A Support Vector Machine-based Approach to Chance Constrained Problems using Huge Data Sets

Kiyoharu Tagawa

1Department of Informatics, School of Science and Engineering, Kindai University
3-4-1 Kowakae, Higashi-Osaka, Osaka 577-8502, Japan
E-mail: tagawa@info.kindai.ac.jp

Abstract

In this paper, a new approach to solve data-driven Chance Constrained Problems (CCPs) is proposed. First of all, a large data set is used to formulate CCP because such a large data set is available nowadays due to advanced information technologies. However, since the size of the data set is too large, a Support Vector Machine (SVM) is used to estimate the probability of meeting all constraints of CCP for the large data set. In order to generate a training data set for the SVM, a sampling technique called Space Stratified Sampling (SSS) is proposed in this paper. According to the first principal component obtained by Principal Component Analysis (PCA), SSS divides the large data into several strata and selects some data from each stratum. The SVM trained by SSS is called S\textsubscript{SVM}. In order to solve CCPs based on large data sets efficiently, a new optimization method called Adaptive Differential Evolution with Pruning technique (ADEP) is also proposed.

1 Introduction

Chance Constrained Problem (CCP) [1], which is also referred to as probabilistic constrained problem [2], is one of the possible formulations of the stochastic optimization problem. CCP is a risk-averse formulation of problem under uncertainties. Specifically, CCP ensures that the probability of meeting all constraints is above a certain level. Since the balance between optimality and reliability can be designated by CCP, many real-world applications have been formulated as CCPs [2, 3, 4].

CCP has been studied in the field of the stochastic programming for many years [1]. In the stochastic programming, the optimization methods of the nonlinear programming [5] have been used to solve CCP. Recently, Evolutionary Algorithms (EAs) have been also reported for solving CCPs [6, 7, 8]. However, in the conventional formulation of CCP, a well-known probability distribution such as the normal distribution is used widely as a mathematical model of unknown uncertainties. Specifically, the mathematical model is used to generate pseudo data randomly with the Monte Carlo method [9]. Otherwise, the mathematical model is used to derive a deterministic formulation of CCP [1, 6, 10]. In some cases, very few data observed actually, which are called scenarios, are used as uncertainties in the formulation of CCP. As a drawback of the conventional formulation of CCP, the estimation error of uncertainties is unavoidable in the evaluation of solutions. If CCP is defined incompletely by using a loose mathematical model, the solution of CCP is also defective. Consequently, we cannot enjoy the benefit of CCP.

In recent years, due to advanced information technologies such as Wireless Sensor Networks (WSN) and Internet of Things (IoT) [11], huge data sets called “big data” can be easily obtained in various fields including culture, science, and industry [12]. In many real-world applications, the variance of observed data is caused by some uncertainties. Therefore, applications might be formulated as CCP more accurately by using such a huge data set instead of the mathematical model.

In this paper, a new approach to solve data-driven CCPs is proposed. First of all, a large data set is used to formulate CCP [13, 14]. However, we suppose that the data set is too large to evaluate the probability of meeting all constraints of CCP. Therefore, in order to estimate the probability from the large data set efficiently, a Support Vector Machine (SVM) is used. SVMs are supervised learning models with associated learning algorithms that analyze data for classification [15, 16]. SVMs have some advantages. Unlike in neural networks, SVM is not solved for local optima. Besides, it scales relatively well to high dimensional data.

In this paper, in order to generate a training data set for the above SVM, a new sampling technique called Space Stratified Sampling (SSS) is proposed. According to the first principal component obtained by Principal Component Analysis (PCA), SSS divides the large data into several strata and selects some data from each stratum. The performance of SSS is compared with a well-known sampling technique, or Simple Random Sampling (SRS) [16, 17]. SVM trained by SSS is called S\textsubscript{SVM}, while SVM trained by SRS is called R\textsubscript{SVM}. Experimental results indicate that S\textsubscript{SVM} estimates the true probability more accurately than R\textsubscript{SVM}, especially if the true probability value is large.

For solving CCP based on a large data set efficiently, a new Adaptive Differential Evolution with Pruning technique (ADEP) is also proposed. In the proposed ADEP, an elaborate pruning technique is employed for reducing the number of candidate solutions to be examined.
completely on the process of optimization [14].

The remainder of this paper is organized as follows. Section 2 formulates CCP by using a large data set. Section 3 explains two types of SVMs, namely R_SVM and S_SVM. Section 4 explains ADEP combined with SVM for solving CCP. Section 5 compares S_SVM with R_SVM through numerical experiments. In Section 6, the proposed approach is applied to a real-world application, or the flood control planning formulated as CCP. Section 7 gives conclusions and future work.

