Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Improved Integer Programming Approaches for Chance-Constrained Stochastic Programming

Hiroki Yanagisawa and Takayuki Osogami
IBM Research - Tokyo
Japan
{yanagis,osogami}@jp.ibm.com

Abstract

The Chance-Constrained Stochastic Programming (CCSP) is one of the models for decision making under uncertainty. In this paper, we consider the special case of the CCSP in which only the right-hand side vector is random with a discrete distribution having a finite support. The unit commitment problem is one of the applications of the special case of the CCSP. Existing methods for exactly solving the CCSP problems require an enumeration of scenarios when they model a CCSP problem using a Mixed Integer Programming (MIP). We show how to reduce the number of scenarios enumerated in the MIP model. In addition, we give another compact MIP formulation to approximately solve the CCSP problems.

1 Introduction

Decision making is one of the important areas in artificial intelligence. Not to mention the frame problem [McCarthy and Hayes, 1969], when we make decisions, it is unrealistic to assume that all of the necessary information is available (e.g., [Bonet and Geffner, 2011]). Therefore, it is important to model the decision making problem as an appropriate model, which depends on the available information. For example, we can model a problem with some random variables as a stochastic programming problem [Heyman and Sobel, 2003] if we know the distributions of the random variables. For another example, we can model a problem with some uncertain values as a robust optimization problem [Ben-Tal *et al.*, 2009] if we know the maximum and minimum values of the uncertain values.

A Chance-Constrained Stochastic Programming (CCSP) problem is an important stochastic programming problem having many applications, where an objective function is minimized under probabilistic constraints. It can be formulated as

$$\begin{aligned} \min \quad & cx \\ \text{subject to} \quad & \Pr(\tilde{A}x \geq \tilde{b}) \geq 1-\epsilon \\ & x \in \mathcal{X}, \end{aligned}$$

where c is a vector, $\tilde{A}$ and $\tilde{b}$ are a random matrix and a vector, respectively, ϵ is a constant parameter between 0 and 1, and $\mathcal{X}$ is the solution space of the decision variable x. The objective of the CCSP is to find $x \in \mathcal{X}$ that minimizes the inner product cx under the probabilistic constraint that the relation $\tilde{A}x \geq \tilde{b}$ holds with probability at least $1-\epsilon$. Applications of CCSP problems include production planning [Lejeune and Ruszczyński, 2007] and the probabilistic lot-sizing problem [Beraldi and Ruszczyński, 2002].

The studies for CCSP have a long history. Charnes, Cooper, and Symonds [1958] first studied the CCSP with disjoint probabilistic constraints, where the probability of violating each constraint is separately evaluated. The CCSP with joint probabilistic constraints was first studied by Miller and Wagner [1965], but the random variables in their formulation are independent. The CCSP with joint probabilistic constraints with dependent random variables was first introduced in Prékopa [1973].

In this paper, we consider a special case of the CCSP, where (i) the random variables appear only in the right-hand side of the constraints and (ii) the random variable $\tilde{b}$ has a discrete distribution having a finite support. Condition (i) means that the probabilistic constraint can be written as $\Pr(Ax \geq \tilde{b}) \geq 1-\epsilon$, where A is a constant matrix. Condition (ii) means that each random variable in $\tilde{b}$ has a set of discrete realizations with probabilities. If we allow approximation, Condition (ii) can be avoided by converting each random variable with an infinite or continuous support into a discrete random variable with a finite number of realizations using sampling methods. Since there are many applications of the special case of the CCSP (such as the unit commitment problem), there exist many papers on this constrained CCSP such as [Luedtke *et al.*, 2010; Küçükyavuz, 2012].

A standard approach to obtain the optimal solution for the CCSP (introduced in [Ruszczyński, 2002]) is to formulate it as a Mixed Integer Programming (MIP), but this requires scenario enumeration. This means that we need to enumerate all of the possible combinations of the realizations of the random variables (i.e., scenarios) to formulate the MIP. If we enumerate all of the scenarios, then the number of scenarios is exponentially large with regard to the number of random variables. Since the size of the MIP problem is proportional to the number of scenarios, the resulting MIP formulation becomes huge. In practice, we cannot exactly solve such a large

MIP formulation.

In this paper, we show how to reduce the number of required scenarios by using a vector domination relation without loss of accuracy. While the standard approach includes all of the scenarios $\mathcal{S}$ in the MIP formulation, our new idea is to construct an MIP formulation that includes only a small subset $\mathcal{S}_\epsilon \subseteq \mathcal{S}$. Although the number of scenarios $|\mathcal{S}_\epsilon|$ in the worst case is still exponential in the number of random variables, the number of scenarios is significantly reduced in practice, which allows us to exactly solve a larger instance of the CCSP.

