A Stochastic Optimization Method Using Weighted Empirical Distribution Function

Kiyoharu Tagawa† and Shun Miyanaga‡
†School of Science and Engineering, Kindai University
‡Graduate School of Science and Engineering Research, Kindai University
3-4-1 Kowakae, Higashi-Osaka 577-8502 Japan

E-mail: tagawa@info.kindai.ac.jp

Abstract

This paper provides a new approach to solve a Chance Constrained Problem (CCP). The CCP is formulated via Cumulative Distribution Function (CDF). Hence, instead of the primitive Monte Carlo simulation, an approximation of CDF can be used to evaluate the solution of the CCP. In order to approximate CDF, two kinds of techniques, Empirical CDF (ECDF) and Weighted Empirical CDF (W_ECDF), are presented. Furthermore, for solving the CCP efficiently, a new Differential Evolution (DE) based optimization method combined with either ECDF or W_ECDF is proposed. The results of numerical experiments show that DE with W_ECDF outperforms DE with ECDF.

1 Introduction

In real-world optimization problems, a wide range of uncertainties have to be taken into account. The presence of uncertainty leads to different results for repeated evaluations of the same solution. Currently, two principal methods have been proposed for handling uncertainties. The first one is stochastic optimization [1] and the second one is robust optimization [2]. Since the robust optimization always considers the worst-case performance, its solution is rather conservative.

Chance Constrained Problem (CCP) [1] is one of the possible formulations of the stochastic optimization. In the CCP formulation, the quality of its solution can be guaranteed with a specified probability. However, CCP has been regarded as a very hard problem. That is because time-consuming Monte Carlo simulations are needed to evaluate solutions. Furthermore, traditional optimization methods that cope with CCP are based on the techniques of the non-linear programming [3]. Even though a number of Evolutionary Algorithms (EAs) have been reported for solving optimization problems under uncertainties [4], a few EAs deal with CCP [5].

This paper provides a new approach to solve a CCP. The CCP is formulated via Cumulative Distribution Function (CDF). Therefore, the solution of the CCP is evaluated by using an approximation of CDF instead of the primitive Monte Carlo simulation. To approximate CDF, two kinds of techniques, namely Empirical CDF (ECDF) [6] and Weighted Empirical CDF (W_ECDF) [7], are presented. Furthermore, for solving the CCP efficiently, a new Differential Evolution (DE) [8] based optimization method combined with either ECDF or W_ECDF is proposed. From the results of numerical experiments conducted on test problems, it is shown that DE with W_ECDF outperforms DE with ECDF.

The remainder of the paper is organized as follows. Section 2 formulates CCP. Section 3 presents ECDF and W_ECDF. Section 4 proposes an optimization method based on DE. Section 5 gives some numerical experiments. Section 6 provides conclusions.

2 Problem Formulation

Let \( x \in X = \{x_1, \ldots, x_D\} \subseteq \mathbb{R}^D \) be a vector of decision variables. By using a cost function \( g : X \rightarrow \mathbb{R} \), a deterministic optimization problem is stated as

\[
\min_{x \in X} \ g(x) \quad (1)
\]

where \( X = \{x \in \mathbb{R}^D \mid \underline{x}_j \leq x_j \leq \overline{x}_j, \ j = 1, \ldots, D\} \).

We assume that the uncertainty is given by a vector of random variables \( \xi \in \Xi \) with support \( \Xi \). Then the cost function in (1) is extended as \( g : X \times \Xi \rightarrow \mathbb{R} \).

Since \( \xi \in \Xi \) is stochastic, the value of the new cost function \( Y = g(x, \xi) \) becomes a random variable, too. Therefore, we transform the deterministic problem in (1) into a CCP in which an objective value \( y_0 \in \mathbb{R} \) is minimized for a specified probability \( \alpha \in (0, 1) \) as

\[
\left\{ \begin{array}{l}
\min_{x \in X} \ y_0 \\
\text{sub. to } \ Pr(g(x, \xi) \leq y_0) \geq \alpha 
\end{array} \right\} \quad (2)
\]

The CDF of the random variable \( Y = g(x, \xi) \) is defined as \( F(y) = Pr(Y \leq y) \). Now, we consider the \( \alpha \)-quantile of \( Y \) that depends on the solution \( x \in X \).

Then, by using the \( \alpha \)-quantile of \( Y \), the CCP in (2) can be described as an unconstrained problem:

\[
\min_{x \in X} \ y_\alpha(x) = F^{-1}(\alpha) . \quad (3)
\]
We suppose that the Probability Density Function (PDF) of $\xi \in \Xi$, which is denoted by $f(\xi)$, is known. However, in many real-world problems, we can’t derive $F(y)$ from $f(\xi)$ analytically. That is because the procedure of $g(x, \xi)$ is too complex. Therefore, time-consuming Monte Carlo simulations have been used widely for evaluating the solutions of CCPs [3, 5].

3 Approximation of CDF

By using a technique of the computational statistics, we try to compose an approximated CDF of $Y$ from a set of samples $Y^n = g(x, \xi^n)$, $n = 1, 2, \cdots, N$. Then, instead of the Monte Carlo simulation, we use the approximated CDF of $Y$ to evaluate the objective value $y_\alpha(x) \in \mathbb{R}$ of CCP in (3). For composing the approximated CDF from a set of samples, we present two techniques, namely ECDF and W_ECDF.