2 Problem Formulation

Symbols used in this paper are defined as follows:

- Decision variables \( x = (x_1, \cdots, x_D) \in X \subseteq \mathbb{R}^D \)
- Random variables \( \xi = (\xi_1, \cdots, \xi_K) \in \Omega \subseteq \mathbb{R}^K \)
- A large data set \( \xi^i \in B \subseteq \Omega \)
- Index set \( [N] = \{1, \cdots, n, \cdots, N\} \)
- A set of samples \( \xi^n \in \Phi \subseteq B, \ n \in [N] \)
- Measurable functions \( g_m : X \times \Omega \rightarrow \mathbb{R}, \ m \in [M] \)
- Objective function to be minimized \( f : X \rightarrow \mathbb{R} \)
- Sufficiency level given by a probability \( \alpha \in (0, 1) \)

2.1 Chance Constrained Problem (CCP)

Let \( \Pr(A) \) be the probability that an event \( A \) occurs. The probability of meeting all probabilistic constraints \( g_m(x, \xi) \leq 0, \ m \in [M] \) for a solution \( x \in X \) is

\[
p(x, \Omega) = \Pr(\forall \xi \in \Omega : g_m(x, \xi) \leq 0, \ m \in [M]).
\]

(1)

By using \( p(x, \Omega) \) in (1), CCP is formulated as

\[
\min_{x \in X} f(x) \text{ s.t. } p(x, \Omega) \geq \alpha
\]

(2)

where \( \alpha \in (0, 1) \) is given by an arbitrary probability.

In real-world applications, both the sample space \( \Omega \)
and the distribution of \( \xi \in \Omega \) are usually unknown.
Thus, it is impossible to solve CCP in (2) directly.

2.2 Formulation of Data-Driven CCP

As stated above, due to advanced information technologies such as WSN and IoT, we can suppose that a large data set \( B \subseteq \Omega \) is available for estimating the unknown probability \( p(x, \Omega) \) in (2) empirically.

First of all, the indicator function is defined as

\[
\mathbb{I}(\xi^i | x) = \begin{cases} 
1 & \text{if } g_m(x, \xi^i) \leq 0, \ m \in [M] \\
0 & \text{otherwise.}
\end{cases}
\]

(3)

By using the indicator function in (3), the unknown probability \( p(x, \Omega) \) in (1) is evaluated empirically from the large data set \( \xi^i \in B \) for a solution \( x \in X \) as

\[
\hat{p}(x, B) = \frac{1}{|B|} \sum_{\xi^i \in B} \mathbb{I}(\xi^i | x)
\]

(4)

where \( |B| \) denotes the size of the data set \( B \).

From the law of large numbers [18], we can expect that \( p(x, \Omega) \simeq \hat{p}(x, B) \) holds. Therefore, by using the probability in (4), CCP in (2) can be rewritten as

\[
\min_{x \in X} f(x) \text{ s.t. } \hat{p}(x, B) \geq \alpha.
\]

(5)

It is also hard to solve CCP in (5). In many real-world applications, the function value \( g_m(x, \xi^i) \) in (3) has to be evaluated for each of the data \( \xi^i \in B \) through a time-consuming computer simulation. Therefore, the number of data \( \xi^i \in B \) is too large to evaluate the empirical probability in (4). For solving CCP in (5) practically, we have to neglect the most of data [14].

3 Estimation of Probability

3.1 Probability Estimation by SVM

A SVM is used to evaluate the probability \( \hat{p}(x, B) \) in (4) from the large data set \( B \) for a solution \( x \in X \). First of all, as a training data set for the SVM, some samples \( \xi_n \in \Phi, \ n = 1, \cdots, N \) are selected from \( B \).