In addition, we propose an approach to obtain an approximate solution for huge CCSP problems, by using Boole's inequality (also known as the union bound) to construct a compact formulation of the CCSP, avoiding any scenario generation. Similar to the scenario reduction, our new approach is to identify an approximate subset $\mathcal{S}'_\epsilon \subseteq \mathcal{S}_\epsilon$ such that $\mathcal{S}'_\epsilon$ and $\mathcal{S}_\epsilon$ are similar, using Boole's inequality. Then we formulate a new MIP problem using $\mathcal{S}'_\epsilon$. This approach can be seen as decomposing the joint probabilistic constraints in the original CCSP into a problem with individual chance constraints.

We also show that the number of binary variables in the compact CCSP formulation can be further reduced without loss of accuracy. Using technique similar to the ones in [Ibaraki, 1976; Vielma *et al.*, 2010], we reformulate the compact CCSP problem by reducing the number of binary variables logarithmically.

2 The Unit Commitment Problem

The unit commitment problem is one of the important applications of a special case of the CCSP in which only the right-hand side vector is random with a discrete distribution having a finite support. In electric power utility companies, they want to turn off unneeded generators to save cost. However, the generators cannot be switched quickly between the on and off states, and the switching is expensive. Therefore typical electric utilities use predictions of the electricity demands and create a schedule for turning the generators on and off. Ignoring some of the complex constraints arising in real schedules of generators, we can formulate naturally the most essential part of the unit commitment problem as this CCSP problem:

$$\begin{aligned} \min \quad & \sum_i \sum_t c_i x_{it} \\ \text{subject to} \quad & \Pr\left(\sum_i g_i x_{it} \geq \tilde{d}_t \ (\forall t)\right) \geq 1 - \epsilon \\ & x_{it} \in \{0, 1\} \qquad \forall i, t, \end{aligned}$$

where the constant g_i is the maximum power generator i can generate, the constant c_i is the cost for using generator i, and the random variable $\tilde{d}_t$ is the power demand at time t. The decision variable x_{it} is equal to 1 if the generator i is active at time t (that is, it is generating electricity), and otherwise inactive (that is, the generator is turned off). The objective is to minimize the total cost for using these power generators, and the constraint ensures that the sum of the power that can be generated by the generators is at least the power demand at time t, with high probability of $1 - \epsilon$. If we consider the worst case (setting $\epsilon = 0$), then the problem can be solved as standard MIP (taking the maximum realization for each random variable $\tilde{d}_t$). However, the obtained solution is too pessimistic, because it is unrealistic to assume that the random variable $\tilde{d}_t$ for power demand attains its maximum value at every t. Therefore it is natural to model the unit commitment problem as CCSP where the random variables appear only in the right-hand side of the constraints. We can easily incorporate complex constraints arising in real schedules of generators into this CCSP formulation.

The unit commitment problem is an important problem in the energy industry and for sustainability, because variants of this problem are also used in large buildings and factories. The large buildings and factories are typically equipped with heat exchanging systems and private power generators. For economical and environmental reasons, it is critical to create an efficient schedule for the cooling machines in the heat exchanging system as well as for the private power generators to satisfy the cold water and power demands of the buildings and factories.

3 Chance-Constrained Stochastic Programming

In this paper, we consider chance-constrained stochastic programming where (i) only the right-hand side vector $\tilde{b}$ is random and (ii) the random vector $\tilde{b}$ has a discrete distribution having a finite support, which can be written as

$$\begin{aligned} \min \quad & cx & (1) \\ \text{subject to} \quad & \Pr(Ax \geq \tilde{b}) \geq 1 - \epsilon & (2) \\ & x \in \mathcal{X}, & (3) \end{aligned}$$

where c is a vector in $\mathbb{R}^m$, A is an $d \times m$ matrix, $\tilde{b} = (\tilde{b_1}, \ldots, \tilde{b_d})$ is a random vector of values in $\mathbb{R}^d$, ϵ is a parameter between 0 and 1 (typically close to 0), and $\mathcal{X}$ is a subset of $\mathbb{R}^{m_1} \times \mathbb{Z}^{m_2}$ where $m_1 + m_2 = m$. The random variables in $\tilde{b}$ may depend on each other. Constraint (2) is a joint probabilistic constraint that ensures that the inequality $Ax \geq \tilde{b}$ holds with at least probability $1 - \epsilon$. The parameter ϵ determines the tradeoff between the cost and the robustness. That is, a larger ϵ (or equivalently, more robustness) yields a smaller objective function value. We assume that the constraint $x \in \mathcal{X}$ can be written in the form of the constraints of an MIP and that $\mathcal{X}$ is defined by the combination of m_1 continuous variables and m_2 integer variables.

