Journal of Computations & Modelling, vol.2, no.4, 2012, 133-148
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2012

Dynamic Slope Scaling Procedure to solve Stochastic Integer Programming Problem

Takayuki Shiina[1] and Chunhui Xu[2]

Abstract

Stochastic programming deals with optimization under uncertainty. A stochastic programming problem with recourse is referred to as a two-stage stochastic problem. We consider the stochastic programming problem with simple integer recourse in which the value of the recourse variable is restricted to a multiple of a nonnegative integer. The algorithm of a dynamic slope scaling procedure to solve the problem is developed by using the property of the expected recourse function. The numerical experiments show that the proposed algorithm is quite efficient. The stochastic programming model defined in this paper is quite useful for a variety of design and operational problems.

Mathematics Subject Classification: 90C11, 90C15, 90C90
Keywords: Stochastic programming problem with recourse, Simple integer recourse, Dynamic slope scaling procedure

---

[1] Chiba Institute of Technology, e-mail : shiina.takayuki@it-chiba.ac.jp
[2] Chiba Institute of Technology, e-mail : xu.chunhui@it-chiba.ac.jp

Article Info: *Received* : September 19, 2012. *Revised* : October 23, 2012
Published online : December 30, 2012

1 Introduction

Mathematical programming has been applied to many problems in various fields. However for many actual problems, the assumption that the parameters involved in the problem are deterministic known data is often unjustified. These data contain uncertainty and are thus represented as random variables, since they represent information about the future. Decision-making under uncertainty involves potential risk. Stochastic programming (Birge [1], Birge and Louveaux [2], Kall and Wallace [3]) deals with optimization under uncertainty. A stochastic programming problem with recourse is referred to as a two-stage stochastic problem. In the first stage, a decision has to be made without complete information on random factors. After the value of random variables are known, recourse action can be taken in the second stage. For the continuous stochastic programming problem with recourse, an L-shaped method (Van Slyke and Wets [4]) is well-known.

The L-shaped method was used to solve stochastic programs having discrete decisions in the first stage (Laporte and Louveaux[5]). This method was applied to solve a stochastic concentrator location problem (Shiina [6, 7]).

In this paper, we consider a stochastic programming problem in which the recourse variables are restricted to integers. If integer variables are involved in a second stage problem, optimality cuts based on the Benders [8] decomposition do not provide facets of the epigraph of recourse function. It is difficult to approximate the recourse function which is in general nonconvex and discontinuous, since the function is defined as the value function of the second stage integer programming problem.

For stochastic programs with simple integer recourse, Ahmed, Tawarmalani, and Sahinidis[9] developed a finite algorithm based on the branching of the first stage integer variables. However, variables involved in the stochastic program with simple integer recourse are restricted to having a nonnegative integer value. Such restriction of variables to pure integers makes application of the problem difficult. Therefore, we consider a practical stochastic programming model which is applicable to various real problems, and deal with the problem in which the recourse variables are restricted to take the multiples of some nonnegative integer. These recourse variables represent that the additional actions are taken in a certain amount units. This mathematical programming model is quite useful for a variety of design and operational problems which arise in diverse contexts, such as investment planning, capacity expansion, network design and facility location.

In Section 2, the basic model of the stochastic programming problem with recourse and the L-shaped method are shown. Then, we consider the variant of the stochastic program with simple integer recourse, which is a natural extension of the continuous simple recourse. In Section 3, we investigate the property of the recourse function. The algorithm of a dynamic slope scaling procedure to solve the problem is developed by using the property of the expected recourse function. In Section 4, the numerical experiments show that the proposed algorithm is quite efficient. The stochastic programming model defined in this paper is quite useful for a variety of design and operational problems.

2 Formulation

2.1 Stochastic programming problem with recourse

We first form the basic two-stage stochastic linear programming problem with recourse as (SPR).

$$
\begin{array}{rl}
\text{(SPR): min} & c^\top x + \mathcal{Q}(x) \\
\text{subject to} & Ax = b \\
& x \geq 0 \\
& \mathcal{Q}(x) = E_{\tilde{\xi}}[Q(x, \tilde{\xi})] \\
& Q(x, \xi) = \min\{q(\xi)^\top y(\xi) \mid Wy(\xi) = h(\xi) - T(\xi)x, y(\xi) \geq 0\}, \xi \in \Xi
\end{array}
$$

In the formulation of (SPR), c is a known n_1-vector, b a known m_1-vector, and A and W are known matrices of size $m_1 \times n_1$ and $m_2 \times n_2$, respectively. The first stage decisions are represented by the n_1-vector x. We assume the l-random vector $\tilde{\xi}$ is defined on a known probability space. Let Ξ be the support of $\tilde{\xi}$, i.e. the smallest closed set such that $P(\Xi) = 1$.

