Multicut L-Shaped Algorithm for Stochastic Convex Programming with Fuzzy Probability Distribution

Miaomiao Han  
School of Mathematics and Physics  
North China Electric Power University  
Baoding, Hebei Province, China  
Hanmiaomiao_1@163.com

Xinshun MA  
School of Mathematics and Physics  
North China Electric Power University  
Baoding, Hebei Province, China  
xsma@ncepubd.edu.cn

Abstract—Two-stage problem of stochastic convex programming with fuzzy probability distribution is studied in this paper. Multicut L-shaped algorithm is proposed to solve the problem based on the fuzzy cutting and the minimax rule. Theorem of the convergence for the algorithm is proved. Finally, a numerical example about two-stage convex recourse problem shows the essential character and the efficiency.

Keywords—stochastic convex programming; fuzzy probability distribution; two-stage problem; multicut L-shaped algorithm

1. Introduction

Stochastic programming is an important class of mathematical programming with random parameters, and has been widely applied to various fields such as economic management and optimization control[1]. Two-stage stochastic programming is a kind of mathematical programming where the decision variables and the decision process can be decomposed into two stages based on random parameters are observed before and after the specific value.

Stochastic linear programming as a basic issue has been studied widely, and many research results have been reported. In [2], two-stage problem of stochastic linear programming and the basic algorithm were first proposed and applied to the linear optimal control problem. Since then, a large variety of algorithms including Benders decomposition[3][4], stochastic decomposition[4], subgradient decomposition[5], nested decomposition[6], and disjunctive decomposition[8] for the two-stage stochastic linear programming had been developed. Among these methods, Benders decomposition also called the L-shaped method has become the main approach to deal with stochastic programming problems.

The theories and algorithms obtained before on stochastic linear programming all are based on a hypothesis that the probability distributions of random parameters have completely information. However, in many situations, due to lack of the date, the probability of a random event is not fully known, and need to get an approximate range with the help of experts’ experience. Recently, model of the stochastic linear program with fuzzy probability distribution was proposed in [9], and the modified L-shaped method has become the main approach to deal with stochastic programming problems.

Stochastic convex programming is an important class of stochastic nonlinear program and has more widely application than stochastic linear programming[10]. As a result, stochastic convex programming with fuzzy probability distribution will have more useful in many practical situations. In this paper, two-stage stochastic convex programming with fuzzy probability distribution and the solving method are studied, a numerical example shows the essential character and the efficiency.

2. Two-stage stochastic convex programming under fuzzy probability distribution

Let \((\Omega, \Sigma, P)\) be a probability space, sample space \(\Omega = \{\omega_1, \cdots, \omega_s\}\) is a finite set, and \(\Sigma = 2^\Omega\) is the \(\sigma\)-algebra composed by power set of \(\Omega\), \(p_i = P\{\omega_i = \omega\}\). The two-stage stochastic convex programming problem is

\[
\begin{align*}
\min & \quad f(x) + \sum_{i=1}^{s} p_i Q(x, \omega_i) \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

(1)

where,

\[
Q(x, \omega_i) = \min_{y} g(y, \omega_i) \\
\text{s.t.} \quad Wy = h(\omega_i) - T(\omega_i)x \\
& \quad y \geq 0
\]

(2)

\(x \in \mathbb{R}^n\) and \(y \in \mathbb{R}^n\) are decision vectors, \(f(x)\) is convex function, \(A\) is \(m_1 \times n_1\) matrix, \(b \in \mathbb{R}^m\) is known vector, \(W\) is \(m_2 \times n_2\) recourse matrix, for each \(\omega_i \in \Omega\), \(g(y, \omega_i)\) is convex function on \(y\), \(h(\omega_i) \in \mathbb{R}^m\) is vector, and \(T(\omega_i)\) is
where $x$ and $y$ are the first stage decision variable and the second stage decision variable respectively. The mathematical expected value $E[Q(x, o_i)] = \sum_{i=1}^{N} p_i Q(x, o_i)$.

When the random variable obeys fuzzy probability distribution, the scope of $p_i$ is as follows:

$$
\pi_o = \{ P \in \mathbb{R}^N \mid d_i - (1 - \alpha_i)l_i \leq p_i \leq d_i + (1 - \alpha_i)l_i, \sum_{i=1}^{N} p_i = 1; p_i \geq 0, i = 1, \cdots, N \}
$$

(3)

where $P = (p_1, p_2, \cdots, p_N)^T \in \mathbb{R}^N$ consists of probabilities, $l = (l_1, \cdots, l_N)^T$ denotes the vagueness level, and the level value $\alpha(0 \leq \alpha_i \leq 1)$ expresses the DM credibility degree of the partial information on probability distribution. The fuzzy probability distributions results in that mathematical expectation $E[Q(x, o_i)]$ is uncertain, here, $\max_{p_{ox}} E_r[Q(x, o_i)] = \max_{p_{ox}} \sum_{i=1}^{N} p_i Q(x, o_i)$ will be used instead of $E[Q(x, o_i)]$, and then (1) can be expressed as follows

$$
\begin{align*}
\min f(x) + \max_{p_{ox}} \sum_{i=1}^{N} p_i Q(x, o_i) \\
\text{s.t.} \quad Ax = b, x \geq 0
\end{align*}
$$

(4)

where $Q(x, o_i)$ is confirmed by (2).

