation from the target of goal i, $d_i^- \vee 0$ is the negative deviation from the target of goal i, g_j is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

Theorem 2.8. *Assume $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, and $\xi_1, \xi_2, \cdots, \xi_n$ are independently linear uncertain variables $\mathcal{L}(a_1, b_1)$, $\mathcal{L}(a_2, b_2)$, $\cdots, \mathcal{L}(a_n, b_n)$, respectively. When*

$$t \in \left[\sum_{i=1}^n a_i x_i, \sum_{i=1}^n b_i x_i\right], \tag{2.37}$$

we have

$$\mathcal{M}\left\{\sum_{i=1}^n \xi_i x_i \le t\right\} = \frac{t - \sum_{i=1}^n a_i x_i}{\sum_{i=1}^n (b_i - a_i) x_i}. \tag{2.38}$$

Otherwise, the measure will be 0 if t is on the left-hand side of interval (2.37) or 1 if t is on the right-hand side.

Proof: Since $\xi_1, \xi_2, \cdots, \xi_n$ are independently linear uncertain variables, their weighted sum $\xi_1 x_1 + \xi_2 x_2 + \cdots + \xi_n x_n$ is also a linear uncertain variable

$$\mathcal{L}\left(\sum_{i=1}^n a_i x_i, \sum_{i=1}^n b_i x_i\right).$$

From this fact we may derive the result immediately.

Theorem 2.9. *Assume that $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, and $\xi_1, \xi_2, \cdots, \xi_n$ are independently zigzag uncertain variables $\mathcal{Z}(a_1, b_1, c_1)$, $\mathcal{Z}(a_2, b_2, c_2), \cdots, \mathcal{Z}(a_n, b_n, c_n)$, respectively. When*

$$t \in \left[\sum_{i=1}^n a_i x_i, \sum_{i=1}^n b_i x_i\right], \tag{2.39}$$

we have

$$\mathcal{M}\left\{\sum_{i=1}^n \xi_i x_i \le t\right\} = \frac{t - \sum_{i=1}^n a_i x_i}{2\sum_{i=1}^n (b_i - a_i) x_i}. \tag{2.40}$$

When

$$t \in \left[\sum_{i=1}^{n} b_i x_i, \sum_{i=1}^{n} c_i x_i\right], \tag{2.41}$$

we have

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \le t\right\} = \frac{t + \sum_{i=1}^{n}(c_i - 2b_i)x_i}{2\sum_{i=1}^{n}(c_i - b_i)x_i}. \tag{2.42}$$

Otherwise, the measure will be 0 if t is on the left-hand side of interval (2.39) or 1 if t is on the right-hand side of interval (2.41).

Proof: Since $\xi_1, \xi_2, \cdots, \xi_n$ are independently zigzag uncertain variables, their weighted sum $\xi_1 x_1 + \xi_2 x_2 + \cdots + \xi_n x_n$ is also a zigzag uncertain variable

$$\mathcal{Z}\left(\sum_{i=1}^{n} a_i x_i, \sum_{i=1}^{n} b_i x_i, \sum_{i=1}^{n} c_i x_i\right).$$

From this fact we may derive the result immediately.

Theorem 2.10. *Assume $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, and $\xi_1, \xi_2, \cdots, \xi_n$ are independently normal uncertain variables $\mathcal{N}(e_1, \sigma_1)$, $\mathcal{N}(e_2, \sigma_2), \cdots, \mathcal{N}(e_n, \sigma_n)$, respectively. Then*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \le t\right\} = \left(1 + \exp\left(\frac{\pi\left(\sum_{i=1}^{n} e_i x_i - t\right)}{\sqrt{3}\sum_{i=1}^{n} \sigma_i x_i}\right)\right)^{-1}. \tag{2.43}$$

Proof: Since $\xi_1, \xi_2, \cdots, \xi_n$ are independently normal uncertain variables, their weighted sum $\xi_1 x_1 + \xi_2 x_2 + \cdots + \xi_n x_n$ is also a normal uncertain variable

$$\mathcal{N}\left(\sum_{i=1}^{n} e_i x_i, \sum_{i=1}^{n} \sigma_i x_i\right).$$

From this fact we may derive the result immediately.

Theorem 2.11. *Assume $x_1, x_2, \cdots, x_n$ are nonnegative decision variables, $\xi_1, \xi_2, \cdots, \xi_n$ are independently lognormal uncertain variables $\mathcal{LOGN}(e_1, \sigma_1)$, $\mathcal{LOGN}(e_2, \sigma_2), \cdots, \mathcal{LOGN}(e_n, \sigma_n)$, respectively. Then*

$$\mathcal{M}\left\{\sum_{i=1}^{n} \xi_i x_i \le t\right\} = \Psi(t) \tag{2.44}$$

where Ψ *is determined by*

$$\Psi^{-1}(\alpha) = \sum_{i=1}^{n} \exp(e_i) \left(\frac{\alpha}{1-\alpha}\right)^{\sqrt{3}\sigma_i/\pi} x_i. \tag{2.45}$$

Proof: Since $\xi_1, \xi_2, \cdots, \xi_n$ are independently lognormal uncertain variables, the uncertainty distribution Ψ of $\xi_1 x_1 + \xi_2 x_2 + \cdots + \xi_n x_n$ is just determined by (2.45). From this fact we may derive the result immediately.

2.5 Uncertain Dynamic Programming

In order to model uncertain decision processes, Liu [122] proposed a general framework of uncertain dynamic programming, including expected value dynamic programming, chance-constrained dynamic programming as well as dependent-chance dynamic programming.

Expected Value Dynamic Programming

Consider an N-stage decision system in which $(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$ represents the state vector, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ the decision vector, $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_N)$ the uncertain vector. We also assume that the state transition function is

$$\boldsymbol{a}_{n+1} = T(\boldsymbol{a}_n, \boldsymbol{x}_n, \boldsymbol{\xi}_n), \quad n = 1, 2, \cdots, N-1. \tag{2.46}$$

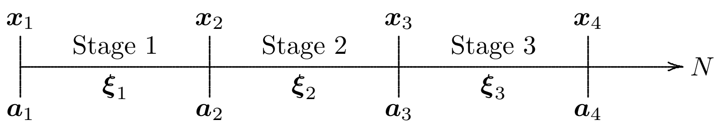

Figure 2.1: A Multistage Decision System

In order to maximize the expected return over the horizon, we may use the following expected value dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \max\limits_{E[g_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N)] \le 0} E[r_N(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{\xi}_N)] \\ f_n(\boldsymbol{a}) = \max\limits_{E[g_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)] \le 0} E[r_n(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{\xi}_n) + f_{n+1}(T(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{\xi}_n))] \\ n \le N-1 \end{cases} \tag{2.47}$$

where r_n are the return functions at the nth stages, $n = 1, 2, \cdots, N$, respectively.

Chance-Constrained Dynamic Programming

In order to maximize the optimistic return over the horizon, we may use the following chance-constrained dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \displaystyle\max_{\mathcal{M}\{g_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N)\le 0\}\ge\alpha} \overline{r}_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \\ f_n(\boldsymbol{a}) = \displaystyle\max_{\mathcal{M}\{g_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)\le 0\}\ge\alpha} \overline{r}_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) + f_{n+1}(T(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)) \\ n \le N-1 \end{cases} \tag{2.48}$$

where the functions $\overline{r}_n$ are defined by

$$\overline{r}_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) = \sup\left\{\overline{r} \;\middle|\; \mathcal{M}\{r_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \ge \overline{r}\} \ge \beta\right\} \tag{2.49}$$

for $n = 1, 2, \cdots, N$. If we want to maximize the pessimistic return over the horizon, then we must define the functions $\overline{r}_n$ as

$$\overline{r}_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) = \inf\left\{\overline{r} \;\middle|\; \mathcal{M}\{r_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le \overline{r}\} \ge \beta\right\} \tag{2.50}$$

for $n = 1, 2, \cdots, N$.