Given a first stage decision x, the realization of random vector ξ of $\tilde{\xi}$ is observed. The second stage data m_2-vector $h(\xi)$, n_2-vector $q(\xi)$ and $m_2 \times n_1$ matrix $T(\xi)$ become known. Then, the second stage decision $y(\xi)$ must be taken so as to satisfy the constraints $Wy(\xi) = \xi - Tx$ and $y(\xi) \geq 0$. The second stage decision $y(\xi)$ is assumed to cause a penalty of $q(\xi)$. The objective function contains a deterministic term $c^\top x$ and the expectation of the second stage objective. The symbol $E_{\tilde{\xi}}$ represents the mathematical expectation with respect to $\tilde{\xi}$, and the function $Q(x, \xi)$ is called the recourse function in state ξ. The value of the recourse function is given by solving a second stage linear programming problem.

It is assumed that the random vector $\tilde{\xi}$ has a discrete distribution with finite support $\Xi = \{\xi^1, \ldots, \xi^S\}$ with $\text{Prob}(\tilde{\xi} = \xi^s) = p^s, s = 1, \ldots, S$. A particular realization ξ of the random vector $\tilde{\xi}$ is called a scenario. Given the finite discrete distribution, the problem (SPR) is restated as (SPR'), the deterministic equivalent problem for (SPR).

$$
\begin{array}{rl}
\text{(SPR'): } \min & c^\top x + \displaystyle\sum_{s=1}^{S} p^s Q(x, \xi^s) \\
\text{subject to} & Ax = b \\
& x \geq 0 \\
& Q(x, \xi^s) = \min\{q(\xi^s)^\top y(\xi^s) \mid Wy(\xi^s) = h(\xi^s) - T(\xi^s)x, \\
& \quad y(\xi^s) \geq 0\}, s = 1, \ldots, S
\end{array}
$$

The problem (SPR') is reformulated as (DEP-SPR) setting $y(\xi^s)$, $q(\xi^s), T(\xi^s), h(\xi^s)$, $Q(x, \xi^s)$ as y^s, q^s, T^s, h^s, ${q^s}^\top y^s$, respectively.

$$
\begin{array}{rl}
\text{(DEP-SPR) :} & \\
\min_{x, y^1, \cdots, y^s} & cx + \displaystyle\sum_{s=1}^{S} p^s {q^s}^\top y^s \\
\text{subject to} & Ax = b \\
& Wy^s = h^s - T^s x, s = 1, \ldots, S \\
& x \geq 0, y^s \geq 0, s = 1, \ldots, S
\end{array}
$$

To solve (DEP-SPR), an L-shaped method (Van Slyke and Wets [4]) has been used. This approach is based on Benders [8] decomposition. The expected recourse function is piecewise linear and convex, but it is not given explicitly in advance. In the algorithm of the L-shaped method, we solve the following problem (MASTER). The new variable θ denotes the upper bound for the expected recourse function such that $\theta \geq \sum_{s=1}^{S} p^s Q(x, \xi^s)$.

$$\left|\begin{array}{rl} \text{(MASTER): min} & c^\top x + \theta \\ \text{subject to} & Ax = b \\ & x \geq 0 \\ & \theta \geq 0 \end{array}\right.$$

The recourse function is given by an outer linearization using a set of feasibility and optimality cuts as shown in Figure 1. In the case of $n_2 = 2 \times m_2$ and

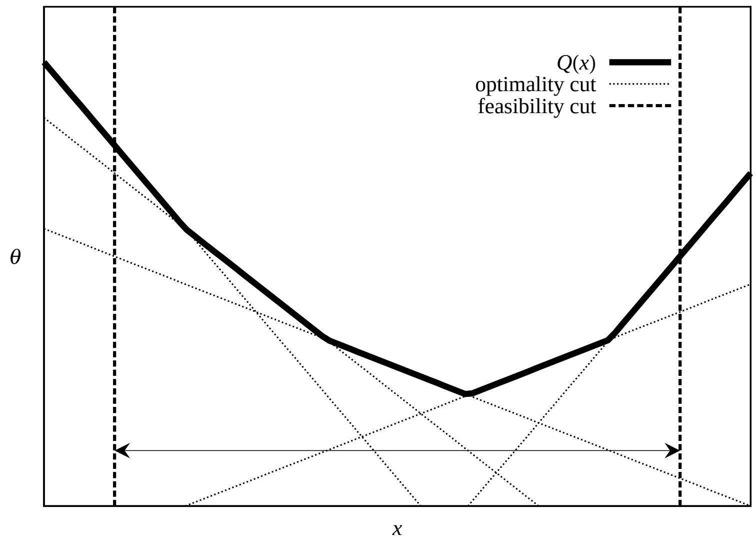

Figure 1: L-shaped method

$W = (I, -I)$, the problem (SPR) is said to have a simple recourse.