Obviously, for a given $x$, there exists $\bar{P} = (\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_N)^T \in \pi_o$, such that $\sum_{i=1}^{N} \bar{p}_i Q(x, o_i) = \max_{p_{ox}} \sum_{i=1}^{N} p_i Q(x, o_i)$.

I. MULTICUT L-SHAPED ALGORITHM

The problem (4) is equivalent to:

$$
\begin{align*}
\min f(x) + \theta, \\
\text{s.t.} \quad \max_{p_{ox}} \sum_{i=1}^{N} p_i Q(x, o_i) \leq \theta, \\
x \in K = K_1 \cap K_2,
\end{align*}
$$

(5)

where $Q(x, o_i) = \min_{y_i} g(y_i, o_i)$,

$$
\begin{align*}
\text{s.t.} \quad W_y = h(o_i) - T(o_i)x, \\
\quad y_i \geq 0,
\end{align*}
$$

(6)

and

$$
\begin{align*}
K_1 = \{ x \mid Ax = b, x \geq 0 \}, \\
K_2 = \{ x \mid \forall o_i \in \Omega, \exists y_i \geq 0, s.t. W_y = h(o_i) - T(o_i)x \}.
\end{align*}
$$

The standard L-shaped algorithm for solving above problem can be designed under outer linearization (see e.g. [9]). Suppose that $\tilde{F} = (\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_N)^T$ is solution of

$$
\begin{align*}
\max \sum_{i=1}^{N} p_i Q(x, o_i),
\end{align*}
$$

(7)

because of each constrain in (5) corresponds to $N$ constraints in (7).

The multicut L-shaped algorithm is defined as follows:

**S0.** Set $s = k = 0$, $t_i = 0$ for all $i = 1, 2, \cdots, N$, and $x^\circ$ is given.

**S1.** Set $k = k + 1$, solve the following master problem:

$$
\begin{align*}
\min f(x) + \sum_{i=1}^{N} \bar{p}_i \theta_i \\
\text{s.t.} \quad Q(x, o_i) \leq \theta_i, \quad \text{for all } i \\
x \in K = K_1 \cap K_2
\end{align*}
$$

(8)

Let $(x^\circ, \theta_1^\circ, \cdots, \theta_N^\circ)$ be an optimal solution of problem (8). Note that if no constraint (a4) is present for some $i$, $\theta_i^\circ$ is set equal to $-\infty$, $\theta_i^\circ$ and $\bar{p}_i$ are not considered in the calculation of $x^\circ$. Go to S2.

**S2.** For $i = 1, \cdots, N$, solve the following linear programming problems

$$
\begin{align*}
\min z^i = e^T u^i + e^T u^i \\
\text{s.t.} \quad W_y + u^i - u^i = h(o_i) - T(o_i)x^i \\
y \geq 0, u^i \geq 0, u^i \geq 0
\end{align*}
$$

(9)

where $e^T = (1, \cdots, 1)$, until, for some $i$, if the optimal value $z^i > 0$, let $\sigma^i$ be optimal dual variables value, and define

$$
\begin{align*}
D_{i+1} &= (\sigma^i)^T T(o_i) \\
d_{i+1} &= (\sigma^i)^T h(o_i)
\end{align*}
$$

set $s = s + 1$, add the constraint $D_{i+1}x^i \geq d_{i+1}$ to the set (a3) and return to S1. If for all $i, z_i = 0$, then go to S3.

**S3.** For $i = 1, \cdots, N$, and a fixed $x^\circ$, solve the following convex programming problems

$$
\begin{align*}
\min g(y_i, o_i) \\
\text{s.t.} \quad W_y = h(o_i) - T(o_i)x^i \\
y_i \geq 0
\end{align*}
$$

(10)
Let \( Q(x^*, \omega) \) be the optimal value, and \( y^* \) the optimal solution. Solve the problem \( \max_{x \in \mathcal{X}} \sum_{p(j)} p_j Q(x^j, \omega_j) \), and suppose \((\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N)^T \) is the optimal solution, then update the objective function of the master problem. Let \( v_i^j \) and \( u_i^j \) be the optimal dual variables associated with constrain (a6) and (a7) respectively. Compute

\[
\begin{align*}
E_{i+1} &= -(v_i^j)^T T(\omega_j), \\
e_i^j &= g(y_i, \omega_j) - (u_i^j)^T y^j_i + (v_i^j)^T \left[ W(y_i^j) - h(\omega_j) \right].
\end{align*}
\]

If \( \theta_i^j < e_i^j - E_{i+1}x^j \) does not hold for any \( i = 1, \ldots, N \), stop, then \((x^j, \bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N)^T \) is an optimal solution.