The technique of selecting \( \Phi \subseteq B \) will be detailed later. Then, each \( \xi_n \in \Phi \) is given a label \( y_n \in \{-1, 1\} \) as

\[
y_n(\xi_n | x) = \begin{cases} 
1 & \text{if } g_m(x, \xi_n) \leq 0, \ m \in [M] \\
-1 & \text{otherwise.}
\end{cases}
\]

(6)

Then, \( \xi_n \in \Phi \) and \( y_n = y_n(\xi_n | x), \ n = 1, \cdots, N \) are used to train the above SVM. Since functions \( g_m(x, \xi^i) \) in (6) are not always linear, Radial Basis Function (RBF) [15] is assigned for the kernel function of SVM to learn a nonlinear classification rule. As a result, the decision function of SVM is obtained as

\[
d(\xi | x, \Phi) = \sum_{\xi_n \in \Phi} a_n y_n \mathcal{K}(\xi_n, \xi) + b
\]

(7)

where \( \Phi \subseteq B \) is the sample set. \( \mathcal{K}(\xi_n, \xi) \) is the kernel function. Coefficients \( b \in \mathbb{R} \) and \( a_n \in \mathbb{R}, \ n = 1, \cdots, N \) are decided by using a quadratic programming [15].

The SVM is used to predict the indicator function value in (3) for \( \xi^i \in B \), namely \( \mathbb{I}(\xi^i | x) \in \{0, 1\} \), by using the decision function in (7) as

\[
\mathbb{I}(\xi^i | x, \Phi) = \begin{cases} 
1 & \text{if } d(\xi^i | x, \Phi) > 0 \\
0 & \text{if } d(\xi^i | x, \Phi) < 0.
\end{cases}
\]

(8)

By using \( \mathbb{I}(\xi^i | x, \Phi) \in \{0, 1\} \) in (8) given by the SVM, the probability in (4) is estimated as

\[
\hat{p}(x, \Phi, B) = \frac{1}{|B|} \sum_{\xi^i \in B} \mathbb{I}(\xi^i | x, \Phi).
\]

(9)
4 Optimization Method

Differential Evolution (DE) [19] has been proven to be one of the most powerful global optimization algorithms. However, the original DE [19] is only applicable to unconstrained problems. Besides, the performance of DE is significantly influenced by its control parameter settings [20]. Therefore, in order to solve CCP in (10) efficiently, a new optimization method called Adaptive DE with Pruning technique (ADEP) is proposed. In the proposed ADEP, three techniques are integrated into the original DE: 1) Adaptive control of parameters [21]; 2) Constraint handling based on feasibility rule [22]; and 3) Pruning technique in selection [14].

4.1 Constraint Handling with Pruning

ADEP has a set of candidate solutions \( x_i \in P_t \), \( i = 1, \cdots, N_P \) called population in each generation \( t \). Each candidate solution \( x_i \in P_t \) is a vector of decision variables. The initial population \( x_i \in P_0 \subseteq X \) is randomly generated according to a uniform distribution. For each candidate solution \( x_i \in P_t \) of CCP in (10), the amount of constraint violation is defined as

\[
\eta(x_i) = \max\{\alpha - \hat{p}(x_i, \Phi, B), 0\}
\]

(13)

where \( x_i \in P_t \) is feasible if \( \eta(x_i) = 0 \) holds.

At each generation \( t \), a trial vector \( z_i \in X \) is generated from an existing target vector \( x_i \in P_t \). Either the trial vector \( z_i \in X \) or the target vector \( x_i \in P_t \) is selected for a new vector \( x_i \in P_{t+1} \) of the next generation. First of all, if the following condition is satisfied,

\[
(\eta(x_i) = 0) \land (f(x_i) < f(z_i))
\]

(14)

the trial vector \( z_i \in X \) is discarded immediately because \( x_i \in P_t \) is better than \( z_i \in X \). Then, the target vector \( x_i \in P_t \) survives to the next generation. Since the pruning technique based on the condition in (14) does not require the evaluation of \( \eta(z_i) \) in (13), it is very effective to reduce the run time of ADEP.

Only when the condition in (14) is not satisfied, the constraint violation \( \eta(z_i) \) defined by (13) is evaluated for \( z_i \in X \) by using the SVM for \( \hat{p}(z_i, \Phi, B) \). Then, if either of the following conditions is satisfied,

\[
\begin{align*}
\eta(z_i) < \eta(x_i) \\
(\eta(z_i) = \eta(x_i)) \land (f(z_i) \leq f(x_i))
\end{align*}
\]

(15)

the newborn trial vector \( z_i \in X \) is selected for a new vector \( x_i \in P_{t+1} \) of the next generation. Otherwise, the current target vector \( x_i \in P_t \) survives to the next generation and becomes the new vector \( x_i \in P_{t+1} \).