2.2 Simple integer recourse

In this section, we consider the special case of (SPR) setting $q(\xi) = q(> 0), T(\xi) = T$, $h(\xi) = \xi$ and $W = rI$, where r is a positive integer. Furthermore, we define the constraints of the recourse problem as $y(\xi) \geq \xi - Tx, y(\xi) \geq 0$ taking acount of the relationship between the value of the random variable ξ and the first stage decision variable Tx. The size of the random vector $\tilde{\xi}$ is defined as $l = m_2$, and the size of the recourse variable $y(\xi)$ is $n_2 = m_2$. Then we define the new variables $\chi = Tx$, where χ is called a tender to be bid against random outcomes.

In the case the recourse variables are defined as the nonnegative integer variables, the problem is called to have a simple integer recourse. For this problem, the constraints of the recourse problem are $y(\xi) \geq \xi - \chi, y(\xi) \in Z_+^{n_2}$. The opmimal solution of the recourse problem is a mimimal nonnegative integer satisfying $y(\xi) \geq \xi - \chi$.

As the recourse decisions are represented as urgent and additional production, order, or investment, the recourse decisions are taken in a certain amount of unit. Louveaux-van der Vlerk [10] presented the lower and upper bounds for the problem. But considering the application of the mathematical programming model to real problems, the recourse decisions should be modefied to take some batch size.

In this paper, we formulate the stochastic programming problem (SPSIR) in which the recourse variable $y(\xi)$ is defined as nonnegative integer variable and the recourse action $ry(\xi)$ is restricted to nonnegative multiple of some integer r.

$$\begin{array}{ll}
\text{(SPSIR):} & \\
\min & c^\top x + \Psi(\chi) \\
\text{subject to} & Ax = b, x \geq 0 \\
& Tx = \chi \\
& \Psi(\chi) = \sum_{s=1}^{S} p^s \psi(\chi, \xi^s) \\
& \psi(\chi, \xi^s) = \min\{q^\top y(\xi^s) | ry(\xi^s) \geq \xi^s - \chi, y(\xi^s) \in Z_+^{n_2}\}, s = 1, \ldots, S
\end{array}$$

3 Solution Algorithm

3.1 Property of the recourse function

In this section, we investigate the property of the recourse function. The optimal solution of the recourse problem is obtained as follows.

$$y(\xi^s)_i = \begin{cases} \lceil \frac{\xi_i^s - \chi_i}{r} \rceil, & \text{if } \chi_i < \xi_i^s \\ 0 & \text{if } \xi_i^s \leq \chi_i \end{cases}, i = 1, \ldots, m_2$$

It is shown that the recourse function $\psi(\chi, \xi)$ is separable in the elements of $\chi^\top = (\chi_1, \ldots, \chi_{m_2})^\top$. We define $\psi_i(\chi_i, \xi_i) = \min\{q_i y(\xi)_i \mid ry(\xi)_i \geq \xi_i -$

$\chi_i, y(\xi)_i \in Z_+\}$ in the following equation.

$$\begin{aligned}\psi(\chi,\xi) &= \min\{q^\top y(\xi) \mid ry(\xi) \geq \xi - \chi, y(\xi) \in Z_+\} \\ &= \sum_{i=1}^{m_2} \min\{q_i y(\xi)_i \mid ry(\xi)_i \geq \xi_i - \chi_i, y(\xi)_i \in Z_+\} \\ &= \sum_{i=1}^{m_2} \psi_i(\chi_i, \xi_i) \end{aligned} \tag{1}$$

Let $\tilde{\xi}_i$ and Ξ_i be the i-th component of the random vector $\tilde{\xi}$ and the support of $\tilde{\xi}_i$, respectively. We make the following assumptions.

