International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:6, No:5, 2012

Dynamic Slope Scaling Procedure for Stochastic Integer Programming Problem

Takayuki Shiina

Abstract—Mathematical programming has been applied to various problems. For many actual problems, the assumption that the parameters involved are deterministic known data is often unjustified. In such cases, 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 is a commonly used method for optimization under uncertainty. A stochastic programming problem with recourse is referred to as a two-stage stochastic problem. In this study, we consider a 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 for solving this problem is developed by using a property of the expected recourse function. Numerical experiments demonstrate 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.

Keywords—stochastic programming problem with recourse, simple integer recourse, dynamic slope scaling procedure

I. 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. In such cases, 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 [3], Birge and Louveaux [4], Kall and Wallace [5]) 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 values of random variables are known, a recourse action can be taken in the second stage. For a continuous stochastic programming problem with recourse, the L-shaped method (Van Slyke and Wets [13]) is well known.

The L-shaped method has been used to solve stochastic programs having discrete decisions in the first stage (Laporte and Louveaux [8]) and has been applied to solve a stochastic concentrator location problem (Shiina [11], [12]).

In the present paper, we consider a stochastic programming problem in which the recourse variables are restricted to integers. If integer variables are involved in the second stage problem, optimality cuts based on the Benders [2] decomposition do not provide the facets of the epigraph of the recourse function. In such a case, it is difficult to approximate the recourse function, which in general is 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 [1] 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 multiples of some nonnegative integer. These restrictions on the recourse variables represent that the additional actions are taken in units of a certain amount. 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 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. In Section 4, numerical experiments demonstrate 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.

II. Formulation

A. Stochastic programming problem with recourse

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

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

In the formulation of (SPR), c is a known n_1-vector, b is a known m_1-vector, and A and W are known matrices of size

T. Shiina is with Department of Management Information Science, Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino, Chiba 275-0016, JAPAN

$m_1 \times n_1$ and $m_2 \times n_2$, respectively. The first stage decisions are represented by the n_1-vector x. We assume the random l-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 obtained 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 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).

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

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

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

To solve (DEP-SPR), an L-shaped method (Van Slyke and Wets [13]) has been used. This approach is based on Benders [2] 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 $\theta \geq \sum_{s=1}^{S} p^s Q(x, \xi^s)$.

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

The recourse function is given by an outer linearization using a set of feasibility and optimality cuts as shown in Fig. 1. In the case of $n_2 = 2 \times m_2$ and $W = (I, -I)$, the problem (SPR) is said to have a simple recourse.

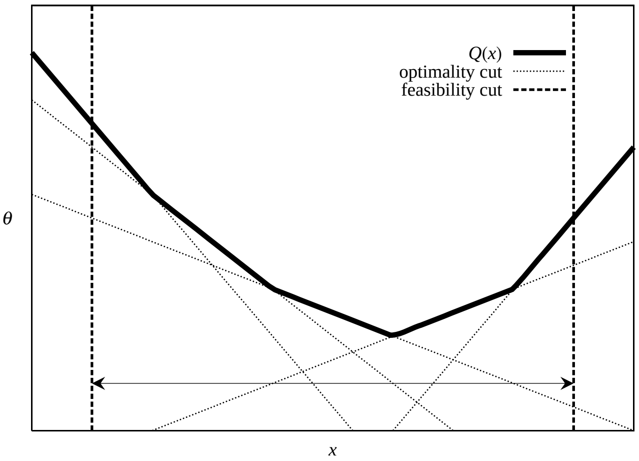

Fig. 1. L-shaped method

B. Simple integer recourse

In this section, we consider the special case of the (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 $y(\xi) \geq \xi - Tx$ and $y(\xi) \geq 0$ to take account 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 that the recourse variables are defined as nonnegative integer variables, the problem is said to have a simple integer recourse. For this problem, the constraints of the recourse problem are $y(\xi) \geq \xi - \chi$, and $y(\xi) \in Z_+^{n_2}$. The optimal solution of the recourse problem is a minimal nonnegative integer variable satisfying $y(\xi) \geq \xi - \chi$.

As the recourse decisions are represented as urgent and additional production, orders, or investment, the recourse decisions are in terms of units of a certain amount. Louveaux-van der Vlerk [9] presented the lower and upper bounds for this problem. But for application of the mathematical programming model to real problems, the recourse decisions should be modified to consider a fixed batch size.