Dependent-Chance Dynamic Programming

In order to maximize the chance over the horizon, we may employ the following dependent-chance dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \displaystyle\max_{g_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N)\le 0} \mathcal{M}\{h_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \le 0\} \\ f_n(\boldsymbol{a}) = \displaystyle\max_{g_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)\le 0} \mathcal{M}\{h_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0\} + f_{n+1}(T(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)) \\ n \le N-1 \end{cases}$$

where $h_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0$ are the events, and $g_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0$ are the uncertain environments at the nth stages, $n = 1, 2, \cdots, N$, respectively.

2.6 Uncertain Multilevel Programming

In order to model uncertain decentralized decision systems, Liu [122] presented three types of uncertain multilevel programming, including expected value multilevel programming, chance-constrained multilevel programming and dependent-chance multilevel programming, and provided the concept of Stackelberg-Nash equilibrium to uncertain multilevel programming.

Expected Value Multilevel Programming

Assume that in a decentralized two-level decision system there is one leader and m followers. Let $\boldsymbol{x}$ and $\boldsymbol{y}_i$ be the control vectors of the leader and the ith followers, $i = 1, 2, \cdots, m$, respectively. We also assume that the objective functions of the leader and ith followers are $F(\boldsymbol{x},\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})$ and $f_i(\boldsymbol{x},\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})$, $i = 1, 2, \cdots, m$, respectively, where $\boldsymbol{\xi}$ is an uncertain vector.

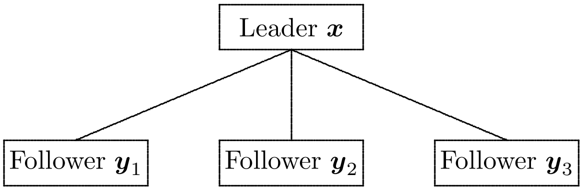

Figure 2.2: A Decentralized Decision System

Let the feasible set of control vector $\boldsymbol{x}$ of the leader be defined by the expected constraint

$$E[G(\boldsymbol{x}, \boldsymbol{\xi})] \le 0 \tag{2.51}$$

where G is a vector-valued function and 0 is a zero vector. Then for each decision $\boldsymbol{x}$ chosen by the leader, the feasibility of control vectors $\boldsymbol{y}_i$ of the ith followers should be dependent on not only $\boldsymbol{x}$ but also $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i+1}, \cdots, \boldsymbol{y}_m$, and generally represented by the expected constraints,

$$E[g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \le 0 \tag{2.52}$$

where g_i are vector-valued functions, $i = 1, 2, \cdots, m$, respectively.

Assume that the leader first chooses his control vector $\boldsymbol{x}$, and the followers determine their control array $(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)$ after that. In order to maximize the expected objective of the leader, we have the following expected value bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} E[F(\boldsymbol{x}, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad E[G(\boldsymbol{x}, \boldsymbol{\xi})] \le 0 \\ \qquad (\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*) \text{ solves problems } (i = 1, 2, \cdots, m) \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y}_i} E[f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad E[g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \le 0. \end{cases} \end{cases} \tag{2.53}$$

Definition 2.11. *Let $\boldsymbol{x}$ be a feasible control vector of the leader. A Nash equilibrium of followers is the feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ with respect to $\boldsymbol{x}$ if*

$$\begin{aligned} & E[f_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \\ & \quad \le E[f_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i^*, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \end{aligned} \tag{2.54}$$

for any feasible array $(\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \cdots, m$.

Definition 2.12. *Suppose that $\boldsymbol{x}^*$ is a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ is a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$.*

We call the array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ a Stackelberg-Nash equilibrium to the expected value bilevel programming (2.53) if

$$E[F(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m, \boldsymbol{\xi})] \le E[F(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \tag{2.55}$$

for any feasible control vector $\overline{\boldsymbol{x}}$ and the Nash equilibrium $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m)$ with respect to $\overline{\boldsymbol{x}}$.

Chance-Constrained Multilevel Programming

In order to maximize the optimistic return subject to the chance constraint, we may use the following chance-constrained bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \max\limits_{\overline{F}} \overline{F} \\ \text{subject to:} \\ \qquad \mathcal{M}\{F(\boldsymbol{x}, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi}) \ge \overline{F}\} \ge \beta \\ \qquad \mathcal{M}\{G(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \alpha \\ \qquad (\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*) \text{ solves problems } (i = 1, 2, \cdots, m) \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y_i}} \max\limits_{\overline{f}_i} \overline{f}_i \\ \text{subject to:} \\ \qquad \mathcal{M}\{f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \ge \overline{f}_i\} \ge \beta_i \\ \qquad \mathcal{M}\{g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0\} \ge \alpha_i \end{cases} \end{cases} \tag{2.56}$$

where $\alpha, \beta, \alpha_i, \beta_i$, $i = 1, 2, \cdots, m$ are predetermined confidence levels.

Definition 2.13. *Let $\boldsymbol{x}$ be a feasible control vector of the leader. A Nash equilibrium of followers is the feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ with respect to $\boldsymbol{x}$ if*

$$\begin{aligned} &\overline{f}_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*) \\ &\quad \le \overline{f}_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i^*, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*) \end{aligned} \tag{2.57}$$

for any feasible array $(\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \cdots, m$.

Definition 2.14. *Suppose that $\boldsymbol{x}^*$ is a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ is a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. The array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ is called a Stackelberg-Nash equilibrium to the chance-constrained bilevel programming (2.56) if*

$$\overline{F}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m) \le \overline{F}(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*) \tag{2.58}$$

for any feasible control vector $\overline{\boldsymbol{x}}$ and the Nash equilibrium $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m)$ with respect to $\overline{\boldsymbol{x}}$.

In order to maximize the pessimistic return, we have the following minimax chance-constrained bilevel programming,

$$\left\{\begin{array}{l}\max\limits_{\boldsymbol{x}}\min\limits_{\overline{F}}\overline{F}\\ \text{subject to:}\\ \qquad \mathcal{M}\{F(\boldsymbol{x},\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*,\boldsymbol{\xi})\le\overline{F}\}\ge\beta\\ \qquad \mathcal{M}\{G(\boldsymbol{x},\boldsymbol{\xi})\le 0\}\ge\alpha\\ \qquad (\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*)\text{ solves problems } (i=1,2,\cdots,m)\\ \qquad \left\{\begin{array}{l}\max\limits_{\boldsymbol{y_i}}\min\limits_{\overline{f}_i}\overline{f}_i\\ \text{subject to:}\\ \qquad \mathcal{M}\{f_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le\overline{f}_i\}\ge\beta_i\\ \qquad \mathcal{M}\{g_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0\}\ge\alpha_i\end{array}\right.\end{array}\right. \tag{2.59}$$

where $\alpha,\beta,\alpha_i,\beta_i$, $i=1,2,\cdots,m$ are predetermined confidence levels.