4.2 Proposed Algorithm of ADEP

The algorithm of ADEP is described as follows. The maximum number of generations \( N_P \) is given as the termination condition. The population size is chosen as \( N_P = 5D \) [19]. Other parameters of ADEP are

We expect that \( \hat{p}(x, \Phi, B) \approx \hat{p}(x, B) \) holds. Then, in order to obtain a solution of CCP in (5), we solve the following CCP defined by the probability in (9).

\[
\min_{x \in X} f(x) \quad \text{s.t.} \quad \hat{p}(x, \Phi, B) \geq \alpha
\]

(10)

\[\text{Step 3:} \quad \text{For each candidate solution } x_i \in P_t \text{ of CCP in (10), the amount of constraint violation is defined as}\]

\[
\eta(x_i) = \max\{\alpha - \hat{p}(x_i, \Phi, B), 0\}
\]

(13)

\[\text{where } x_i \in P_t \text{ is feasible if } \eta(x_i) = 0 \text{ holds.}\]

\[\text{At each generation } t, \text{ a trial vector } z_i \in X \text{ is generated from an existing target vector } x_i \in P_t. \text{ Either the trial vector } z_i \in X \text{ or the target vector } x_i \in P_t \text{ is selected for a new vector } x_i \in P_{t+1} \text{ of the next generation. First of all, if the following condition is satisfied,}\]

\[
(\eta(x_i) = 0) \land (f(x_i) < f(z_i))
\]

(14)

\[\text{the trial vector } z_i \in X \text{ is discarded immediately because } x_i \in P_t \text{ is better than } z_i \in X. \text{ Then, the target vector } x_i \in P_t \text{ survives to the next generation. Since the pruning technique based on the condition in (14) does not require the evaluation of } \eta(z_i) \text{ in (13), it is very effective to reduce the run time of ADEP.}\]

\[\text{Only when the condition in (14) is not satisfied, the constraint violation } \eta(z_i) \text{ defined by (13) is evaluated for } z_i \in X \text{ by using the SVM for } \hat{p}(z_i, \Phi, B). \text{ Then, if either of the following conditions is satisfied,}\]

\[
\begin{align*}
\eta(z_i) < \eta(x_i) \\
(\eta(z_i) = \eta(x_i)) \land (f(z_i) \leq f(x_i))
\end{align*}
\]

(15)

\[\text{the newborn trial vector } z_i \in X \text{ is selected for a new vector } x_i \in P_{t+1} \text{ of the next generation. Otherwise, the current target vector } x_i \in P_t \text{ survives to the next generation and becomes the new vector } x_i \in P_{t+1}.\]

\[\text{The algorithm of ADEP is described as follows. The maximum number of generations } N_P \text{ is given as the termination condition. The population size is chosen as } N_P = 5D \text{ [19]. Other parameters of ADEP are}\]

\[\text{We expect that } \hat{p}(x, \Phi, B) \approx \hat{p}(x, B) \text{ holds. Then, in order to obtain a solution of CCP in (5), we solve the following CCP defined by the probability in (9).}\]

\[
\min_{x \in X} f(x) \quad \text{s.t.} \quad \hat{p}(x, \Phi, B) \geq \alpha
\]

(10)

\[\text{Simple Random Sampling (SRS) is the most popular}\]

\[\text{sampling technique [16]. SRS is widely used due to its}\]

\[\text{easy execution and simplicity [17]. SRS selects some}\]

\[\text{samples } \xi^n \in B, n \in [N] \text{ randomly from the large data}\]

\[\text{set for making a sample set } \Phi_R = \{\xi^n\} \subseteq B. \text{ The sample size } N \text{ is far smaller than the data size } |B|.\]

\[\text{As shown in (6), each sample } \xi^n \in \Phi_R \text{ is labeled for a solution } x \in X \text{ as } y_n(\xi^n | x) \in \{-1, 1\}. \text{ Then the sample set } \Phi_R \text{ is used as the training data. The sample}\]

\[\text{set } \Phi_R \subseteq B \text{ is also used for } \Phi \subseteq B \text{ in (7) to (10).}\]

\[\text{Space Stratified Sampling (SSS) is a new sampling}\]

\[\text{technique proposed in this paper. SSS makes a set of}\]

\[\text{samples } \Phi_S \subseteq B \text{ for training SVM as follows:}\]

\[\text{Step 1: From the Principal Component Analysis}\]

\[\text{(PCA) [16] of the data set } B \subseteq \mathbb{R}^K, \text{ compute the principal components } v_j \in \mathbb{R}^K, j = 1, \cdots, K.\]

\[\text{Step 2: Divide the large data set } B \subseteq \mathbb{R}^K \text{ exclusively}\]

\[\text{into several strata } B_h \subseteq B, h \in [H] \text{ along the}\]

\[\text{axis of the first principal component } v_1 \in \mathbb{R}^K \text{ as}\]

\[
B = B_1 \cup \cdots \cup B_h \cup \cdots \cup B_H
\]