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

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

III. Solution Algorithm

A. 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.

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

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 $\psi_i(\chi_i, \xi_i) = \min\{q_i y(\xi)_i \mid ry(\xi)_i \ge \xi_i - \chi_i, y(\xi)_i \in Z_+\}$ in the following equation.

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

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 $M \ge \max\{\xi_i^s, s = 1, \ldots, |\Xi_i|, i = 1, \ldots, n_2\}$. From assumptions 3.1 and 3.2, the joint probability $P(\tilde{\xi} = \xi^s)$ is calculated as follows.

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

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).

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

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 $\xi_i^1 \le \ldots \le \xi_i^{|\Xi_i|}$ by substituting indices if required. The expectation of the recourse function $\psi_i(\chi_i, \xi_i)$ is shown as follows.

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

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

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

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 $\Psi_i(\chi_i)$ for the case $\Xi = \{11, 22\}, p^1 = p^2 = 1/2, r = 5, q = 1$ is shown in Fig. 2.

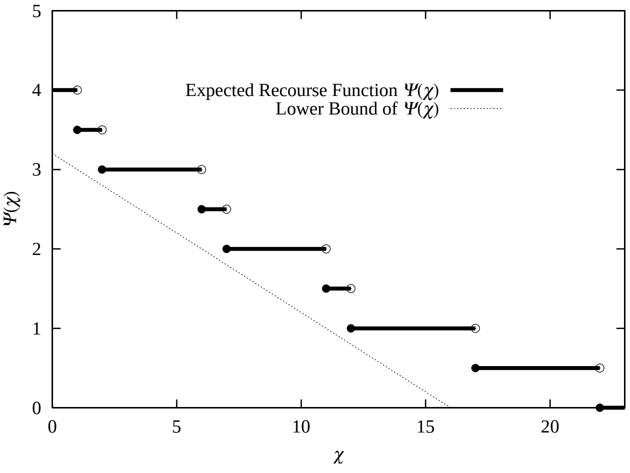

Fig. 2. Expected recourse function

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

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

B. 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.

(SPSIR$_{LP}$):

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

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 [6] developed an approach, called the dynamic slope scaling procedure (DSSP), which solves successive linear programming problems with recursively updated objective functions. Kim and Pardalos [7] 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 Fig. 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).

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

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.

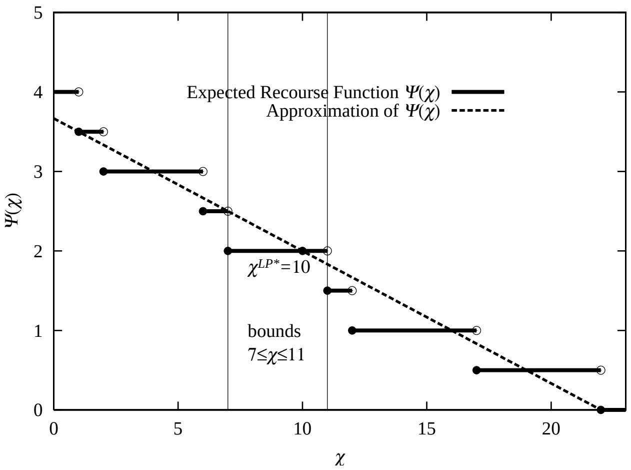

Fig. 3. Algorithm of DSSP

Taking account 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 $0 < \bar{\chi}_i^1 \leq \bar{\chi}_i^2 \leq \ldots \leq \bar{\chi}_i^w$, and define $\bar{\chi}_i^0 = 0$.

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

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

(MASTER):