Dependent-Chance Multilevel Programming

Let $H(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0$ and $h_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0$ be the tasks of the leader and ith followers, $i=1,2,\cdots,m$, respectively. In order to maximize the chance functions of the leader and followers, we have the following dependent-chance bilevel programming,

$$\left\{\begin{array}{l}\max\limits_{\boldsymbol{x}}\mathcal{M}\{H(\boldsymbol{x},\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*,\boldsymbol{\xi})\le 0\}\\ \text{subject to:}\\ \qquad G(\boldsymbol{x},\boldsymbol{\xi})\le 0\\ \qquad (\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*)\text{ solves problems } (i=1,2,\cdots,m)\\ \qquad \left\{\begin{array}{l}\max\limits_{\boldsymbol{y_i}}\mathcal{M}\{h_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0\}\\ \text{subject to:}\\ \qquad g_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0.\end{array}\right.\end{array}\right. \tag{2.60}$$

Definition 2.15. *Let $\boldsymbol{x}$ be a control vector of the leader. We call the array $(\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*)$ a Nash equilibrium of followers with respect to $\boldsymbol{x}$ if*

$$\begin{array}{l}\mathcal{M}\{h_i(\boldsymbol{x},\boldsymbol{y}_1^*,\cdots,\boldsymbol{y}_{i-1}^*,\boldsymbol{y}_i,\boldsymbol{y}_{i+1}^*,\cdots,\boldsymbol{y}_m^*,\boldsymbol{\xi})\le 0\}\\ \qquad \le\mathcal{M}\{h_i(\boldsymbol{x},\boldsymbol{y}_1^*,\cdots,\boldsymbol{y}_{i-1}^*,\boldsymbol{y}_i^*,\boldsymbol{y}_{i+1}^*,\cdots,\boldsymbol{y}_m^*,\boldsymbol{\xi})\le 0\}\end{array} \tag{2.61}$$

subject to the uncertain environment $g_i(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})\le 0, i=1,2,\cdots,$ m for any array $(\boldsymbol{y}_1^*,\cdots,\boldsymbol{y}_{i-1}^*,\boldsymbol{y}_i,\boldsymbol{y}_{i+1}^*,\cdots,\boldsymbol{y}_m^*)$ and $i=1,2,\cdots,m$.

Definition 2.16. *Let $\boldsymbol{x}^*$ be a control vector of the leader, and $(\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,$ $\boldsymbol{y}_m^*)$ a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. Then $(\boldsymbol{x}^*,\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,$ $\boldsymbol{y}_m^*)$ is called a Stackelberg-Nash equilibrium to the dependent-chance bilevel programming (2.60) if*

$$\mathcal{M}\{H(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}}_1,\overline{\boldsymbol{y}}_2,\cdots,\overline{\boldsymbol{y}}_m,\boldsymbol{\xi})\le 0\}\le\mathcal{M}\{H(\boldsymbol{x}^*,\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\cdots,\boldsymbol{y}_m^*,\boldsymbol{\xi})\le 0\}$$

subject to the uncertain environment $G(\boldsymbol{x},\boldsymbol{\xi})\le 0$ *for any control vector* $\overline{\boldsymbol{x}}$ *and the Nash equilibrium* $(\overline{\boldsymbol{y}}_1,\overline{\boldsymbol{y}}_2,\cdots,\overline{\boldsymbol{y}}_m)$ *with respect to* $\overline{\boldsymbol{x}}$.

2.7 Hybrid Intelligent Algorithm

From the mathematical viewpoint, there is no difference between deterministic mathematical programming and uncertain programming except for the fact that there exist uncertain functions in the latter. Essentially, there are three types of uncertain functions in uncertain programming,

$$\begin{aligned} &U_1:\boldsymbol{x}\to E[f(\boldsymbol{x},\boldsymbol{\xi})],\\ &U_2:\boldsymbol{x}\to\mathcal{M}\left\{f(\boldsymbol{x},\boldsymbol{\xi})\le 0\right\},\\ &U_3:\boldsymbol{x}\to\max\left\{\overline{f}\;\middle|\;\mathcal{M}\left\{f(\boldsymbol{x},\boldsymbol{\xi})\ge\overline{f}\right\}\ge\alpha\right\}. \end{aligned}\tag{2.62}$$

Note that those uncertain functions may be calculated by the 99-method if the function f is monotone. Otherwise, I give up! It is fortunate for us that almost all functions in practical problems are indeed monotone.

In order to solve uncertain programming models, we must find a numerical method for solving deterministic mathematical programming, for example, genetic algorithm, particle swarm optimization, neural networks, tabu search, or any classical algorithms.

Then, for example, we may integrate the 99-method and the genetic algorithm to produce a hybrid intelligent algorithm for solving uncertain programming models:

Step 1. Initialize chromosomes whose feasibility may be checked by the 99-method.

Step 2. Update the chromosomes by the crossover operation in which the 99-method may be employed to check the feasibility of offsprings.

Step 3. Update the chromosomes by the mutation operation in which the 99-method may be employed to check the feasibility of offsprings.

Step 4. Calculate the objective values for all chromosomes by the 99-method.

Step 5. Compute the fitness of each chromosome based on the objective values.

Step 6. Select the chromosomes by spinning the roulette wheel.

Step 7. Repeat the second to sixth steps a given number of cycles.

Step 8. Report the best chromosome as the optimal solution.

Please visit the website at http://orsc.edu.cn/liu/resources.htm for computer source files of hybrid intelligent algorithm and numerical examples.

2.8 Ψ Graph

Any types of uncertain programming (including stochastic programming, fuzzy programming and hybrid programming) may be represented by a Ψ graph

(**P**hilosophy, **S**tructure, **I**nformation)

which is essentially a coordinate system in which, for example, the plane

"**P** = CCP"

represents the class of chance-constrained programming; the plane

"**S** = MOP"

represents the class of multiobjective programming; the plane

"**I** = Uncertain"

represents the class of uncertain programming; and the point

"(**P**,**S**,**I**) = (DCP, GP, Uncertain)"

represents the uncertain dependent-chance goal programming.

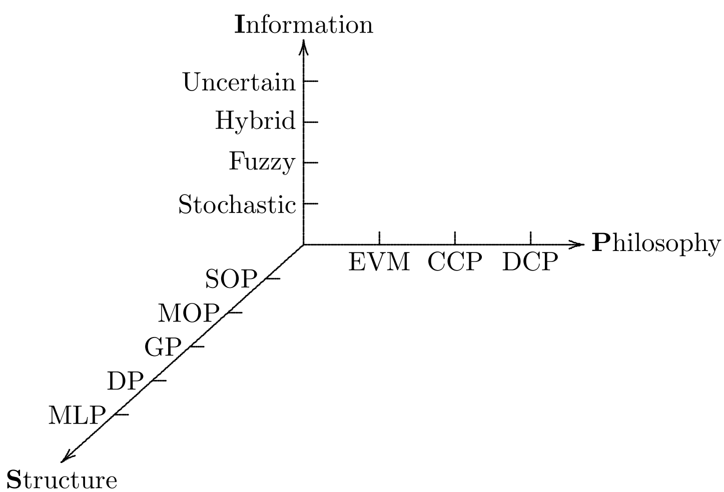

Figure 2.3: Ψ Graph for Uncertain Programming Classifications (Liu [112])

2.9 Project Scheduling Problem

Project scheduling problem is to determine the schedule of allocating resources so as to balance the total cost and the completion time. The study of project scheduling problem with uncertain factors was started by Liu [122] in 2009. This section presents an uncertain programming model for project scheduling problem in which the duration times are assumed to be uncertain variables with known uncertainty distributions.