(11)

\[\text{where } B_h \cap B_h = \emptyset \text{ if } h \neq h, \text{ and } B_h \neq \emptyset.\]

\[\text{Step 3: Select some samples } \xi^n \in \Phi_h \subseteq B_h \text{ randomly}\]

\[\text{from each stratum } B_h, h \in [H]. \text{ Then get a set}\]

\[\text{of samples } \Phi_S \subseteq B \text{ by a union of them as}\]

\[
\Phi_S = \Phi_1 \cup \cdots \cup \Phi_h \cup \cdots \cup \Phi_H
\]

(12)

\[\text{where } |\Phi_1| = \cdots = |\Phi_h| = \cdots = |\Phi_H|.\]

Fig. 1 illustrates the stratification of the big data \( B \) along the axis \( v_1 \in \mathbb{R}^K \) by SSS in a case of \( H = 5.\)

As shown in (6), each sample \( \xi^n \in \Phi_S \) is labeled for a solution \( x \in X \) as \( y_n(\xi^n | x) \in \{-1, 1\}. \) Then the sample set \( \Phi_S \) is used as the training data. The sample set \( \Phi_S \subseteq B \) is also used for \( \Phi \subseteq B \) in (7) to (10).
Table 1: Probability for (17) by data set $B$

| $\gamma$ | 1.5 | 1.6 | 1.7 |
|---------|-----|-----|-----|
| $\hat{p}(x, B)$ | 0.50 | 0.75 | 0.92 |

controlled adaptively in the same way with a popular ADE [21]. The pruning technique is used in Step 6.

[ADEP Algorithm]

Step 1: Randomly generate the initial population $x_i \in P_t \subseteq X$, $i = 1, \cdots, N_p$, $t = 0$.

Step 2: For $i = 1$ to $N_p$, evaluate $f(x_i)$ and $\eta(x_i)$ for each vector $x_i \in P_0$ in the population.

Step 3: If $t = N_T$ holds, output the best solution $x_b \in P_t$ in the population and terminate ADEP.

Step 4: For $i = 1$ to $N_p$, generate the trial vector $z_i \in X$ from the target vector $x_i \in P_t$.

Step 5: For $i = 1$ to $N_p$, evaluate $f(z_i)$ for $z_i \in X$.

Step 6: For $i = 1$ to $N_p$, evaluate $\eta(z_i)$ for $z_i \in X$ only if the condition in (14) is not satisfied.

Step 7: For $i = 1$ to $N_p$, select either $z_i \in X$ or $x_i \in P_{t+1}$ for $x_i \in P_{t+1}$. $t = t + 1$.

Step 8: Go back to Step 3.

5 Performance Evaluation of SSS

The performance of the proposed SSS is compared with the conventional SRS through a test problem. All programs in this paper are coded in MATLAB [23].

5.1 Test Problem

In practical CCP in (2), each element $\xi_i \in \mathbb{R}$ of $\xi \in \Omega$ usually has a limited range. Therefore, a large data set $\xi^1 = (\xi^1_1, \xi^1_2) \in B$, $|B| = 10^6$ is generated randomly by a multivariate truncated normal distribution as

$$
\begin{cases}
\xi^1_1 \sim \mathcal{N}(\mu_1, \sigma^2_1) = \mathcal{N}(2, 0.2^2) \\
\xi^1_2 \sim \mathcal{N}(\mu_2, \sigma^2_2) = \mathcal{N}(1, 0.1^2)
\end{cases}
$$

(16)
Fig. 7: Estimation error for \( \hat{p}(x, B) = 0.50 \)

Fig. 8: Estimation error for \( \hat{p}(x, B) = 0.75 \)

Fig. 9: Estimation error for \( \hat{p}(x, B) = 0.92 \)

where \( \mu_j - 3 \sigma_j \leq \xi^\ell \leq \mu_j + 3 \sigma_j \), \( j = 1, 2 \). The correlation coefficient between \( \xi_1^\ell \) and \( \xi_2^\ell \) is \( \rho_{12} = 0.8 \).

By using the SVM with \( x = (x_1, x_2) = (1, 1) \), we estimate the probability that the condition

\[
g(x, \xi^\ell) = \frac{x_1 \xi_1^\ell + x_2 \xi_2^\ell}{2} \leq \gamma
\]

is satisfied for the large data set \( \xi^\ell \in B \) in (16).