We assume that each random variable in the vector $\tilde{b} = (\tilde{b_1}, \ldots, \tilde{b_d})$ has a finite number of realizations. Specifically, the random variable $\tilde{b_i}$ for $i \in [1, d]$ takes a realization from h_{i1}, h_{i2}, $\ldots$, h_{iK_i} with probability π_{i1}, π_{i2}, $\ldots$ π_{iK_i}, where $\sum_k \pi_{ik} = 1$. In this paper, for simplicity, we assume that $K_i = K$ for all i. Without loss of generality, we assume that every realization of the random variable $\tilde{b_i}$ takes a non-negative value. We refer to a combination of realizations of d random variables as a *scenario*, and let $\mathcal{S}$ be the set of all scenarios. Specifically, we assume that there are $|\mathcal{S}| = K^d$ scenarios $h^1, \ldots, h^{|\mathcal{S}|}$, where

$h^s = (h_1^s, h_2^s, \ldots, h_d^s)$ for $s \in [1, |\mathcal{S}|]$, with probabilities $\pi_1, \pi_2, \ldots, \pi_{|\mathcal{S}|}$, where $\sum_{s\in\mathcal{S}} \pi_s = 1$.

4 Solution Methods

In Section 4.1, we review a straightforward method to exactly solve the CCSP using MIP, and we improve this method by using a scenario reduction technique in Section 4.2. In Section 4.3, we show another compact MIP formulation to approximately solve the CCSP problem.

4.1 The Standard Method

Generating all of the scenarios in $\mathcal{S}$ explicitly, we can rewrite the CCSP problem in the form of an MIP problem without any random variables.

$$\min \quad cx \tag{4}$$

$$\text{subject to} \quad y = Ax \tag{5}$$

$$y \geq h^s(1 - z_s) \quad \forall s \in \mathcal{S} \tag{6}$$

$$\sum_{s\in\mathcal{S}} \pi_s z_s \leq \epsilon \tag{7}$$

$$x \in \mathcal{X} \tag{8}$$

$$y \geq 0 \tag{9}$$

$$z_s \in \{0, 1\} \quad \forall s \in \mathcal{S}. \tag{10}$$

The binary variable z_s is 1 if scenario s does not satisfy $Ax \geq h^s$. Thus Constraint (7) ensures that the total probability of the relation $Ax \geq \tilde{b}$ being violated is bounded by ϵ from above. Conversely, $z_s = 0$ implies that under scenario s we have no violated inequality in $Ax \geq h^s$.

A drawback of this method is the large size of the MIP formulation. Since the number of scenarios $|\mathcal{S}|$ is equal to K^d, even if K is small (e.g., $K = 3$), the number of scenarios is huge (e.g., $|S| = 3^d$) and exponential in d. Since the size of the MIP grows with increases in $|\mathcal{S}|$, it is unrealistic to exactly solve such a huge MIP.

One of the approaches to tackle this problem is to solve the CCSP approximately by using sample-based techniques, also known as Monte Carlo methods [Campi and Garatti, 2011; Pagnoncelli *et al.*, 2009]. These methods sample a small subset $\mathcal{S}'$ from $\mathcal{S}$, and solve the MIP with $\mathcal{S}$ replaced by $\mathcal{S}'$ [Luedtke, 2010]. It is known that we require roughly $O(|\mathcal{S}|/\epsilon)$ samples to yield a highly reliable solution that is feasible and optimal [Luedtke and Ahmed, 2008]. However, it may be computationally prohibitive to solve large problems or to solve problems with high feasibility requirements, because the subset $\mathcal{S}'$ may be still too large to solve the MIP.

4.2 Reducing the Number of Scenarios

We propose a more efficient approach to exactly solve the CCSP by reducing the number of scenarios, using vector dominance relations. Before showing the details of our approach, we define some notation. Given two vectors $h^a = (h_1^a, \ldots, h_d^a)$ and $h^b = (h_1^b, \ldots, h_d^b)$ of length d, the relation $h^a \leq h^b$ means that $h_i^a \leq h_i^b$ holds for every $i \in [1, d]$, whereas the relation $h^a \not\leq h^b$ means that there exists at least one i such that $h_i^a > h_i^b$.