Otherwise, set \( t_j = t_j + 1 \), add the constraint \( \theta_i^j \geq e_i^j - E_{i+1}x^j \) to the set (a4), return to S1.

### 3. Theorem of convergence for the algorithm

**Proposition 1.** In the algorithm, constraint set (a3) is finite.

**Proof** The proof of this proposition is the same to the standard L-shaped algorithm (see e.g. [2]).

**Proposition 2.** For any \( \omega \in \Omega \) and on all \( x \in \mathcal{K} \), \( Q(x, \omega) \) is either a finite convex function or is identically \(-\infty\),

\[
\mathcal{K} = \{ x | \omega \in \Omega, \exists y \geq 0, s.t. \ W y_i = h(\omega) - T(\omega) x \}.
\]

**Proof** (see e.g. [10])

**Proposition 3.** If \( Q(x, \omega) \) is a finite convex function for each \( \omega \in \Omega \), then the function \( e_{i+1} - E_{i+1}x \) is linear supporting hyper planes of \( Q(x, \omega) \).

**Proof** By the duality theory (see e.g. [11]), it holds that

\[
Q(x^j, \omega_j) = g(y^j, \omega_j) - (u_i^j)^T y^j_i + (v_i^j)^T \left[ W(y^j_i) - h(\omega_j) + T(\omega_j)x^j_i \right].
\]

We know from the convexity of \( Q(x, \omega) \) that

\[
Q(x, \omega) \geq g(y^j, \omega_j) - (u_i^j)^T y^j_i + (v_i^j)^T \left[ W(y^j_i) - h(\omega_j) \right]
\]

\[
+(v_i^j)^T T(\omega_j)x
\]

Thus

\[
g(y_i, \omega_j) - (u_i^j)^T y^j_i + (v_i^j)^T \left[ W(y^j_i) - h(\omega_j) \right] + (v_i^j)^T T(\omega_j)x
\]

is a linear support of \( Q(x, \omega) \).

**Theorem.** Suppose that the algorithm generates an infinite sequence of \((x^j, \theta_1^j, \ldots, \theta_N^j) \). If \((\bar{\theta}_1, \bar{\theta}_2, \ldots, \bar{\theta}_N) \) is the limit point of an arbitrary subsequence \((x^j, \theta_1^j, \ldots, \theta_N^j) \), and for each \( i \), \( \lim_{j \to \infty} (e_i^j - E_{i+1}x^j - \theta_i^j) = 0 \), \( i = 1, \ldots, N \), then

\[
(\bar{x}, \bar{\theta}_1, \ldots, \bar{\theta}_N)
\]

is an optimal solution of problem(7); \( \bar{x} \) is an optimal solution of problem(4).

**Proof** Since the number of the constraints of type (a3) is finite, we have that \( x^j \in \mathcal{K} \) for \( k \) sufficiently large. We also know \( \mathcal{K} \) is closed convex set, then \( x \in \mathcal{K} \). By known, for each \( i \),

\[
\theta_i^j \geq Q(x_i^j, \omega_i) = e_i^j - E_{i+1}x_i^j,
\]

and

\[
\lim_{j \to \infty} (e_i^j - E_{i+1}x_i^j - \theta_i^j) = 0
\]

Then

\[
\bar{\theta} = Q(\bar{x}, \omega)
\]

for all \( i = 1, \ldots, N \). Thus, \((\bar{x}, \bar{\theta}_1, \ldots, \bar{\theta}_N) \) is a feasible solution of problem(7).