Assumption 3.1. *The random variables $\tilde{\xi}_i, i = 1, \ldots, n_2$ are independent and follow a discrete distribution.*

Assumption 3.2. *A probability p_i^s is associated with each outcome ξ_i^s, $s = 1, \ldots, |\Xi_i|$ of $\tilde{\xi}_i$. The random variable $\tilde{\xi}_i$ takes only positive values and is bounded as $0 < \xi_i^s < \infty, s = 1, \ldots, |\Xi_i|, i = 1, \ldots, n_2$.*

Then, the support of $\tilde{\xi}$ is described as $\Xi = \Xi_1 \times \cdots \times \Xi_{n_2}$. And the positive constant M can be taken so as to satisfy $M \geq \max\{\xi_i^s, s = 1, \ldots, |\Xi_i|, i = 1, \ldots, n_2\}$. From assumption 3.1, 3.2, the joint probability $P(\tilde{\xi} = \xi^s)$ is calculated as follows.

$$\begin{aligned}\text{Prob}(\tilde{\xi} = \xi^s) &= \text{Prob}(\tilde{\xi}_1 = \xi^{s_1}) \times \cdots \times \text{Prob}(\tilde{\xi}_{m_2} = \xi^{s_{m_2}}) \\ &= \prod_{i=1}^{m_2} \text{Prob}(\tilde{\xi}_i = \xi^{s_i}) \\ &= \prod_{i=1}^{m_2} p_i^{s_i} \end{aligned} \tag{2}$$

It is shown that the expected recourse function $\Psi(\chi)$ is also separable in $\chi_i, i = 1, \ldots, m_2$ as (3), where $\Psi_i(\chi_i) = \sum_{s=1}^{|\Xi_i|} p_i^s \psi_i(\chi_i, \xi_i^s)$ denotes the expectation of the i-th recourse function (3).

$$\begin{aligned}
\Psi(\chi) &= \sum_{s=1}^{S} p^s \psi(\chi, \xi^s) \\
&= \sum_{s_1=1}^{|\Xi_1|} \cdots \sum_{s_{n_2}=1}^{|\Xi_{n_2}|} p_1^{s_1} \cdots p_{n_2}^{s_{n_2}} \sum_{i=1}^{m_2} \psi_i(\chi_i, \xi_i^{s_i}) \\
&= \sum_{i=1}^{m_2} (\sum_{s_1=1}^{|\Xi_1|} \cdots \sum_{s_{n_2}=1}^{|\Xi_{n_2}|} p_i^{s_i} \prod_{\substack{j=1 \\ j \neq i}}^{m_2} p_j^{s_j}) \psi_i(\chi_i, \xi_i^{s_i}) \\
&= \sum_{i=1}^{m_2} \sum_{s_i=1}^{|\Xi_i|} p_i^{s_i} \psi_i(\chi_i, \xi_i^{s_i}) \\
&= \sum_{i=1}^{m_2} \Psi_i(\chi_i)
\end{aligned} \tag{3}$$

For the list of the realization of the random variable $\{\xi_i^1, \ldots, \xi_i^{|\Xi_i|}\}$, we sort $\xi_i^s, s = 1, \ldots, |\Xi_i|$ in non-decreasing order so as to satisfy $\xi_i^1 \leq \ldots \leq \xi_i^{|\Xi_i|}$ by substituting indices if required. The expectation of the recourse function $\psi_i(\chi_i, \xi_i)$ is shown as follows.

$$\Psi_i(\chi_i) = E_{\tilde{\xi}_i}[\psi_i(\chi_i, \tilde{\xi}_i)] = \sum_{s_i=1}^{|\Xi_i|} p_i^{s_i} q \lceil \frac{\xi_i^{s_i} - \chi_i}{r} \rceil^+ \tag{4}$$

The discontinuous breakpoints of the expected function $\Psi_i(\chi_i)$ are shown as 5 in the region $0 \leq \chi_i \leq \xi_i^{|\Xi_i|}$.

$$\chi_i = \xi_i^{s_i} - mr \quad (s_i = 1, \ldots, |\Xi_i|, m = 0, 1, \ldots, \lfloor \frac{\xi^{s_i}}{r} \rfloor) \tag{5}$$

The expected recourse function $\Psi_i(\chi_i)$ has at most $\sum_{s_i=1}^{|\Xi_i|}(\lfloor \frac{\xi_i^{s_i}}{r} \rfloor + 1)$ discontinuous points, and the length of the continuous region depends the value of the constant r.

For example, the expected recourse function $\Psi_i(\chi_i)$ in the case $\Xi = \{11, 22\}, p^1 = p^2 = 1/2, r = 5, q = 1$ is shown in Figure 2.

And the expected recourse function $\Psi_i(\chi_i)$ can be calculated using the distribution function F_i of $\tilde{\xi}_i$.