Table 1 shows the true probability that the condition in (17) is satisfied for \( \xi^\ell \in B \) in (16) with \( \gamma \in \mathbb{R} \).

5.2 Experimental Result

Fig. 2 shows the large data \( \xi^\ell \in B \), \( |B| = 10^6 \) that is classified by the indicator function in (3) with \( \gamma = 1.6 \). Fig. 3 shows a set of samples \( \xi^n \in \Phi_R \subseteq B \), \( N = 80 \) selected by SRS. Similarly, Fig. 4 shows a set of samples \( \xi^n \in \Phi_S \subseteq B \), \( N = 80 \) selected by SSS with \( H = 10 \).

From Fig. 3 and Fig. 4, we can see that samples \( \xi^n \in \Phi_S \) selected by SSS are scattered more widely as compared to the random samples \( \xi^n \in \Phi_R \) selected by SRS. Especially, SRS has not taken any samples \( \xi^\ell \in B \) from the sparse part of \( B \subseteq \mathbb{R}^2 \) shown in Fig. 2.

As stated above, SVM trained by SSS is called S_SVM, while SVM trained by SRS is called R_SVM. Fig. 5 shows the large data \( \xi^\ell \in B \) classified by R_SVM.

Similarly, Fig. 6 shows \( \xi^\ell \in B \) classified by S_SVM. From Fig. 5 and Fig. 6, we can see that the estimated classification in Fig. 6 resembles the true classification in Fig. 2. Therefore, S_SVM is better than R_SVM.

The estimation error is defined for \( \Phi \subseteq B \) as

\[
e = |\hat{p}(x, B) - \hat{p}(x, \Phi, B)|
\]

where the sample set \( \Phi \) denotes either \( \Phi_R \) or \( \Phi_S \).

Each value \( \gamma \in \mathbb{R} \) shown in Table 1 is used in (17) for a comparison between S_SVM and R_SVM. Then, by changing the sample size \( N \), the estimation error in (18) is evaluated 100 times for each sample set and averaged in Fig. 7 to Fig. 9. From Fig. 7 to Fig. 9, the estimation error depends not only on the sample size but also on the probability value to be estimated. Besides, S_SVM outperforms R_SVM, especially when the true probability value is large. Since a large value is usually chosen for the sufficiency level \( \alpha \in (0, 1) \) of CCP, we can say that the sample set \( \Phi_S \subseteq B \) by SSS is more suitable for formulating CCP in (10).

6 Flood Control Planning

6.1 Formulation of CCP

In order to protect an urban area at the lower part of river from the flood damage caused by torrential rain, the flood control planning is formulated as CCP [4].

Fig. 10 shows a topological river model. Symbol \( \bigcirc \) denotes a forest. Symbol \( \triangle \) denotes a reservoir. There are three forests in watersheds. Rain falls in the three forests. The gross area of each forest \( a_j \in \mathbb{R}, j = 1, 2, 3 \) is a constant. The amount of rainfall \( \xi_j \in \mathbb{R} \) per unit area in each forest is regarded as a random variable.
The water-retaining capacity of forest \( x_j \in \mathbb{R} \) per unit area is regarded as a decision variable because it can be controlled through the forest maintenance such as afforestation. According to the model of the forest mechanism [24], the inflow of water \( q_j \in \mathbb{R}, j = 1, 2, 3 \) from each forest to the river is described as

\[
q_j(x_j, \xi_j) = a_j (\xi_j - x_j (1 - \exp(-\xi_j/x_j))). \tag{19}
\]

The capacity of each reservoir \( x_j \in \mathbb{R}, j = 4, 5, 6 \) is also a decision variable. The maintenance cost of a forest is proportional to its capacity. The construction cost of a reservoir is proportional to the square of its capacity. Therefore, the flood control planning to minimize the total cost is formulated as CCP [4]:

\[
\begin{align*}
\min_{x \in \mathcal{X}} \quad & f(x) = \sum_{j=1}^{3} a_j x_j + \sum_{j=4}^{6} x_j^2 \\
\text{sub. to} \quad & \Pr(g_m(x, \xi) \leq 0, m \in [3]) \geq \alpha, \\
& 0.5 \leq x_1 \leq 1.5, \ 0.5 \leq x_2 \leq 1.5, \\
& 0.5 \leq x_3 \leq 1.5, \ 0 \leq x_4 \leq 3, \\
& 0 \leq x_5 \leq 3, \ 0 \leq x_6 \leq 4
\end{align*}
\tag{20}
\]

where \( a_j = 2, j = 1, 2, 3 \). Functions \( g_m(x, \xi), m \in [3] \) are defined by using \( q_j(x_j, \xi_j) \) given in (19) as

\[
\begin{align*}
g_1(x, \xi) &= g_1(x_1, \xi_1) = x_4 - 0 \\
g_2(x, \xi) &= g_2(x, \xi) + q_2(x_2, \xi_2) - x_5 \leq 0 \\
g_3(x, \xi) &= g_3(x, \xi) + q_3(x_3, \xi_3) - x_6 \leq 0.
\end{align*}
\tag{21}
\]