On the other hand, if \( x^j \) is optimal solution to the minimax problem(4), but not necessarily an optimal solution in iteration \( k_j \), then

\[
f(x^j) + \max_{p \in \Omega} \sum_{p(j)} p_j Q(x^j, \omega_j) \leq f(x^j) + \max_{p \in \Omega} \sum_{p(j)} p_j Q(x^j, \omega_j).
\]

By continuity of the convex function we have that

\[
f(\bar{x}) + \max_{p \in \Omega} \sum_{p(j)} p_j Q(x^j, \omega_j) \leq f(\bar{x}) + \max_{p \in \Omega} \sum_{p(j)} p_j Q(x^j, \omega_j),
\]

then, for a certain value \((\bar{\theta}_1, \bar{\theta}_2, \ldots, \bar{\theta}_N) \in \mathcal{P} \),

\[
f(\bar{x}) + \sum_{p(j)} p_j Q(x^j, \omega_j) \leq f(\bar{x}) + \max_{p \in \Omega} \sum_{p(j)} p_j Q(x^j, \omega_j).
\]

Hence, \((\bar{x}, \bar{\theta}_1, \ldots, \bar{\theta}_N) \) is an optimal solution to problem(7), and \( \bar{x} \) is an optimal solution to problem(4).

### 4. Numerical example

Consider the following two-stage stochastic convex programming

\[
\begin{align*}
\min & -x_1 + 3x_2 \\
& x_1 + x_2 \geq 5, \ x_1 \leq x_2,
\end{align*}
\]

\[
\begin{align*}
x_1 + x_2 & \leq 3, \ y_1 = x_1, \\
y_1 & \leq \xi_1, \ y_2 \leq \xi_2, \\
x_1 + x_2, y_1, y_2 & \geq 0.
\end{align*}
\]

(11)

where \( \xi_1 \) takes the three values 3.5, 3.8 and 4.0 with probability 1/3, that \( \xi_2 \) takes the values 0.5, 1.0 and 1.5 with probability 1/3, and that \( \xi_1 \) and \( \xi_2 \) are independent of each other, then \( \xi = (\xi_1, \xi_2)^T \) can take each vector in the set

\[
\Pi = \{(k_1, k_2) \}^T \mid \ k_1 = 3.5, 3.8, 4.0, \ k_2 = 0.5, 1.0, 1.5 \} \text{ with probability } 1/9.
\]

Under fuzzy probability distribution, assume that \( \xi \) takes the each values in \( \Pi \) with probability around 1/9, i.e., \( p_i \equiv 1/9 \) \( (i = 1, 2, \ldots, 9) \), it can be confirmed by(3), where \( d_i = 1/12 \).
Then we get two-stage stochastic convex programming with fuzzy probability distribution. We solve problem (11) by the proposed algorithm and take the initial value \( x^0 = (1, 1)^T \). Iterations procedure and outputs are as follows.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( \text{obj.val} )</th>
<th>( x^k )</th>
<th>( \pi^k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-20.167</td>
<td>(3.000, 0.000)</td>
<td>(0.153, 0.111, 0.069, 0.153, 0.111, 0.069, 0.153, 0.111, 0.069)</td>
</tr>
<tr>
<td>2</td>
<td>-15.947</td>
<td>(2.356, 0.644)</td>
<td>(0.111, 0.111, 0.111, 0.111, 0.111, 0.111)</td>
</tr>
<tr>
<td>3</td>
<td>-14.454</td>
<td>(2.566, 0.434)</td>
<td>(0.153, 0.111, 0.069, 0.153, 0.111, 0.069, 0.153, 0.111, 0.069)</td>
</tr>
<tr>
<td>4</td>
<td>-14.020</td>
<td>(2.501, 0.499)</td>
<td>(0.153, 0.090, 0.090, 0.153, 0.090, 0.090)</td>
</tr>
<tr>
<td>5</td>
<td>-14.007</td>
<td>(2.499, 0.501)</td>
<td>(0.153, 0.090, 0.090, 0.153, 0.090, 0.090)</td>
</tr>
<tr>
<td>6</td>
<td>-14.005</td>
<td>(2.500, 0.500)</td>
<td>(0.153, 0.090, 0.090, 0.153, 0.090, 0.090)</td>
</tr>
<tr>
<td>7</td>
<td>-14.005</td>
<td>(2.500, 0.500)</td>
<td>(0.153, 0.090, 0.090, 0.153, 0.090, 0.090)</td>
</tr>
<tr>
<td>8</td>
<td>-14.005</td>
<td>(2.500, 0.500)</td>
<td>(0.153, 0.090, 0.090, 0.153, 0.090, 0.090)</td>
</tr>
<tr>
<td>( t )</td>
<td>(8, 8, 7, 8, 5, 6, 8, 5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where \( \text{obj.val} \) refers to the objective value, \( t = (t_1, \ldots, t_8) \) is the vector on the number of iterations of each scenario.

5. Conclusion

Two-stage stochastic convex programming with fuzzy probability distribution is proposed in this paper. The multicut L-shaped algorithm for solving the problem is presented, and the theorem of convergence is given. Finally, a numerical test example demonstrates the essential character and the efficiency of the algorithm.

6. Acknowledgment

This work is supported partially by the Natural Science Foundation of Hebei Province under Grant A2012502061.

REFERENCES