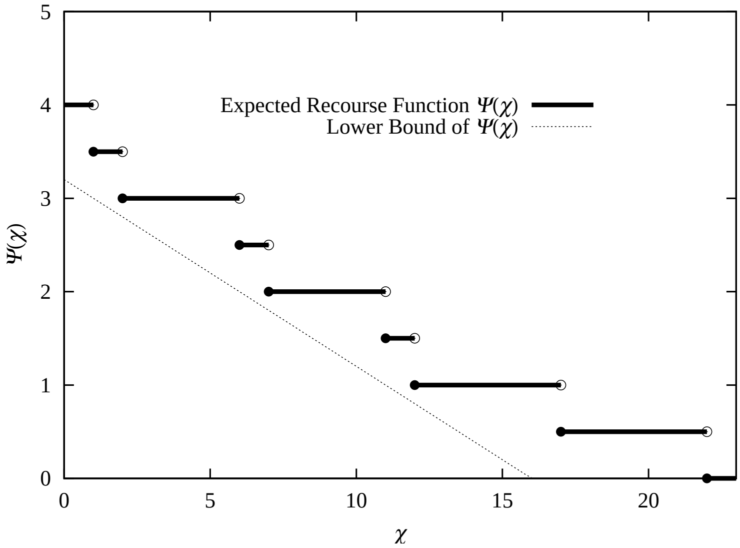

Figure 2: Expected recourse function

$$\begin{aligned}
\Psi_i(\chi_i) &= E_{\tilde{\xi}_i}\left[q_i\lceil\frac{\tilde{\xi}_i-\chi_i}{r}\rceil_+\right] \\
&= q_i\sum_{j=1}^{\infty} j\mathrm{Prob}(\lceil\frac{\tilde{\xi}_i-\chi_i}{r}\rceil_+=j) \\
&= q_i\sum_{j=1}^{\infty}\sum_{k=0}^{j-1}\mathrm{Prob}(\lceil\frac{\tilde{\xi}_i-\chi_i}{r}\rceil_+=j) \\
&= q_i\sum_{k=0}^{\infty}\sum_{j=k+1}^{\infty}\mathrm{Prob}(\lceil\frac{\tilde{\xi}_i-\chi_i}{r}\rceil_+=j) \\
&= q_i\sum_{k=0}^{\infty}\mathrm{Prob}(\frac{\tilde{\xi}_i-\chi_i}{r}>k) \\
&= q_i\sum_{k=0}^{\infty}(1-F_i(\chi_i+rk)) \qquad (6)
\end{aligned}$$

3.2 Algorithm of DSSP

Let (SPSIR$_{LP}$) be the problem in which the integer constraints are relaxed. The recourse function $\Psi(\chi)$ of the problem (SPSIR$_{LP}$) corresponds to the lower bound for the original $\Psi(\chi)$ of (SPSIR) as shown in Figure 2.

$$
\begin{array}{rl}
(\text{SPSIR}_{LP})\text{:min} & c^\top x + \Psi(\chi) \\
\text{subject to} & Ax = b, x \geq 0 \\
& Tx = \chi \\
& \Psi(\chi) = \sum_{s=1}^{S} p^s \psi(\chi, \xi^s) \\
& \psi(\chi, \xi^s) = \min\{q^\top y(\xi^s) | ry(\xi^s) \geq \xi^s - \chi, y(\xi^s) \geq 0\}, s = 1, \ldots, S
\end{array}
$$

After solving the problem (SPSIR$_{LP}$), the optimal solution $(x^{LP*}, \chi^{LP*}, y^{LP*}(\xi^1), \ldots, y^{LP*}(\xi^S))$ is obtained.

Next, we consider a heuristic algorithm to solve (SPFCRT). For the fixed charge network flow problem, Kim and Pardalos [11] developed an approach, called the dynamic slope scaling procedure (DSSP), which solves successive linear programming problems with recursively updated objective functions. Kim and Pardalos [12] modified DSSP, which repeats the reduction and refinement of the feasible region and the algorithm is effective when the objective function is staircase or sawtooth type. The algorithm of DSSP is used to obtain a good feasible solution to the second stage integer programming problem which defines the recourse function. The algorithm of DSSP is promising since the recourse function is monotonically nonincreasing as shown in Figure 2.