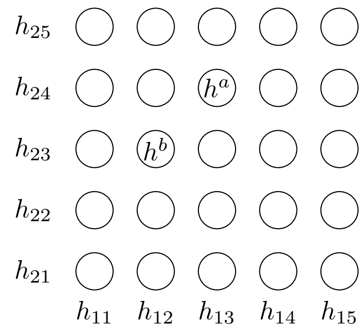

Figure 1: An illustration of the scenarios for two random variables $\tilde{b}_1$ and $\tilde{b}_2$, where random variable $\tilde{b}_1$ has five realizations $h_{11} < h_{12} < h_{13} < h_{14} < h_{15}$ and random variable $\tilde{b}_2$ has five realizations $h_{21} < h_{22} < h_{23} < h_{24} < h_{25}$. Each circle corresponds to a scenario and so scenario $h^a = (h_{13}, h_{24})$ and scenario $h^b = (h_{12}, h_{23})$. Since $h_{12} \leq h_{13}$ and $h_{23} \leq h_{24}$, we see that h^a dominates h^b.

Definition 4.1 *We say that scenario $a \in \mathcal{S}$* dominates *scenario $b \in \mathcal{S}$ if $h^b \leq h^a$ holds. (See Figure 1.)*

Definition 4.2 *We say that a solution x of a CCSP with probabilistic constraint Pr$(Ax \geq \tilde{b}) \geq 1 - \epsilon$ is* feasible for scenario s *if $Ax \geq h^s$ holds.*

Definition 4.3 *Given a set of scenarios $\{1, \ldots, k\} \subseteq \mathcal{S}$, let h be the vector that is obtained from the elementwise max operation for $h^1, \ldots, h^k$. Since the relations $h^i \leq h$ holds for any $i \in [1, k]$, we call such a vector h the* lowest common ancestor.

Now we have Lemma 4.4.

Lemma 4.4 *For any feasible solution x of a CCSP, there exists a scenario s such that (i) x is feasible for any scenario s' that satisfies $h^{s'} \leq h^s$ and (ii) x is infeasible for any scenario s' that satisfies $h^{s'} \not\leq h^s$.*

Proof. Suppose that the solution x is feasible for a CCSP whose probabilistic constraint is $Ax \geq \tilde{b}$. Let $\mathcal{S}_x$ be the set of scenarios that are feasible for x. That is, relation $Ax \geq h$ holds for any scenario $h \in \mathcal{S}_x$ and relation $Ax \not\geq h$ holds for any scenario $h \notin \mathcal{S}_x$.

We show that the lowest common ancestor h^s of the scenarios in $\mathcal{S}_x$ satisfies the conditions (i) and (ii).

Proof for condition (i): We have $Ax \geq h^s$ from the definition of h^s. Hence, for any $h^{s'} \leq h^s$, we have $Ax \geq h^s \geq h^{s'}$.

Proof for condition (ii): Assume that x is feasible for a scenario s' such that $h^{s'} \not\leq h^s$. Scenario s' must be in $\mathcal{S}_x$ and so $h^{s'} \leq h^s$ by the definition of h^s, which contradicts the assumption. Therefore (ii) holds. □

From Lemma 4.4, we know that there exists a scenario s^* that corresponds to an optimal solution x^* of a given CCSP instance. Let $\mathcal{S}^*$ be the set of scenarios that are dominated by s^*. Then the sum of π_s for $s \in S^*$ must be at least $1 - \epsilon$. Therefore, the set of scenarios that are candidates for the scenario s^* is given by

$$\mathcal{S}_\epsilon = \{s \mid \Pr(\tilde{b_1} \leq h_1^s, \ldots, \tilde{b_d} \leq h_d^s) \geq 1 - \epsilon\}.$$

By using the set $\mathcal{S}_\epsilon$, we rewrite the CCSP as

$$\begin{aligned} \min \quad & cx && (11)\\ \text{subject to} \quad & y = Ax && (12)\\ & y \geq h^s z_s \quad \forall s \in \mathcal{S}_\epsilon && (13)\\ & \sum_{s \in \mathcal{S}_\epsilon} z_s = 1 && (14)\\ & x \in \mathcal{X} && (15)\\ & y \geq 0 && (16)\\ & z_s \in \{0,1\} \quad \forall s \in \mathcal{S}_\epsilon, && (17) \end{aligned}$$

where z_s for $s \in \mathcal{S}_\epsilon$ is a binary variable. The binary variable z_s is 1 if scenario s corresponds to an optimal solution x^* of the original CCSP. Constraint (14) ensures that one of the scenarios must correspond to the optimal solution x^*. Conversely, $z_s = 0$ implies that under scenario s we may violate the inequality in the probabilistic constraint ($Ax \ngeq h^s$).