Project scheduling is usually represented by a directed acyclic graph where nodes correspond to milestones, and arcs to activities which are basically characterized by the times and costs consumed.

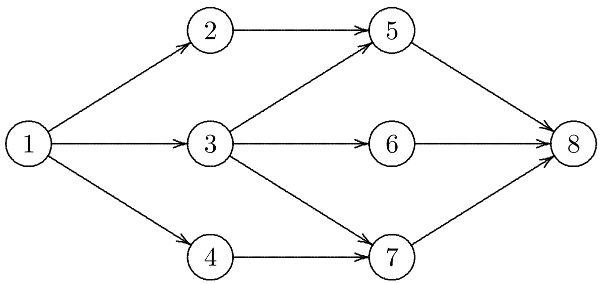

Figure 2.4: A Project with 8 Milestones and 11 Activities

Let $(\mathcal{V}, \mathcal{A})$ be a directed acyclic graph, where $\mathcal{V} = \{1, 2, \cdots, n, n+1\}$ is the set of nodes, $\mathcal{A}$ is the set of arcs, $(i, j) \in \mathcal{A}$ is the arc of the graph $(\mathcal{V}, \mathcal{A})$ from nodes i to j. It is well-known that we can rearrange the indexes of the nodes in $\mathcal{V}$ such that $i < j$ for all $(i, j) \in \mathcal{A}$.

Before we begin to study project scheduling problem with uncertain activity duration times, we first make some assumptions: (a) all of the costs needed are obtained via loans with some given interest rate; and (b) each activity can be processed only if the loan needed is allocated and all the foregoing activities are finished.

In order to model the project scheduling problem, we introduce the following indices and parameters:

ξ_{ij}: uncertain duration time of activity (i, j) in $\mathcal{A}$;

Φ_{ij}: uncertainty distribution of ξ_{ij};

c_{ij}: cost of activity (i, j) in $\mathcal{A}$;

r: interest rate;

x_i: integer decision variable representing the allocating time of all loans needed for all activities (i, j) in $\mathcal{A}$.

Starting Times

For simplicity, we write $\boldsymbol{\xi} = \{\xi_{ij} : (i, j) \in \mathcal{A}\}$ and $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$. Assume each uncertain duration time ξ_{ij} is represented by a 99-table,

0.01	0.02	0.03	$\cdots$	0.99
t_{ij}^1	t_{ij}^2	t_{ij}^3	$\cdots$	t_{ij}^{99}

(2.63)

Let $T_i(\boldsymbol{x},\boldsymbol{\xi})$ denote the starting time of all activities (i,j) in $\mathcal{A}$. According to the assumptions, the starting time of the total project (i.e., the starting time of of all activities $(1,j)$ in $\mathcal{A}$) should be

$$T_1(\boldsymbol{x},\boldsymbol{\xi}) = x_1 \tag{2.64}$$

whose inverse uncertainty distribution may be written as

$$\Psi_1^{-1}(\alpha) = x_1 \tag{2.65}$$

and has a 99-table,

0.01	0.02	0.03	$\cdots$	0.99
x_1	x_1	x_1	$\cdots$	x_1

(2.66)

Generally, suppose that the starting time $T_k(\boldsymbol{x},\boldsymbol{\xi})$ of all activities (k,i) in $\mathcal{A}$ has an inverse uncertainty distribution $\Psi_k^{-1}(\alpha)$ and has a 99-table,

0.01	0.02	0.03	$\cdots$	0.99
y_k^1	y_k^2	y_k^3	$\cdots$	y_k^{99}

(2.67)

Then the starting time $T_i(\boldsymbol{x},\boldsymbol{\xi})$ of all activities (i,j) in $\mathcal{A}$ should be

$$T_i(\boldsymbol{x},\boldsymbol{\xi}) = x_i \vee \max_{(k,i)\in\mathcal{A}} (T_k(\boldsymbol{x},\boldsymbol{\xi}) + \xi_{ki}) \tag{2.68}$$

whose inverse uncertainty distribution is

$$\Psi_i^{-1}(\alpha) = x_i \vee \max_{(k,i)\in\mathcal{A}} \left(\Psi_k^{-1}(\alpha) + \Phi_{ki}^{-1}(\alpha)\right) \tag{2.69}$$

and has a 99-table,

0.01	$\cdots$	0.99
$x_i \vee \max_{(k,i)\in\mathcal{A}} (y_k^1 + t_{ki}^1)$	$\cdots$	$x_i \vee \max_{(k,i)\in\mathcal{A}} (y_k^{99} + t_{ki}^{99})$

(2.70)

where $y_k^1, y_k^2, \cdots, y_k^{99}$ are determined by (2.67). This recursive process may produce all starting times of activities.

Completion Time

The completion time $T(\boldsymbol{x},\boldsymbol{\xi})$ of the total project (i.e, the finish time of all activities $(k,n+1)$ in $\mathcal{A}$) is

$$T(\boldsymbol{x},\boldsymbol{\xi}) = \max_{(k,n+1)\in\mathcal{A}} (T_k(\boldsymbol{x},\boldsymbol{\xi}) + \xi_{k,n+1}) \tag{2.71}$$

whose inverse uncertainty distribution is

$$\Psi^{-1}(\alpha) = \max_{(k,n+1)\in\mathcal{A}} \left(\Psi_k^{-1}(\alpha) + \Phi_{k,n+1}^{-1}(\alpha)\right) \tag{2.72}$$

and has a 99-table,

$$\begin{array}{|c|c|c|}\hline 0.01 & \cdots & 0.99 \\ \hline \max\limits_{(k,n+1)\in\mathcal{A}}(y_k^1+t_{k,n+1}^1) & \cdots & \max\limits_{(k,n+1)\in\mathcal{A}}(y_k^{99}+t_{k,n+1}^{99}) \\ \hline \end{array} \tag{2.73}$$

where $y_k^1, y_k^2, \cdots, y_k^{99}$ are determined by (2.67).

Total Cost

Based on the completion time $T(\boldsymbol{x},\boldsymbol{\xi})$, the total cost of the project can be written as

$$C(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{(i,j)\in\mathcal{A}} c_{ij}\,(1+r)^{\lceil T(\boldsymbol{x},\boldsymbol{\xi})-x_i\rceil} \tag{2.74}$$

where $\lceil a\rceil$ represents the minimal integer greater than or equal to a. Note that $C(\boldsymbol{x},\boldsymbol{\xi})$ is a discrete uncertain variable whose inverse uncertainty distribution is

$$\Upsilon^{-1}(\boldsymbol{x};\alpha) = \sum_{(i,j)\in\mathcal{A}} c_{ij}\,(1+r)^{\lceil \Psi^{-1}(\boldsymbol{x};\alpha)-x_i\rceil} \tag{2.75}$$

for $0<\alpha<1$. Since $T(\boldsymbol{x},\boldsymbol{\xi})$ is obtained by the recursive process and represented by a 99-table,

$$\begin{array}{|c|c|c|c|c|}\hline 0.01 & 0.02 & 0.03 & \cdots & 0.99 \\ \hline s_1 & s_2 & s_3 & \cdots & s_{99} \\ \hline \end{array} \tag{2.76}$$

the total cost $C(\boldsymbol{x},\boldsymbol{\xi})$ has a 99-table,

$$\begin{array}{|c|c|c|}\hline 0.01 & \cdots & 0.99 \\ \hline \sum\limits_{(i,j)\in\mathcal{A}} c_{ij}\,(1+r)^{\lceil s_1-x_i\rceil} & \cdots & \sum\limits_{(i,j)\in\mathcal{A}} c_{ij}\,(1+r)^{\lceil s_{99}-x_i\rceil} \\ \hline \end{array} \tag{2.77}$$