Let $(x^{LP*}, \chi^{LP*}, y^{LP*}(\xi^1), \ldots, y^{LP*}(\xi^S))$ be the optimal solution of the problem (SPSIR$_{LP}$). We compute the approximate value θ_i of $\Psi_i(\chi_i)$ using the following inequality (7).

$$
\theta_i \geq \frac{\Psi_i(\chi_i^{LP*})}{\chi_i^{LP*} - \xi_i^{|\Xi_i|}}(\chi_i - \chi_i^{LP*}) + \Psi_i(\chi_i^{LP*}) \tag{7}
$$

The constraint (7) provides the upper bounds for the linear function which connects $(\xi_i^{|\Xi_i|}, 0)$ and $(\chi_i^{LP*}, \Psi_i(\chi_i^{LP*}))$. The value of θ_i gives the exact value of $\Psi_i(\chi_i)$ at these two points.

Taking accounts of the breakpoints (5) of the recourse function, we set the lower and upper bounds for the variable χ_i. Let the breakpoints of the recourse function $\Psi_i(\chi_i)$ be $0 < \bar{\chi}_i^1 \leq \bar{\chi}_i^2 \leq \cdots \leq \bar{\chi}_i^w$, and we define $\bar{\chi}_i^0 = 0$.

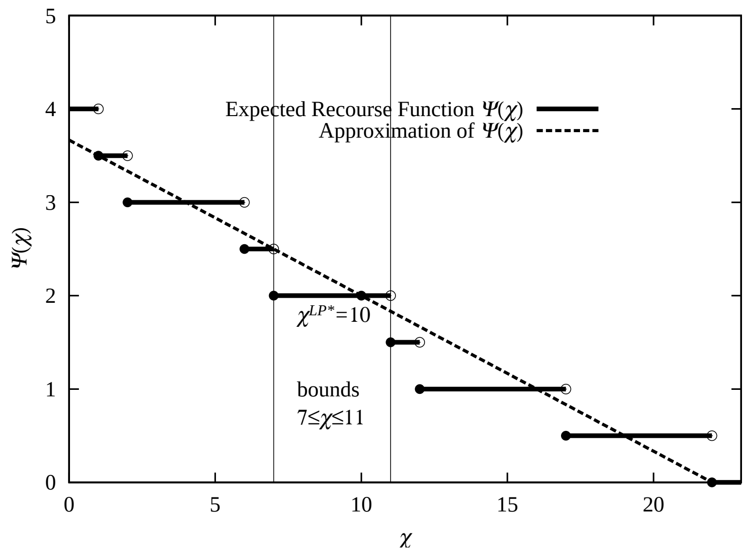

Figure 3: Algorithm of DSSP

If we have a χ_i^{LP*} satisfying $\bar{\chi}_i^j < \chi_i^{LP*} < \bar{\chi}_i^{j+1}$ for some j $(0 \le j \le w-1)$, the constraint $\bar{\chi}_i^j \le \chi \le \bar{\chi}_i^{j+1}$ is added to the formulation. Otherwise if we have a χ_i^{LP*} satisfying $\chi_i^{LP*} = \bar{\chi}_i^j$ for some j $(1 \le j \le w-1)$, the constraint $\bar{\chi}_i^{j-1} \le \chi \le \bar{\chi}_i^{j+1}$ is added.

Then the following linear programming problem (MASTER) is solved.

$$
\begin{array}{rl}
\text{(MASTER):}\min & c^\top x + \sum_{i=1}^{m_2} \theta_i \\
\text{subject to} & Ax = b, x \ge 0 \\
& Tx = \chi \\
& \theta_i \ge \frac{\Psi_i(\chi_i^{LP*})}{\chi_i^{LP*} - \xi_i^{|\Xi_i|}}(\chi_i - \chi_i^{LP*}) + \Psi_i(\chi_i^{LP*}), i = 1, \ldots, m_2 \\
& \text{bound constraints for } \theta_i
\end{array}
$$

Solution algorithm using DSSP

Step1 Given $\varepsilon > 0$ for the convergence check. Solve (SPSIR$_{LP}$) to obtain $(x^{LP*}, \chi^{LP*}, y^{LP*}(\xi^1), \ldots, y^{LP*}(\xi^S))$. The constraint (7) and the lower and upper bounds for θ_i are added to (MASTER). Set $k = 1$.

Step2 Solve (MASTER) to obtain (x^k, χ^k, θ^k).

Table 1: Computational Results

	Number of random variable m_2	Number of scenarios $\|\Xi_i\|$	Parameter r	GAP (%)	Relative error (%)	CPU time (sec) DSSP	Branch-and-Bound
Experiment 1	10	10	25	3.60	0.59	8.00	18.12
	10	20	25	3.74	0.49	10.86	2380.51
Experiment 2	15	10	10	0.76	0.12	8.90	606.82
	15	10	20	2.64	0.57	11.48	8894.02
	15	10	30	4.41	0.49	7.20	21.99
	15	10	40	6.24	0.53	8.11	11.16

Step3 If $k > 1$ and $\sum_{i=1}^{n_1} |x_i^k - x_i^{k-1}| + \sum_{i=1}^{m_2} |\chi_i^k - \chi_i^{k-1}| + \sum_{i=1}^{m_2} |\theta_i^k - \theta_i^{k-1}| > \varepsilon$, modify the constraint (7) and the lower and upper bounds for θ_i of (MASTER), $k = k + 1$, and go to Step 2.