The mean and variance in (22) are given as

\[
\begin{align*}
\mu_j - \sigma_j & \leq \xi_j \leq \mu_j + \sigma_j, j = 1, 2, 3.
\end{align*}
\tag{23}
\]

The correlation matrix of \( \xi_j \in \mathbb{R} \) is also given as

\[
R = \begin{pmatrix}
1.0 & \rho_{12} & \rho_{13} \\
\rho_{21} & 1.0 & \rho_{23} \\
\rho_{31} & \rho_{32} & 1.0
\end{pmatrix}
\]

\[
= \begin{pmatrix}
1.0 & 0.5 & 0.0 \\
0.5 & 1.0 & 0.3 \\
0.0 & 0.3 & 1.0
\end{pmatrix} \tag{24}
\]

By using the above large data set \( \mathcal{B} \subseteq \mathbb{R}^3, |\mathcal{B}| = 10^6 \), CCP in (20) is transformed into CCP shown in (5).

Table 2: Parameters of ADEP and SVM

| \( N_P \) | \( N_T \) | \( N \) | \( H \) |
|----------|----------|-------|-------|
| 30       | 60       | 160   | 10    |

Fig. 11: Convergence of \( f(x_b) \) and \( \eta(x_b) \)

Fig. 12: Convergence of \( x_{b,j} \in \mathbb{R}, j = 1, \cdots, 6 \)

6.2 Solution of CCP

ADEP combined with SVM is applied to CCP in (20). Table 2 shows the parameters of ADEP and SVM. Fig. 11 shows the objective function value \( f(x_b) \) and the constraint violation value \( \eta(x_b) \) achieved by the best solution \( x_b \in P_t \) in each generation \( t \). The decision variables \( x_{b,j} \in \mathbb{R}, j = 1, \cdots, 6 \) of the best solution \( x_b \in P_t \) are shown in Fig. 12. From Fig. 11 and Fig. 12, we can see that the best solution \( x_b \in P_t \) obtained by ADEP has converged with \( N_T = 60 \).

The best solution \( x_b \in X \) obtained by ADEP has been verified to be a feasible solution of CCP as

\[
\begin{align*}
f(x_b) &= 14.519 \\
\hat{\mu}(x_b, \Phi_S, \mathcal{B}) &= 0.914 \tag{25} \\
\hat{\sigma}(x_b, \mathcal{B}) &= 0.919.
\end{align*}
\]

The effect of the pruning technique is also confirmed because ADEP has discarded about 48% of the trial vectors \( z_i \in X \) without the evaluation of \( \eta(z_i) \).

In regards to the best solution \( x_b \in X \) of CCP, SVM is compared with R SVM. By using R SVM and SVM, respectively, the true probability \( \hat{\mu}(x_b, \mathcal{B}) \) in
(25) is estimated by changing the sample size $N$. Fig. 13 shows the estimation error in (18) averaged over 100 runs. From the result in Fig. 13, we can confirm that $S_{SVM}$ outperforms $R_{SVM}$ exactly. Furthermore, the sample size $N = 160$ for SSS in Table 2 is suitable.

7 Conclusions

For solving a data-driven CCP formulated by a large data set, a new approach is proposed. By using the large data set instead of the conventional mathematical model, the effect of uncertainties on the solution of CCP can be evaluated more accurately. However, the data set is usually too large to evaluate the probability of meeting all constraints of CCP. In order to utilize the large data set completely, a SVM is used to estimate the above probability from the large data set.

In order to generate a training data set for the SVM, a new sampling technique called SSS is proposed. The proposed SSS was compared with the well-known SRS. Experimental results indicated that SSS outperforms SRS. Furthermore, in order to solve CCP efficiently, an evolutionary algorithm called ADEP is proposed.

Finally, the performance of proposed approach is demonstrated through a real-world application, namely the flood control planning formulated as CCP.