Project Scheduling Model

If we want to minimize the expected cost of the project under the completion time constraint, we may construct the following project scheduling model,

$$\begin{cases} \min E[C(\boldsymbol{x},\boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad \mathcal{M}\{T(\boldsymbol{x},\boldsymbol{\xi})\le T^0\}\ge\alpha \\ \qquad \boldsymbol{x}\ge 0, \text{ integer vector} \end{cases} \tag{2.78}$$

where T^0 is a due date of the project, α is a predetermined confidence level, $T(\boldsymbol{x},\boldsymbol{\xi})$ is the completion time defined by (2.71), and $C(\boldsymbol{x},\boldsymbol{\xi})$ is the total cost defined by (2.74). This model is equivalent to

$$\begin{cases} \min \displaystyle\int_0^{+\infty}(1-\Upsilon(\boldsymbol{x};z))\mathrm{d}z \\ \text{subject to:} \\ \qquad \Psi(\boldsymbol{x};T^0)\ge\alpha \\ \qquad \boldsymbol{x}\ge 0, \text{ integer vector} \end{cases} \tag{2.79}$$

where Ψ is determined by (2.72) and Υ is determined by (2.75). Note that the completion time $T(\boldsymbol{x},\boldsymbol{\xi})$ and total cost $C(\boldsymbol{x},\boldsymbol{\xi})$ are obtained by the recursive process and are respectively represented by 99-tables,

0.01	0.02	0.03	$\cdots$	0.99
s_1	s_2	s_3	$\cdots$	s_{99}

0.01	0.02	0.03	$\cdots$	0.99
c_1	c_2	c_3	$\cdots$	c_{99}

(2.80)

Thus the project scheduling model is simplified as follows,

$$\begin{cases} \min\ (c_1 + c_2 + \cdots + c_{99})/99 \\ \text{subject to:} \\ \qquad k/100 \ge \alpha \text{ if } s_k \ge T^0 \\ \qquad \boldsymbol{x} \ge 0, \text{ integer vector.} \end{cases} \tag{2.81}$$

Numerical Experiment

Consider a project scheduling problem shown by Figure 2.4 in which there are 8 milestones and 11 activities. Assume that all duration times of activities are linear uncertain variables,

$$\xi_{ij} \sim \mathcal{L}(3i, 3j), \quad \forall (i,j) \in \mathcal{A}$$

and assume that the costs of activities are

$$c_{ij} = i + j, \quad \forall (i,j) \in \mathcal{A}.$$

In addition, we also suppose that the interest rate is $r = 0.02$, the due date is $T^0 = 60$, and the confidence level is $\alpha = 0.85$. In order to find an optimal project schedule, we integrate the 99-method and a genetic algorithm to produce a hybrid intelligent algorithm. A run of the computer program (http://orsc.edu.cn/liu/resources.htm) shows that the optimal allocating times of all loans needed for all activities are

Date	7	11	13	23	26	29
Node	1	4	3	2, 7	6	5
Loan	12	11	27	22	14	13

whose expected total cost is 166.8, and $\mathcal{M}\{T(\boldsymbol{x}^*,\boldsymbol{\xi}) \le 60\} = 0.89$.

2.10 Vehicle Routing Problem

Vehicle routing problem (VRP) is concerned with finding efficient routes, beginning and ending at a central depot, for a fleet of vehicles to serve a number of customers.

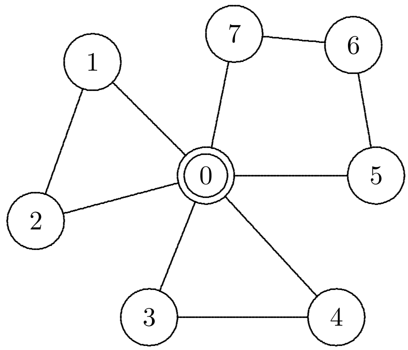

Figure 2.5: A Vehicle Routing Plan with Single Depot and 7 Customers

Due to its wide applicability and economic importance, vehicle routing problem has been extensively studied. Liu [122] first introduced uncertainty theory into the research area of vehicle routing problem in 2009. In this section, vehicle routing problem will be modelled by uncertain programming in which the travel times are assumed to be uncertain variables with known uncertainty distributions.

We assume that (a) a vehicle will be assigned for only one route on which there may be more than one customer; (b) a customer will be visited by one and only one vehicle; (c) each route begins and ends at the depot; and (d) each customer specifies its time window within which the delivery is permitted or preferred to start.

Let us first introduce the following indices and model parameters:

$i = 0$: depot;

$i = 1, 2, \cdots, n$: customers;

$k = 1, 2, \cdots, m$: vehicles;

D_{ij}: travel distance from customers i to j, $i, j = 0, 1, 2, \cdots, n$;

T_{ij}: uncertain travel time from customers i to j, $i, j = 0, 1, 2, \cdots, n$;

Φ_{ij}: uncertainty distribution of T_{ij}, $i, j = 0, 1, 2, \cdots, n$;

$[a_i, b_i]$: time window of customer i, $i = 1, 2, \cdots, n$.

Operational Plan

In this book, the operational plan is represented by the formulation (Liu [112]) via three decision vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{t}$, where

$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$: integer decision vector representing n customers with $1 \le x_i \le n$ and $x_i \ne x_j$ for all $i \ne j$, $i, j = 1, 2, \cdots, n$. That is, the sequence $\{x_1, x_2, \cdots, x_n\}$ is a rearrangement of $\{1, 2, \cdots, n\}$;

$\boldsymbol{y} = (y_1, y_2, \cdots, y_{m-1})$: integer decision vector with $y_0 \equiv 0 \le y_1 \le y_2 \le \cdots \le y_{m-1} \le n \equiv y_m$;

$\boldsymbol{t} = (t_1, t_2, \cdots, t_m)$: each t_k represents the starting time of vehicle k at the depot, $k = 1, 2, \cdots, m$.

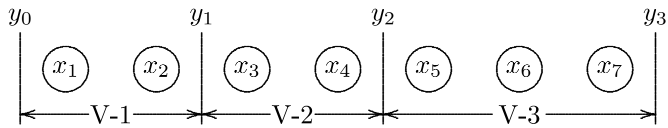

Figure 2.6: Formulation of Operational Plan in which Vehicle 1 Visits Customers x_1, x_2, Vehicle 2 Visits Customers x_3, x_4 and Vehicle 3 Visits Customers x_5, x_6, x_7.