Step4 From the solution (x^k, χ^k, θ^k), calculate $\Psi(\chi^k)$, and set the approximate optimal objective value as $c^\top x^k + \Psi(\chi^k)$.

4 Numerical experiments

4.1 Objective of experiments

In this section, we consider the applications to production planning. It is assumed that the demand for n_2 products are met by existing n_1 production plants.

Suppose the demand of product j is defined as a random variable $\tilde{\xi}_j$. Let $\xi_1, \ldots, \xi_{n_2}$ be the realizations of random variables $\tilde{\xi}_1, \ldots, \tilde{\xi}_{n_2}$, and $\Xi_1, \ldots, \Xi_{n_2}$ be their supports. These random variables are integrated as a random vector $\tilde{\xi} = (\tilde{\xi}_1, \ldots, \tilde{\xi}_{n_2})^\top$, and the support Ξ of $\tilde{\xi}$ is described as $\Xi = \Xi_1 \times \cdots \times \Xi_{n_2}$.

We consider the application of the problem (SPSIR) to the production planning problem(SPSIR'). The first stage decision variable is the amount of products j manufactured by plant i, denoted by x_{ij}, $i = 1, \ldots, n_1, j =$

$1, \ldots, n_2$. Let a_{ij} be the fuel consumption rate of plant i for the production of product j. For the first stage constraints, let b_i be the upper bound for the fuel consumption of the production plant i. The tender variable χ_j is a total amount of product j manufactured by all plants.

Given a first stage decision x and χ, the realization of random demand ξ of $\tilde{\xi}$ becomes known. After observing the realization ξ, the second stage decisions $y_j(\xi_j)$ are taken to meet the demand. The amount of unserved demand has to be supplied by the additional production in the second stage. The multiplication $ry_j(\xi_j)$ of recourse variable $y_j(\xi_j)$ and positive integer r means that the urgent production must be made in r units. The recourse costs q_j are the additional production cost. The formulation of the problem is described as (SPSIR'). The first constraint of the second stage problem to define $\psi_j(\chi_j, \xi_j)$ says the demand must be satisfied, whereas the second constraint of the recourse problem expresses that demand ξ is supplied by the first stage production χ and additional production $ry(\xi)$.

$$
\begin{aligned}
\text{(SPSIR'): min} \quad & \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} c_{ij}x_{ij} + \Psi(\chi) \\
\text{subject to} \quad & \sum_{j=1}^{n_2} a_{ij}x_{ij} \leq b_i, i = 1, \ldots, n_1 \\
& x_{ij} \geq 0, i = 1, \ldots, n_1, j = 1, \ldots, n_2 \\
& \chi_j = \sum_{i=1}^{n_1} x_{ij}, j = 1, \ldots, n_2 \\
& \Psi(\chi) = \sum_{s=1}^{|\Xi_1| \times \cdots \times |\Xi_{n_2}|} p^s \psi(\chi, \xi^s) \\
& \psi(\chi, \xi^s) = \sum_{j=1}^{n_2} \psi_j(\chi_i, \xi_j^s), s = 1, \ldots, (|\Xi_1| \times \cdots \times |\Xi_{n_2}|) \\
& \psi_j(\chi_j, \xi_j^s) = \min\{q_j y_j(\xi_j^s) | r y_j(\xi_j^s) + \chi_j \geq \xi_j^s \\
& \qquad\qquad y_j(\xi_j^s) \in Z_+\}, s = 1, \ldots, |\Xi_j|, j = 1, \ldots, n_2
\end{aligned}
$$

Then two experiments are conducted to show that the algorithm of DSSP is efficient to solve the stochastic programming problem (SPSIR). In experiment 1, the number of scenarios were changed to see the efficiency of the algorithm of DSSP. We show the CPU time of DSSP and the time of Branch-and-Bound. Furthermore, the relative error of DSSP is presented to show the DSSP is precise algorithm.