Although the size of $\mathcal{S}_\epsilon$ may be close to that of $\mathcal{S}$ in the worst case, in practice the size of $\mathcal{S}_\epsilon$ will be far smaller than that of the $\mathcal{S}$. Hence the MIP formulation obtained from this approach is far smaller than the formulation given in Section 4.1, and the solution time of the MIP is significantly reduced.

From Constraint (14), we can see that Constraint (13) can be replaced with

$$y \geq \sum_{s \in \mathcal{S}_\epsilon} h^s z_s. \quad (18)$$

Since Constraint (18) is stronger, in practice we should use this instead of Constraint (13).

4.3 Approximate Solution

As we mentioned in Section 4.2, the number of scenarios $\mathcal{S}_\epsilon$ can still be huge (which means exponential in the number of random variables d, on the order of K^d). Therefore we consider another approach that approximately solves a CCSP problem. Our new idea is to construct another set $\mathcal{S}'_\epsilon \subseteq \mathcal{S}_\epsilon$ that can be expressed with a polynomial number of constraints.

Before showing how to construct such a set $\mathcal{S}'_\epsilon$, let us define some notation. We say that a random variable $\tilde{b_i}$ *touches* an undesirable event for scenario s if a realization h_{ik} of $\tilde{b_i}$ satisfies the relation $h_i^s \leq h_{ik}$. Let E_i^s be the event in which the random variable $\tilde{b_i}$ touches the undesirable event for scenario s. Then, by definition of $\mathcal{S}_\epsilon$, scenario $s \in \mathcal{S}_\epsilon$ if and only if

$$\Pr(\bigcup_{i=1}^{d} E_i^s) \leq \epsilon.$$

Now let us review Boole's inequality (for example, see page 194 in [Comtet, 1974]).

Theorem 4.5

$$\Pr(\bigcup_{i=1}^{d} E_i^s) \leq \sum_{i=1}^{d} \Pr(E_i^s)$$

holds for any scenario s. Note that Boole's inequality holds even if the E_i^ss are dependent. This relation holds with equality when $\Pr(E_i^s \cap E_j^s) = 0$ holds for every distinct pair of i and j.

Now we define a set of scenarios $\mathcal{S}'_\epsilon$ such that scenario $s \in \mathcal{S}'_\epsilon$ if and only if

$$\sum_{i=1}^{d} \Pr(E_i^s) \leq \epsilon.$$

By using Boole's inequality, it is easy to verify that $\mathcal{S}'_\epsilon \subseteq \mathcal{S}_\epsilon$.

To formulate an MIP with the scenario set $\mathcal{S}'_\epsilon$, we transform the following inequality

$$\Pr(E_i^s) \leq \epsilon_i, \quad (19)$$

where ϵ_i is a constant, into the one having the form suitable for an MIP. To do so, for a fixed $i \in [1, d]$, we introduce K binary variables $z_{ik} \in \{0, 1\}$ for $k \in [1, K]$. Then an equivalent form of Inequality (19) is given as

$$\begin{aligned} \sum_{k \in [1,K]} \pi_{ik} z_{ik} \quad & \leq \quad \epsilon_i \\ z_{ik} \quad & \geq \quad z_{i(k+1)} \; \forall k \in [1, K-1] \end{aligned}$$

because $z_{i1} = \cdots = z_{ip} = 1$ and $z_{i(p+1)} = \cdots = z_{iK} = 0$ holds if and only if scenario s takes a realization such that $h_i^s = h_{ip}$.

Now we show our approximate MIP model. Whereas the original CCSP problem can be interpreted as finding a scenario $s^* \in \mathcal{S}_\epsilon$ that corresponds to an optimum solution x^* of the CCSP, we formulate the approximate MIP problem as finding a scenario $s \in \mathcal{S}'_\epsilon$ whose corresponding feasible solution x minimizes the objective value cx. This relation can be modeled as this MIP:

$$\begin{aligned} \min \quad & cx && (20)\\ & y = Ax && (21)\\ & y_i \geq h_{ik}(1 - z_{ik}) \quad \forall i \in [1, d], \forall k \in [1, K] && (22)\\ & \sum_i \epsilon_i = \epsilon && (23)\\ & \sum_k \pi_{ik} z_{ik} \leq \epsilon_i \quad \forall i \in [1, d] && (24)\\ & z_{ik} \geq z_{i(k+1)} \quad \forall i \in [1, d], \forall k \in [1, K] && (25)\\ & x \in \mathcal{X} && (26)\\ & y \geq 0 && (27)\\ & z_{ik} \in \{0, 1\} \quad \forall i \in [1, d], \forall k \in [1, K] && (28)\\ & \epsilon_i \geq 0 \quad \forall i \in [1, d], && (29) \end{aligned}$$

where y_i is the i-th element of vector y and ϵ_i is a continuous variable. Recalling that the combination of Constraints (24) and (25) is equivalent to Constraint (19), it is easy to verify that this MIP model is created as desired. Note that, since Constraint (25) is redundant, we can remove it when we implement this model in practice.