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

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 $\theta_i$ are added to (MASTER). Set $k = 1$.
Step2 Solve (MASTER) to obtain $(x^k,\chi^k,\theta^k)$.
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 $\theta_i$ of (MASTER), $k = k+1$, and go to Step 2.
Step4 From the solution $(x^k,\chi^k,\theta^k)$, calculate $\Psi(\chi^k)$, and set the approximate optimal objective value as $c^\top x^k + \Psi(\chi^k)$.
```

TABLE I
Computational Results

	Number of random variable	Number of scenarios	Parameter	GAP	Relative error	CPU time (sec)	
	m_2	$\lvert\Xi_i\rvert$	r	(%)	(%)	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

IV. Numerical experiments

A. 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 the 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 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 $\psi_j(\chi_j, \xi_j)$ indicates that 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 $ry(\xi)$.

(SPSIR'):

$$\min \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} c_{ij}x_{ij} + \Psi(\chi)$$

subject to

$$\sum_{j=1}^{n_2} a_{ij}x_{ij} \le b_i, i = 1, \ldots, n_1$$

$$x_{ij} \ge 0, i = 1, \ldots, n_1, j = 1, \ldots, n_2$$

$$\chi_j = \sum_{i=1}^{n_1} x_{ij}, j = 1, \ldots, n_2$$

$$\Psi(\chi) = \sum_{s=1}^{|\Xi_1|\times\cdots\times|\Xi_{n_2}|} p^s \psi(\chi, \xi^s)$$

$$\psi(\chi, \xi^s) = \sum_{j=1}^{n_2} \psi_j(\chi_i, \xi_j^s),$$

$$s = 1, \ldots, (|\Xi_1| \times \cdots \times |\Xi_{n_2}|)$$

$$\begin{aligned} \psi_j(\chi_j, \xi_j^s) = \ & \min\{q_j y_j(\xi_j^s)| \\ & ry_j(\xi_j^s) + \chi_j \ge \xi_j^s \\ & y_j(\xi_j^s) \in Z_+\}, \\ & \qquad s = 1, \ldots, |\Xi_j|, j = 1, \ldots, n_2 \end{aligned}$$

Two experiments were conducted to show that the algorithm of DSSP is efficient for solving the stochastic programming problem (SPSIR). In experiment 1, the number of scenarios were varied to see the efficiency of the algorithm of DSSP. We show herein the CPU times of the DSSP and branch-and-bound algorithms. Furthermore, the relative error of DSSP is presented to show the DSSP is a precise algorithm.

The results of the numerical experiments appear in Table I. GAP, listed in Table I, is defined as $(z^* - LB)/LB$, where z^* is an optimal objective value of (SPSIR) and LB is an optimal objective value of the LP relaxation of (SPSIR). The relative error in Table I is defined as $(\hat{z} - z^*)/z^*$, where $\hat{z}$ is the objective value obtained using the algorithm of DSSP. CPU time and relative error using DSSP and branch-and-bound are compared for different numbers of scenarios.

In experiment 2, the values of the relative error and CPU time are measured as r is varied. As r increases, the distance between two adjacent breakpoints increases. In this case, it is worth noting 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 a Dell Dimension 8300 (CPU: Intel Pentium(R)4, 3.20 GHz). The simplex optimizer of CPLEX 9.0 was used to solve the problem. Table I presents the average values of five results of our experiments. The values of the random variables were generated based on the uniform distribution.

B. Experiment 1: Varying the number of scenarios

The problems considered in experiment 1 consist of 10 products. The demand for each product has 10 or 20 scenarios. In order to see the efficiency of the algorithm of DSSP, the CPU time of DSSP is compared with that of branch-and-bound. Using the branch-and-bound algorithm, the CPU time grows rapidly as r is increased because we must solve a large-scale mixed integer programming problem. However, the algorithm of DSSP solves the problem quickly because the algorithm instead uses repeated solution of linear programming problems. It is clear that algorithm of DSSP provides precise solutions, as the relative errors of DSSP are less than 1%.

C. Experiment 2: Varying the positive integer r

Table I shows that the CPU time of the branch-and-bound algorithm tends to be large when the value of the parameter r is small. Because the length of the range in which the recourse function takes a constant value becomes smaller as r becomes smaller, 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 algorithm increases. However, the CPU time of DSSP is shorter than that of the branch-and-bound algorithm.

The GAP value becomes larger as the integer r increases. For similar reasons to that described previously, the length of the range in which the recourse function takes a constant value increases as r increases. Accordingly, the GAP value increases and the CPU time of branch-and-bound increases. However, the relative errors remain less than 1%. Therefore, DSSP provides accurate solutions in short CPU times.

V. Conclusion

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

References

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

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

[3] J. R. Birge, Stochastic programming computation and applications. *INFORMS Journal on Computing*, Vol. 9, pp. 111–133, 1997.

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

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

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

[7] D. Kim and P. Pardalos, A dynamic domain constraction algorithm for nonconvex piecewise linear network flow problems, *Journal of Global optimization*, Vol. 17, pp.225-234, 2000.

[8] 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.

[9] F. V. Louveaux and M. H. van der Vlerk, Stochastic programming with simple integer recourse, *Mathematical Programming*, Vol. 61, pp. 301-325, 1993.

[10] F. V. Louveaux and R. Schulz, Stochastic integer programming, in *Stochastic Programming* (Handbooks in Operations Research and management Science, A. Ruszczyński and A. Shapiro, eds.), Elsevier, pp. 213-266, 2003.

[11] 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.

[12] 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.

[13] 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.