We note that the operational plan is fully determined by the decision vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{t}$ in the following way. For each k $(1 \le k \le m)$, if $y_k = y_{k-1}$, then vehicle k is not used; if $y_k > y_{k-1}$, then vehicle k is used and starts from the depot at time t_k, and the tour of vehicle k is $0 \to x_{y_{k-1}+1} \to x_{y_{k-1}+2} \to \cdots \to x_{y_k} \to 0$. Thus the tours of all vehicles are as follows:

$$\begin{aligned}
&\text{Vehicle 1: } 0 \to x_{y_0+1} \to x_{y_0+2} \to \cdots \to x_{y_1} \to 0;\\
&\text{Vehicle 2: } 0 \to x_{y_1+1} \to x_{y_1+2} \to \cdots \to x_{y_2} \to 0;\\
&\quad\cdots\\
&\text{Vehicle } m\text{: } 0 \to x_{y_{m-1}+1} \to x_{y_{m-1}+2} \to \cdots \to x_{y_m} \to 0.
\end{aligned}$$

It is clear that this type of representation is intuitive, and the total number of decision variables is $n + 2m - 1$. We also note that the above decision variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{t}$ ensure that: (a) each vehicle will be used at most one time; (b) all tours begin and end at the depot; (c) each customer will be visited by one and only one vehicle; and (d) there is no subtour.

Arrival Times

Let $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ be the arrival time function of some vehicles at customers i for $i = 1, 2, \cdots, n$. We remind readers that $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ are determined by the decision variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{t}$, $i = 1, 2, \cdots, n$. Since unloading can start either immediately, or later, when a vehicle arrives at a customer, the calculation of $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ is heavily dependent on the operational strategy. Here we assume that the customer does not permit a delivery earlier than the time window. That is, the vehicle will wait to unload until the beginning of the time window if it arrives before the time window. If a vehicle arrives at a customer after the beginning of the time window, unloading will start immediately. For each k with $1 \le k \le m$, if vehicle k is used (i.e., $y_k > y_{k-1}$), then we have

$$f_{x_{y_{k-1}+1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}) = t_k + T_{0x_{y_{k-1}+1}} \tag{2.82}$$

and

$$f_{x_{y_{k-1}+j}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}) = f_{x_{y_{k-1}+j-1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}) \vee a_{x_{y_{k-1}+j-1}} + T_{x_{y_{k-1}+j-1}x_{y_{k-1}+j}} \tag{2.83}$$

for $2 \le j \le y_k - y_{k-1}$. It follows from the uncertainty of travel times T_{ij}'s that the arrival times $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$, $i = 1, 2, \cdots, n$ are uncertain variables fully determined by (2.82) and (2.83).

Assume that each travel time T_{ij} from customers i to j is represented by a 99-table,

0.01	0.02	0.03	$\cdots$	0.99
t_{ij}^1	t_{ij}^2	t_{ij}^3	$\cdots$	t_{ij}^{99}

(2.84)

If the vehicle k is used, i.e., $y_k > y_{k-1}$, then the arrival time $f_{x_{y_{k-1}+1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ at the customer $x_{y_{k-1}+1}$ is an uncertain variable whose inverse uncertainty distribution is

$$\Psi^{-1}_{x_{y_{k-1}+1}}(\alpha) = t_k + \Phi^{-1}_{0x_{y_{k-1}+1}}(\alpha) \tag{2.85}$$

and has a 99-table,

0.01	0.02	$\cdots$	0.99
$t_k + t^1_{0x_{y_{k-1}+1}}$	$t_k + t^2_{0x_{y_{k-1}+1}}$	$\cdots$	$t_k + t^{99}_{0x_{y_{k-1}+1}}$

(2.86)

Generally, suppose that the arrival time $f_{x_{y_{k-1}+j-1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ has an inverse uncertainty distribution $\Psi^{-1}_{x_{y_{k-1}+j-1}}(\alpha)$, and has a 99-table,

0.01	0.02	$\cdots$	0.99
$s^1_{x_{y_{k-1}+j-1}}$	$s^2_{x_{y_{k-1}+j-1}}$	$\cdots$	$s^{99}_{x_{y_{k-1}+j-1}}$

(2.87)

Since the arrival time $f_{x_{y_{k-1}+j}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ at the customer $x_{y_{k-1}+j}$ has an inverse uncertainty distribution

$$\Psi^{-1}_{x_{y_{k-1}+j}}(\alpha) = \Psi^{-1}_{x_{y_{k-1}+j-1}}(\alpha) \vee a_{x_{y_{k-1}+j-1}} + \Phi^{-1}_{x_{y_{k-1}+j-1}x_{y_{k-1}+j}}(\alpha) \tag{2.88}$$

for $2 \le j \le y_k - y_{k-1}$, the arrival time $f_{x_{y_{k-1}+j}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ has a 99-table,

0.01	$\cdots$	0.99
$s^1_{x_{y_{k-1}+j-1}} \vee a_{x_{y_{k-1}+j-1}}$ $+t^1_{x_{y_{k-1}+j-1}x_{y_{k-1}+j}}$	$\cdots$	$s^{99}_{x_{y_{k-1}+j-1}} \vee a_{x_{y_{k-1}+j-1}}$ $+t^{99}_{x_{y_{k-1}+j-1}x_{y_{k-1}+j}}$

(2.89)

where $s^1_{x_{y_{k-1}+j-1}}, s^2_{x_{y_{k-1}+j-1}}, \cdots, s^{99}_{x_{y_{k-1}+j-1}}$ are determined by (2.87). This recursive process may produce all arrival times at customers.

Travel Distance

Let $g(\boldsymbol{x}, \boldsymbol{y})$ be the total travel distance of all vehicles. Then we have

$$g(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{m} g_k(\boldsymbol{x}, \boldsymbol{y}) \tag{2.90}$$

where

$$g_k(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} D_{0x_{y_{k-1}+1}} + \sum\limits_{j=y_{k-1}+1}^{y_k-1} D_{x_j x_{j+1}} + D_{x_{y_k}0}, & \text{if } y_k > y_{k-1} \\ 0, & \text{if } y_k = y_{k-1} \end{cases}$$

for $k = 1, 2, \cdots, m$.

Vehicle Routing Model

If we hope that each customer i $(1 \le i \le n)$ is visited within its time window $[a_i, b_i]$ with confidence level α_i (i.e., the vehicle arrives at customer i before time b_i), then we have the following chance constraint,

$$\mathcal{M}\{f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}) \le b_i\} \ge \alpha_i. \tag{2.91}$$

If we want to minimize the total travel distance of all vehicles subject to the time window constraint, then we have the following vehicle routing model,

$$\begin{cases} \min g(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to:} \\ \qquad \mathcal{M}\{f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}) \le b_i\} \ge \alpha_i, \quad i = 1, 2, \cdots, n \\ \qquad 1 \le x_i \le n, \quad i = 1, 2, \cdots, n \\ \qquad x_i \ne x_j, \quad i \ne j, \ i, j = 1, 2, \cdots, n \\ \qquad 0 \le y_1 \le y_2 \le \cdots \le y_{m-1} \le n \\ \qquad x_i, y_j, \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m-1, \quad \text{integers} \end{cases} \tag{2.92}$$

which is equivalent to

$$\begin{cases} \min g(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to:} \\ \qquad \Psi_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}; b_i) \ge \alpha_i, \quad i = 1, 2, \cdots, n \\ \qquad 1 \le x_i \le n, \quad i = 1, 2, \cdots, n \\ \qquad x_i \ne x_j, \quad i \ne j, \ i, j = 1, 2, \cdots, n \\ \qquad 0 \le y_1 \le y_2 \le \cdots \le y_{m-1} \le n \\ \qquad x_i, y_j, \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m-1, \quad \text{integers} \end{cases} \tag{2.93}$$

where Ψ_i are uncertainty distributions determined by (2.85) and (2.88) for $i = 1, 2, \cdots, n$. Note that all arrival times $f_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$, $i = 1, 2, \cdots, n$ are obtained by the 99-method and are respectively represented by 99-tables,