Our approach can be interpreted as dividing the original CCSP with joint probabilistic constraints into a combination of the individual probabilistic constraints. More specifically, Constraint (24) can be interpreted as bounding by ϵ_i from above the probability that random variable $\tilde{b_i}$ in $\tilde{b}$ violates the relation $Ax \geq \tilde{b}$. Therefore Constraint (23) can be interpreted as a combination of d individual probabilistic constraints, and their total violation probability is bounded by ϵ from above. In other words, we can interpret the joint probabilistic Constraint (7) as divided into a set of individual probabilistic Constraints (24), each of which is parameterized by ϵ_i, using Boole's inequality.

5 Discussion

In this section, we discuss some existing techniques applicable to our approaches.

5.1 Mixing Set Approach

Here we show that we can accelerate solving the approximate CCSP by using the cutting plane method. For the cutting plane method, we exploit an existing technique on the *mixing set* to generate valid inequalities.

Constraint (22) is the mixing set arising in our approximate CCSP:

$$Q = \{(y, z) \in \mathbb{R}_+^d \times \{0,1\}^K \mid y + h_k z_k \geq h_k, k \in [1, K]\},$$

where we omit subscript i for simplicity. A stronger (but equivalent) form of this mixing set can be written as

$$\begin{aligned} Q' \;\; = \;\; & \{(y, z) \in \mathbb{R}_+^d \times \{0,1\}^K \mid \\ & y + (h_k - h_{v+1}) z_k \geq h_k, k \in [1, v]\}, \end{aligned}$$

for v such that $\sum_{k=1}^{v} \pi_k \leq 1 - \epsilon$ and $\sum_{k=1}^{v+1} \pi_k > 1 - \epsilon$. Günlük and Pochet [2001] studied this mixing set, and obtained many valid inequalities for this mixing set as in Theorem 5.1.

Theorem 5.1 *For $R = \{r_1, r_2, \ldots, r_a\} \subseteq \{1, \ldots, v\}$, the inequalities*

$$y + \sum_{j=1}^{a} (h_{r_j} - h_{r_{j+1}}) z_{r_j} \geq h_{r_1},$$

where $r_1 < r_2 < \cdots < r_a$ and $h_{r_{a+1}} = h_{v+1}$, are valid for Q and are facet-defining for $conv(Q)$ when $r_1 = 1$.

We can use this result to generate valid inequalities when we solve the CCSP by using an MIP solver (such as IBM ILOG CPLEX) with the cutting plane method. Since the number of inequalities generatable from this theorem are exponential in v_i, we cannot generate all of the inequalities in advance, but Luedtke *et al.* [2010] showed how to generate appropriate inequalities to accelerate solving the MIP in the cutting plane method.

5.2 Logarithmic Number of Binary Variables Approach

Here we reformulate the MIP formulation in Section 4.3 so that the number of binary variables in the formulation is reduced, using an approach similar to the formulation for a piecewise-linear function with a logarithmic number of binary variables [Ibaraki, 1976; Vielma *et al.*, 2010].

Recall that Constraint (22) can be written as the mixing set Q and the inequalities $z_1 \geq z_2 \geq \cdots \geq z_v$ always hold in Q. Then we can express Q by using these inequalities

$$\begin{aligned} y + h_k z_k \geq h_k & \qquad \forall k \\ z_{k-1} \geq \lambda_k & \qquad \forall k \in [2, v] \\ z_k \leq 1 - \lambda_k & \qquad \forall k \\ \sum_k \lambda_k = 1 & \\ \sum_{k:l\text{-th digit is } 1} \lambda_k \leq \alpha_l & \qquad \forall l \in [1, \lceil \log v \rceil] \\ \sum_{k:l\text{-th digit is } 0} \lambda_k \leq 1 - \alpha_l & \qquad \forall l \in [1, \lceil \log v \rceil] \\ \lambda_k \geq 0 & \qquad \forall k \in [0, K-1] \\ \alpha_l \in \{0, 1\} & \qquad \forall l \in [1, \lceil \log v \rceil] \\ z_k \geq 0 & \qquad \forall k \in [0, K-1]. \end{aligned}$$

Notice that each z_k is *not* a binary variable but a continuous variable, and that the number of the binary variables is the logarithm of v ($\leq K$). In these inequalities, each variable α_l takes a binary value. Hence only one λ_k can be 1, and the other λ_ks must be 0. It thus turns out that each z_k has a binary value, and the relation $z_1 \geq z_2 \geq \cdots \geq z_v$ always holds.