The results of the numerical experiments appear in Table 1. The GAP described in Table 1, is defined as $(z^* - LB)/LB$, where z^* is an optimal objectve value of (SPSIR) and LB is an optimal objectve value of the LP relaxation of (SPSIR). The relative error in Table 1, is defined as $(\hat{z} - z^*)/z^*$, where $\hat{z}$ is a objectve value obtained using the algorithm of DSSP. The CPU times using DSSP and branch-and-bound or the values of the relative error are compared when the number of scenario is changed.

In experiment 2, the values of the relative error and the CPU time are measured when the value of parameter r is changed. When the value of parameter r becomes large, the length between two adjacent breakpoints becomes long. In this case, it is worthy of notice to see how the value of parameter r affects the precision of DSSP.

The algorithm of DSSP for the stochastic production planning problem was implemented using ILOG OPL Development Studio on DELL DIMENSION 8300 (CPU: Intel Pentium(R)4, 3.20GHz). The simplex optimizer of CPLEX 9.0 was used to solve the problem. Table 1 presents the average values of 5 results of our experiments. The values of the random variables were generated based on the uniform distribution.

4.2 Experiment 1: Changing the number of scenarios

The problems considered in experiment 1, consist of 10 products. The demand for each product has 10 and 20 scenarios. In order to see the efficiency of the algorithm of DSSP, the CPU time of DSSP is compared with the time of Branch-and-Bound. Using the branch-and-bound, the CPU time grows rapidly since we must solve a large scale mixed integer programming problem. However, the algorithm of DSSP solves the problem quickly as the algorithm repeats to solve the linear programming problem. The algorithm of DSSP provides precise solutions as the relative errors of DSSP is less than 1%.

4.3 Experiment 2: Changing the positive integer r

Table 1 shows that the CPU time of tha Branch-and-Bound tends to be long when the value of the parameter r is small. As the length of the range in which the recourse function takes a constant value becomes narrow when r is small, the number of such regions increases. Therefore, the number of times which the lower and upper bounds for θ_i are added, increases. As a result, the CPU time of the Branch-and-bound increases. However, the CPU time of DSSP is shorter than that of the Branch-and-Bound.

The GAP value becomes large when the integer r increases. similarly to the reason described previously, as the length of the range in which the recourse function takes a constant value becomes wide when r is large. Accordingly, the GAP becomes large and the CPU time of branch-and-bound increases. As for the relative errors, it remains within 1%. DSSP provides accurate solutions in short CPU time.

5 Conclusion

We have considered the stochastic programming problem with simple integer recourse in which the value of the recourse variable is restricted to a multiple of a nonnegative integer. The algorithm of a dynamic slope scaling procedure to solve the problem is developed by using the property of the expected recourse function. The numerical experiments show that the proposed algorithm is quite efficient.

References

[1] J.R. Birge, Stochastic programming computation and applications, *INFORMS Journal on Computing*, **9**, (1997), 111–133.

[2] J.R. Birge and F. V. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, 1997.

[3] P. Kall and S.W. Wallace, *Stochastic Programming*, John Wiley & Sons, 1994.

[4] R. Van Slyke and R. J.-B. Wets, L-shaped linear programs with applications to optimal control and stochastic linear programs, *SIAM Journal on Applied Mathematics*, **17**, (1969), 638–663.

[5] G. Laporte and F. V. Louveaux, The integer L-shaped method for stochastic integer programs with complete recourse,*Operations Research Letters*, **13**, (1993), 133–142.

[6] T. Shiina, L-shaped decomposition method for multi-stage stochastic concentrator location problem, *Journal of the Operations Research Society of Japan*, **43**, (2000), 317–332.

[7] T. Shiina, Stochastic programming model for the design of computer network (in Japanese),*Transactions of the Japan Society for Industrial and Applied Mathematics*,**10**, (2000), 37–50.

[8] J.F. Benders, Partitioning procedures for solving mixed variables programming problems, *Numerische Mathematik*, **4**, (1962), 238–252.

[9] S. Ahmed, M. Tawarmalani and N. V. Sahinidis, A finite branch-and-bound algorithm for two-stage stochastic integer programs, *Mathematical Programming*, **100**, (2005), 355–377.

[10] F.V. Louveaux and M.H. van der Vlerk, Stochastic programming with simple integer recourse, *Mathematical Programming*, **61**, (1993), 301–325.

[11] D. Kim and P. Pardalos, A solution approach to fixed charge network flow problem useing a dynamic slope scaling procedure, *Operations Research Letters*, **24**, (1999), 195–203.

[12] D. Kim and P. Pardalos, A dynamic domain constraction algorithm for nonconvex piecewise linear network flow problems, *Journal of Global optimization*, **17**, (2000), 225–234.