0.01	0.02	0.03	$\cdots$	0.99
s_1^1	s_1^2	s_1^3	$\cdots$	s_1^{99}

0.01	0.02	0.03	$\cdots$	0.99
s_2^1	s_2^2	s_2^3	$\cdots$	s_2^{99}

$\vdots$ (2.94)

0.01	0.02	0.03	$\cdots$	0.99
s_n^1	s_n^2	s_n^3	$\cdots$	s_n^{99}

Thus the vehicle routing model is simplified as follows,

$$\begin{cases} \min g(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to:} \\ \qquad k/100 \geq \alpha_i \text{ if } s_i^k \geq b_i, \quad i = 1, 2, \cdots, n \\ \qquad 1 \leq x_i \leq n, \quad i = 1, 2, \cdots, n \\ \qquad x_i \neq x_j, \quad i \neq j, \ i, j = 1, 2, \cdots, n \\ \qquad 0 \leq y_1 \leq y_2 \leq \cdots \leq y_{m-1} \leq n \\ \qquad x_i, y_j, \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m-1, \quad \text{integers.} \end{cases} \tag{2.95}$$

Numerical Experiment

Assume that there are 3 vehicles and 7 customers with the following time windows,

Node	Window	Node	Window
1	$[7:00, 9:00]$	5	$[15:00, 17:00]$
2	$[7:00, 9:00]$	6	$[19:00, 21:00]$
3	$[15:00, 17:00]$	7	$[19:00, 21:00]$
4	$[15:00, 17:00]$		

and each customer is visited within time windows with confidence level 0.90. We also assume that the distances are

$$D_{ij} = |i - j|, \quad i, j = 0, 1, 2, \cdots, 7$$

and travel times are normal uncertain variables

$$T_{ij} \sim \mathcal{N}(2|i - j|, 1), \quad i, j = 0, 1, 2, \cdots, 7.$$

In order to find an optimal operational plan, we integrate the 99-method and a genetic algorithm to produce a hybrid intelligent algorithm. A run of the computer program (http://orsc.edu.cn/liu/resources.htm) shows that the optimal operational plan is

Vehicle 1: depot$\rightarrow$ 1 $\rightarrow$ 3 $\rightarrow$depot, starting time: 6:18
Vehicle 2: deport$\rightarrow$ 2 $\rightarrow$ 5 $\rightarrow$ 7 $\rightarrow$depot, starting time: 4:18
Vehicle 3: depot$\rightarrow$ 4 $\rightarrow$ 6 $\rightarrow$depot, starting time: 8:18

whose total travel distance is 32.

2.11 Machine Scheduling Problem

Machine scheduling problem is concerned with finding an efficient schedule during an uninterrupted period of time for a set of machines to process a set of jobs. A lot of research work has been done on this type of problem. The study of machine scheduling problem with uncertain processing times was started by Liu [122] in 2009.

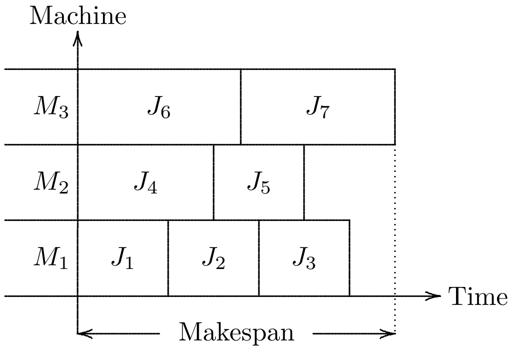

Figure 2.7: A Machine Schedule with 3 Machines and 7 Jobs

In a machine scheduling problem, we assume that (a) each job can be processed on any machine without interruption; (b) each machine can process only one job at a time; and (c) the processing times are uncertain variables with known uncertainty distributions. We also use the following indices and parameters:

$i = 1, 2, \cdots, n$: jobs;
$k = 1, 2, \cdots, m$: machines;
ξ_{ik}: uncertain processing time of job i on machine k;
Φ_{ik}: uncertainty distribution of ξ_{ik}.

How to Represent a Schedule?

The schedule is represented by the formulation (Liu [112]) via two decision vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, where

$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$: integer decision vector representing n jobs with $1 \le x_i \le n$ and $x_i \ne x_j$ for all $i \ne j$, $i, j = 1, 2, \cdots, n$. That is, the sequence $\{x_1, x_2, \cdots, x_n\}$ is a rearrangement of $\{1, 2, \cdots, n\}$;

$\boldsymbol{y} = (y_1, y_2, \cdots, y_{m-1})$: integer decision vector with $y_0 \equiv 0 \le y_1 \le y_2 \le \cdots \le y_{m-1} \le n \equiv y_m$.

We note that the schedule is fully determined by the decision vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in the following way. For each k $(1 \le k \le m)$, if $y_k = y_{k-1}$, then the machine k is not used; if $y_k > y_{k-1}$, then the machine k is used and processes jobs $x_{y_{k-1}+1}, x_{y_{k-1}+2}, \cdots, x_{y_k}$ in turn. Thus the schedule of all machines is as follows,

$$\begin{aligned}
&\text{Machine 1: } x_{y_0+1} \to x_{y_0+2} \to \cdots \to x_{y_1};\\
&\text{Machine 2: } x_{y_1+1} \to x_{y_1+2} \to \cdots \to x_{y_2};\\
&\quad \cdots\\
&\text{Machine } m\text{: } x_{y_{m-1}+1} \to x_{y_{m-1}+2} \to \cdots \to x_{y_m}.
\end{aligned} \tag{2.96}$$

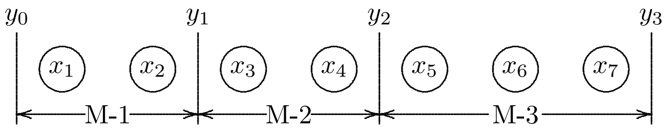

Figure 2.8: Formulation of Schedule in which Machine 1 Processes Jobs x_1, x_2, Machine 2 Processes Jobs x_3, x_4 and Machine 3 Processes Jobs x_5, x_6, x_7.

Completion Times

Let $C_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ be the completion times of jobs i, $i = 1, 2, \cdots, n$, respectively. For each k with $1 \le k \le m$, if the machine k is used (i.e., $y_k > y_{k-1}$), then we have

$$C_{x_{y_{k-1}+1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) = \xi_{x_{y_{k-1}+1}k} \tag{2.97}$$

and

$$C_{x_{y_{k-1}+j}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) = C_{x_{y_{k-1}+j-1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) + \xi_{x_{y_{k-1}+j}k} \tag{2.98}$$

for $2 \le j \le y_k - y_{k-1}$.