Note that, if $\lambda_1 = 1$, then all of the z_ks are 0. If $\lambda_v = 1$, then $z_v = 0$. Recall that, if $z_v = 1$, then all of the z_ks are 1 and so the chance-constraint of the corresponding random variable cannot be satisfied with probability 1, which yields an infeasible solution as long as $\epsilon < 1$.

The technique presented here is to give theoretical improvement of reducing the number of binary variables. Practical performance would depend on various factors such as the distributions of the random variables in CCSP.

6 Computational Experiments

In this section, we show some experimental results with our new MIP formulations. We compared three approaches:

ALL: the MIP approach that generates all of the scenarios in $\mathcal{S}$ as described in Section 4.1,

SMALL: the MIP approach that generates all of the scenarios in $\mathcal{S}_\epsilon$ as described in Section 4.2, where we use the tighter inequality (Constraint (18)), and

APPROX: the compact MIP approach with approximation as described in Section 4.3 using the cutting-plane method mentioned in Section 5.1.

All of the experiments were done on a personal computer, having an Intel Core i5 (M560) running at 2.67 GHz, with 3 GB of RAM using Microsoft Windows XP Professional Service Pack 3. All of the programs were written in Java and compiled with JDK 1.6. We used IBM ILOG CPLEX 12.3 as the MIP solver. All of the execution times are expressed in seconds.

To evaluate our new MIP formulations, we considered a transportation problem similar to [Luedtke *et al.*, 2010]. In this problem, there are a set of suppliers I and a set of customers D. Each supplier $i \in I$ has a constant capacity M_i

	Supplies	Demands	Realizations
Instance 2	30	2	50
Instance 3	30	3	50
Instance 4	30	4	50
Instance 10	30	10	50
Instance 11	30	11	50
Instance 12	30	12	50
Instance 13	30	13	50
Instance 14	30	14	50

Table 1: Instances generated for our experiments.

	ALL		SMALL	
Instance 2	0.2	(2500)	0.1	(6)
Instance 3	184.3	(125 000)	0.1	(10)
Instance 4	> 500	(6 250 000)	0.1	(15)

Table 2: Computation times for ALL and SMALL.

and each customer $j \in D$ has a demand, which is a random variable $\tilde{d}_j$. There is a transportation cost c_{ij} (for example, fuel usage for a truck) for shipping each unit of product from supplier $i \in I$ to customer $j \in D$. We want to ship the products such that

$$\sum_{j \in D} x_{ij} \leq M_i \ \forall i \in I$$

and

$$\Pr\left(\sum_{i \in I} x_{ij} \geq \tilde{d}_j \ (\forall j \in D)\right) \geq 1 - \epsilon,$$

where x_{ij} is the amount of product shipped from supplier i to customer j and ϵ is a constant parameter. The objective is to minimize the total shipping cost, which is expressed as

$$\min \sum_i \sum_j c_{ij} x_{ij}.$$

Table 1 summarizes the instances we generated for the transportation problem. The capacity M_i of each supplier $i \in I$ is fixed to 1000. The demand $\tilde{d}_j$ of each customer $j \in D$ is a discrete distribution having a finite support generated by sampling from the Gaussian distribution with mean 1000 and variance 10. The column 'Realizations' specifies the number of sampling for each demand. The parameter ϵ is set to 0.05.

Table 2 shows computational results for two approaches SMALL and ALL, where each column shows the computation times and each number in parenthesis indicates the number of scenarios generated. These results show that SMALL is clearly faster than ALL for all instances. This is because, whereas ALL generates a large number of scenarios, SMALL generates a very smaller number of scenarios.

Table 3 shows computational results for two approaches SMALL and APPROX, where the fourth column shows the *approximation ratio*: the objective value of the APPROX divided by the optimal objective value. The number in the parenthesis in the second column shows the number of the scenarios generated in SMALL, and the number in the parenthesis in the third column shows the number of the variables z_{ik} generated in APPROX. These results show that both

	SMALL		APPROX		Ratio
Instance 10	1.2	(66)	0.1	(500)	1.0
Instance 11	4.6	(78)	0.1	(550)	1.0
Instance 12	17.3	(91)	0.1	(600)	1.0
Instance 13	73.1	(105)	0.1	(650)	1.0
Instance 14	286.0	(120)	0.1	(700)	1.0

Table 3: Computation times and approximation ratios for SMALL and APPROX.