In our future work, we will improve SSS for generating the training data set of SVM adaptively. For example, the information about support vectors given by SVM may be used to construct the proper strata of SSS.

References

[1] A. Prékopa: Stochastic Programming, Kluwer Academic Publishers, 1995.

[2] S. P. Uryasev: Probabilistic Constrained Optimization: Methodology and Applications, Kluwer Academic Publishers, 2001.

[3] M. Lubin, Y. Dvorkin, and S. Backhaus: A robust approach to chance constrained optimal power flow with renewable generation, IEEE Trans. on Power Systems, Vol. 31, No. 5, pp. 3840–3849, 2016.

[4] K. Tagawa and S. Miyanaga: An approach to chance constrained problems using weighted empirical distribution and differential evolution with application to flood control planning, Electronics and Communications in Japan, Vol. 102, No. 3, pp. 45–55, 2019.

[5] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty: Nonlinear Programming: Theory and Algorithm, John Wiley & Sons, 2006.

[6] C. A. Poojari and B. Varghese: Genetic algorithm based technique for solving chance constrained problems, European Journal of Operational Research, Vol. 185, pp. 1128–1154, 2008.

[7] B. Liu, Q. Zhang, F. V. Fernández, and G. G. E. Gielen: An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization, IEEE Trans. on Evolutionary Computation, Vol. 17, No. 6, pp. 786–796, 2013.

[8] K. Tagawa and S. Miyanaga: Weighted empirical distribution based approach to chance constrained optimization problems using differential evolution, Proc. of IEEE Congress on Evolutionary Computation, pp. 97–104, 2017.

[9] D. P. Kroese, T. Taimre, and Z. I. Botev: Handbook of Monte Carlo Methods, Wiley, 2011.

[10] K. Tagawa: Group-based adaptive differential evolution for chance constrained portfolio optimization using bank deposit and bank loan, Proc. of IEEE Congress on Evolutionary Computation, pp. 1556-1563, 2019.

[11] L. D. Xu, W. He, and S. Li: Internet of things in industries: a survey, IEEE Trans. on Industrial Informatics, Vol. 10, No. 4, pp. 2233–2243, 2014.

[12] E. Rossi, C. Rubatton, and G. Viscusi: Big data use and challenges: insights from two internet-mediated surveys, MDPI Computers, Vol. 8, No. 73, pp. 1–15, 2019.

[13] K. Tagawa: A big data based approach to chance constrained problems using weighted stratified sampling and differential evolution, Proc. of SICE2019, pp. 1473–1478, 2019.

[14] K. Tagawa: An approach to chance constrained problems based on huge data sets using weighted stratified sampling and adaptive differential evolution, MDPI Computers, Vol. 9, No. 32, pp. 1–20, 2020.

[15] S. Abe: Support Vector Machines for Pattern Classification, Springer-Verlag, 2nd. ed., 2010.

[16] J. Han, M. Kamber, and J. Pei: Data Mining - Concepts and Techniques, Morgan Kaufmann, 2012.
[17] R. Tempo, G. Calafiore, and F. Dabbene: *Randomized Algorithms for Analysis and Control of Uncertain Systems: With Applications*, Springer, 2012.

[18] R. B. Ash: *Basic Probability Theory*, Dover, Downers Grove, 2008.

[19] K. V. Price, R. M. Storn, and J. A. Lampinen: *Differential Evolution - A Practical Approach to Global Optimization*, Springer, 2005.

[20] R. Tanabe and A. Fukunaga: Reviewing and benchmarking parameter control methods in differential evolution, *IEEE Trans. on Evolutionary Computation*, Vol. 50, No. 3, pp. 1170–1184, 2020.

[21] J. Brest, S. Greiner, B. Bošković, M. Merink, and V. Žumer: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, *IEEE Trans. on Evolutionary Computation*, Vol. 10, No. 6, pp. 646–657, 2006.

[22] E. E. Montes and C. A. Coello Coello: Constraint-handling in nature inspired numerical optimization: past, present and future, *Swarm and Evolutionary Computation*, Vol. 1, pp. 173–194, 2011.

[23] A. R. Martinez and W. L. Martinez: *Computational Statistics Handbook with MATLAB*, Chapman & Hall/CRC, 2008.

[24] E. Maita and M. Suzuki: Quantitative analysis of direct runoff in a forested mountainous, small watershed, *Journal of Japan Society of Hydrology and Water Resources*, Vol. 22, No. 5, pp. 342–355, 2009.