Assume that each uncertain processing time ξ_{ik} of job i on machine k is represented by a 99-table,

0.01	0.02	0.03	$\cdots$	0.99
t_{ik}^1	t_{ik}^2	t_{ik}^3	$\cdots$	t_{ik}^{99}

(2.99)

If the machine k is used, then the completion time $C_{x_{y_{k-1}+1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ of job $x_{y_{k-1}+1}$ is an uncertain variable whose inverse uncertainty distribution is

$$\Psi^{-1}_{x_{y_{k-1}+1}}(\alpha) = \Phi^{-1}_{x_{y_{k-1}+1}k}(\alpha) \tag{2.100}$$

and has a 99-table,

0.01	0.02	$\cdots$	0.99
$t^1_{x_{y_{k-1}+1}}$	$t^2_{x_{y_{k-1}+1}}$	$\cdots$	$t^{99}_{x_{y_{k-1}+1}}$

(2.101)

Generally, suppose the completion time $C_{x_{y_{k-1}+j-1}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ has an inverse uncertainty distribution $\Psi^{-1}_{x_{y_{k-1}+j-1}}(\alpha)$ and is represented by a 99-table,

0.01	0.02	$\cdots$	0.99
$s^1_{x_{y_{k-1}+j-1}}$	$s^2_{x_{y_{k-1}+j-1}}$	$\cdots$	$s^{99}_{x_{y_{k-1}+j-1}}$

(2.102)

Then the completion time $C_{x_{y_{k-1}+j}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ has an inverse uncertainty distribution

$$\Psi^{-1}_{x_{y_{k-1}+j}}(\alpha) = \Psi^{-1}_{x_{y_{k-1}+j-1}}(\alpha) + \Phi^{-1}_{x_{y_{k-1}+j}k}(\alpha) \tag{2.103}$$

and has a 99-table,

0.01	$\cdots$	0.99
$s^1_{x_{y_{k-1}+j-1}} + t^1_{x_{y_{k-1}+j}k}$	$\cdots$	$s^{99}_{x_{y_{k-1}+j-1}} + t^{99}_{x_{y_{k-1}+j}k}$

where $s^1_{x_{y_{k-1}+j-1}}, s^2_{x_{y_{k-1}+j-1}}, \cdots, s^{99}_{x_{y_{k-1}+j-1}}$ are determined by (2.102), and $t^1_{x_{y_{k-1}+j}k}, t^2_{x_{y_{k-1}+j}k}, \cdots, t^{99}_{x_{y_{k-1}+j}k}$ are determined by (2.99). This recursive process may produce all completion times of jobs.

Makespan

Note that, for each k $(1 \le k \le m)$, the value $C_{x_{y_k}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})$ is just the time that the machine k finishes all jobs assigned to it, and has a 99-table,

0.01	0.02	$\cdots$	0.99
$s^1_{x_{y_k}}$	$s^2_{x_{y_k}}$	$\cdots$	$s^{99}_{x_{y_k}}$

(2.104)

Thus the makespan of the schedule $(\boldsymbol{x}, \boldsymbol{y})$ is determined by

$$f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) = \max_{1 \le k \le m} C_{x_{y_k}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}) \tag{2.105}$$

whose inverse uncertainty distribution is

$$\Upsilon^{-1}(\alpha) = \max_{1 \le k \le m} \Psi^{-1}_{x_{y_k}}(\alpha) \tag{2.106}$$

and has a 99-table,

0.01	0.02	$\cdots$	0.99
$\bigvee_{k=1}^{m} s^1_{x_{y_k}}$	$\bigvee_{k=1}^{m} s^2_{x_{y_k}}$	$\cdots$	$\bigvee_{k=1}^{m} s^{99}_{x_{y_k}}$

(2.107)

Machine Scheduling Model

In order to minimize the expected makespan $E[f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})]$, we have the following machine scheduling model,

$$\begin{cases} \min E[f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi})] \\ \text{subject to:} \\ \quad 1 \le x_i \le n, \quad i = 1, 2, \cdots, n \\ \quad x_i \ne x_j, \quad i \ne j, \ i, j = 1, 2, \cdots, n \\ \quad 0 \le y_1 \le y_2 \cdots \le y_{m-1} \le n \\ \quad x_i, y_j, \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m-1, \quad \text{integers.} \end{cases} \tag{2.108}$$

By using (2.107), the machine scheduling model is simplified as follows,

$$\left\{\begin{array}{l} \min \left(\bigvee_{k=1}^{m} s^1_{x_{y_k}} + \bigvee_{k=1}^{m} s^2_{x_{y_k}} + \cdots + \bigvee_{k=1}^{m} s^{99}_{x_{y_k}} \right) /99 \\ \text{subject to:} \\ \qquad 1 \le x_i \le n, \quad i = 1, 2, \cdots, n \\ \qquad x_i \ne x_j, \quad i \ne j, \ i, j = 1, 2, \cdots, n \\ \qquad 0 \le y_1 \le y_2 \cdots \le y_{m-1} \le n \\ \qquad x_i, y_j, \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, m-1, \quad \text{integers.} \end{array}\right. \tag{2.109}$$

Numerical Experiment

Assume that there are 3 machines and 7 jobs with the following linear uncertain processing times

$$\xi_{ik} \sim \mathcal{L}(i, i+k), \quad i = 1, 2, \cdots, 7, \ k = 1, 2, 3$$

where i is the index of jobs and k is the index of machines. In order to find an optimal machine schedule, we integrate the 99-method and a genetic algorithm to produce a hybrid intelligent algorithm. A run of the computer program (http://orsc.edu.cn/liu/resources.htm) shows that the optimal machine schedule is

Machine 1: $1 \to 4 \to 5$
Machine 2: $3 \to 7$
Machine 3: $2 \to 6$

whose expected makespan is 12.

2.12 Exercises

In order to enhance your ability in modeling, this section provides some exercises.

Exercise 2.1: One approach to improve system reliability is to provide redundancy for components in a system. There are two ways to provide component redundancy: parallel redundancy and standby redundancy. In parallel redundancy, all redundant elements are required to operate simultaneously. This method is usually used when element replacements are not permitted during the system operation. In standby redundancy, one of the redundant elements begins to work only when the active element fails. This method is usually employed when the replacement is allowable and can be finished immediately. The system reliability design is to determine the optimal number of redundant elements for balancing system performance and total cost. Assume the element lifetimes are uncertain variables with known uncertainty distributions. Please construct an uncertain programming model for the system reliability design.

Exercise 2.2: The facility location problem is to find locations for new facilities such that the conveying cost from facilities to customers is minimized. In practice, some factors such as demands, allocations, even locations of customers and facilities are changing and then are assumed to be uncertain variables with known uncertainty distributions. Please construct an uncertain programming model for the facility location problem.

Exercise 2.3: The inventory problem (or supply chain) is concerned with the issues of *when to order* and *how much to order* of some goods. The purpose is to obtain the right goods in the right place, at the right time, and at low cost. Assume the demands and prices are uncertain variables with known uncertainty distributions. Please construct an uncertain programming model to determine the optimal order quantity.

Exercise 2.4: The capital budgeting problem (or portfolio selection) is concerned with maximizing the total profit subject to budget constraint by selecting appropriate combination of projects. Assume the future returns are uncertain variables with known uncertainty distributions. Please construct an uncertain programming model to determine the optimal investment plan.

Exercise 2.5: One of the basic network optimization problems is the shortest path problem which is to find the shortest path between two given nodes in a network, where the arc lengths are assumed to be uncertain variables. Please construct an uncertain programming model to find the shortest path.

Exercise 2.6: The maximal flow problem is related to maximizing the flow of some commodity through the arcs of a network from a given origin to a given destination, where each arc has an uncertain capacity of flow. Please construct an uncertain programming model to discover the maximum flow.

Exercise 2.7: The transportation problem is to determine the optimal transportation plan of some goods from suppliers to customers such that the total transportation cost is minimum. Assume the unit transportation cost of each route is an uncertain variable. Please construct an uncertain programming model to solve the transportation problem.