SMALL and APPROX generate very small number of scenarios and the variables z_{ik}, and that APPROX is clearly faster than SMALL. This difference comes from the fact that it takes long time to generate the scenarios $\mathcal{S}_\epsilon$ in SMALL (actually, the computation time for solving MIP is less than 0.1 second for every input). Therefore an efficient algorithm for generating $\mathcal{S}_\epsilon$ is required. Finally we see that APPROX exactly solves the instances we used in these experiments (i.e., the approximation ratio is 1.0). We expect that the approximation ratio would become worse (increase) for the instances having more constraints and variables.

7 Related Work

The CCSP in which the right-hand side vector is random with a discrete distribution having a finite support has been studied extensively. Ruszczyński [2002] formulated the CCSP as an MIP, and some new cutting-plane methods were proposed to solve CCSP problems exactly under the assumption that all of the scenarios happen with equal probability [Luedtke *et al.*, 2010; Küçükyavuz, 2012]. Küçükyavuz [2012] further gives a polynomial-size CCSP formulation under the same assumption.

For general CCSP problems with a discrete distribution having a finite support, Luedtke [2010] gave an algorithm to solve it, extending their algorithm for the CCSP in which the right-hand side vector is random [Luedtke *et al.*, 2010].

8 Conclusion

We showed new approaches to solve the CCSP problems in which only the right-hand side vector is random with a discrete distribution having a finite support, and we showed that these approaches can solve larger problems than the existing approaches. The unit commitment problem is one of the important applications of the CCSP problem, and it is also important for energy savings and sustainability.

An apparent future work is to extend our approaches to general CCSP problems. Another future work is to investigate the tradeoff between the approximation ratio and the computation time of the compact MIP approach. It would also be interesting to investigate whether generalized variants of Boole's inequality (i.e., Bonferroni's inequalities) can be used instead of Boole's inequality.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments.

References

[Ben-Tal *et al.*, 2009] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[Beraldi and Ruszczyński, 2002] Patrizia Beraldi and Andrzej Ruszczyński. The probabilistic set-covering problem. *Operations Research*, 50(6):956–967, 2002.

[Bonet and Geffner, 2011] Blai Bonet and Hector Geffner. Planning under partial observability by classical replanning: Theory and experiments. In *Proceedings of IJCAI 2011*, pages 1936–1941, 2011.

[Campi and Garatti, 2011] Marco C. Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[Charnes *et al.*, 1958] Abraham Charnes, William W. Cooper, and Gifford H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[Comtet, 1974] Louis Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Springer, 1974.

[Günlük and Pochet, 2001] Oktay Günlük and Yves Pochet. Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

[Heyman and Sobel, 2003] Daniel P. Heyman and Matthew J. Sobel. *Stochastic Models in Operations Research, Vol. II: Stochastic Optimization*. Dover Publications, December 2003.

[Ibaraki, 1976] Toshihide Ibaraki. Integer programming formulation of combinatorial optimization problems. *Discrete Mathematics*, 16:39–52, 1976.

[Küçükyavuz, 2012] Simge Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1–2):31–56, 2012.

[Lejeune and Ruszczyński, 2007] Miguel A. Lejeune and Andrzej Ruszczyński. An efficient trajectory method for probabilistic production-inventory-distribution problems. *Operations Research*, 55(2):378–394, 2007.

[Luedtke and Ahmed, 2008] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[Luedtke *et al.*, 2010] James Luedtke, Shabbir Ahmed, and George L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[Luedtke, 2010] James Luedtke. An integer programming and decomposition approach to general chance-constrained mathematical programs. In *Proceddings of IPCO 2010*, pages 271–284, 2010.

[McCarthy and Hayes, 1969] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969. reprinted in McC90.

[Miller and Wagner, 1965] Bruce L. Miller and Harvey M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13:930–965, 1965.

[Pagnoncelli *et al.*, 2009] B. K. Pagnoncelli, Shabbir Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009.

[Prékopa, 1973] Andras Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4:202–221, 1973.

[Ruszczyński, 2002] Andrzej Ruszczyński. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming*, 93:195–215, 2002.

[Vielma *et al.*, 2010] Juan Pablo Vielma, Shabbir Ahmed, and George Nemhauser. Mixed-integer models for non-separable piecewise linear optimization: unifying framework and extensions. *Operations Research*, 58(2):303–315, 2010.