The indicator of event $Y^n \leq y$ is defined as

$$\iota(Y^n \leq y) = \begin{cases} 1; & \text{if } Y^n \leq y, \\ 0; & \text{otherwise.} \end{cases}$$

3.1 Empirical CDF (ECDF)

ECDF [6] is a step function that is composed from a set of samples $Y^n = g(x, \xi^n)$, $n = 1, 2, \cdots, N$ as

$$F(y) = \frac{1}{N} \sum_{n=1}^{N} \iota(Y^n \leq y).$$

Let $\tilde{F}_e(y)$ be a smoothed ECDF. The objective value is estimated as $y_\alpha(x) = \tilde{F}_e^{-1}(\alpha)$. As a drawback of ECDF, a large number of samples $Y^n$, $n = 1, \cdots, N$ are needed to approximate CDF. That is because the number of samples $Y^n = g(x, \xi^n)$, $\xi^n \sim f(\xi)$ taken from the tail part of PDF $f(\xi)$ is very small.

3.2 Weighted Empirical CDF (W_ECDF)

In order to approximate the CDF of $Y$ exactly from fewer samples than the conventional ECDF, we have proposed W_ECDF [7]. Each sample $Y^n = g(x, \xi^n)$ has a weight $f(\xi^n) \in (0, \infty)$ that is given by its own value of PDF. Thereby, W_ECDF is a step function composed from $Y^n = g(x, \xi^n)$, $n = 1, \cdots, N$ as

$$F_w(y) = \frac{1}{W} \sum_{n=1}^{N} f(\xi^n) \iota(Y^n \leq y),$$

$$W = \sum_{n=1}^{N} f(\xi^n).$$

Let $\tilde{F}_w(y)$ be a smoothed W_ECDF. The objective value is estimated as $y_\alpha(x) = \tilde{F}_w^{-1}(\alpha)$. Contrary to ECDF, the samples $\xi^n \in \Xi$ in (6) are not required to follow the original PDF of $\xi$. Therefore, a multi-dimensional Halton sequence [6] is used to make a set of samples $\xi^n \in \Xi$, $n = 1, \cdots, N$ for W_ECDF.

3.3 Comparison of ECDF and W_ECDF

We show a simple example of ECDF and W_ECDF. Let $\xi \sim \mathcal{N}(0, 1)$ be a random variable following the standard normal distribution. We composed the step
functions of ECDF and \(W_{ECDF}\) respectively from \(N = 10\) samples \(\xi^n \sim N(0, 1), n = 1, \cdots, N\).

Fig. 1 shows the exact CDF \(F(y)\). Fig. 2 shows the step function of ECDF and its smoothed one \(\tilde{F}_e(y)\). Fig. 3 also shows the step function of \(W_{ECDF}\) and its smoothed one \(\tilde{F}_w(y)\). From Figs. 1 to 3, we can confirm that \(\tilde{F}_w(y)\) resembles the exact \(F(y)\) than \(\tilde{F}_e(y)\).

4 Optimization Method

Since DE [8] is arguably one of the most powerful stochastic real-parameter optimization algorithms in current use, DE is employed to solve CCP in (3).

The classic DE proposed by R. Storn and K. Price [9] is based on the generation model. The classic DE uses two populations, namely primary one and secondary one. On the other hand, the novel DE [10, 11] based on the steady-state model uses a single population. The novel DE outperforms the classic DE in the convergence to the best solution [11]. Therefore, the novel DE is used in the proposed optimization method.

4.1 Initialization

Like other EAs, DE holds \(N_p\) tentative solutions of CCP in (3), which are referred as “individuals,” in the population \(P\). The \(i\)-th individual \(x_i \in P\), \(i = 1, \cdots, N_p\) is a real-parameter vector as

\[
x_i = (x_{i,1}, \cdots, x_{j,i}, \cdots, x_{D,i})
\]

where \(x_{j,i} \in \mathbb{R}\) and \(x_{j,i} \leq x_{j,\hat{i}} \leq \mathbb{R}\), \(j = 1, \cdots, D\).

Let \(\text{rand}_j \in [0, 1]\) be a uniformly distributed random value. To restrict \(x_{j,i} \in \mathbb{R}\) within the range \([\underline{\xi}_j, \overline{\xi}_j]\), an initial population \(P\) is generated randomly as

\[
x_{j,i} = \text{rand}_j (\overline{\xi}_j - \underline{\xi}_j) + \underline{\xi}_j
\]

where \(j = 1, \cdots, D\) and \(i = 1, \cdots, N_p\).

Each individual \(x_i \in P\) is evaluated \(N\) times such as \(Y^n = y(x_i, \xi^n), n = 1, \cdots, N\). From the set of samples, the objective value \(y_a(x_i)\) is estimated by using ECDF in (5) or \(W_{ECDF}\) in (6) as stated above.

The performance of DE depends on the values of the control parameters, namely the scale factor \(S_F \in [0, 1]\) and the crossover rate \(C_R \in [0, 1]\), which are given by users in advance. Due to avoid a strong dependency on the control parameters, a self-adaptive setting of them [12] is used. A different pair of parameter values \(S_{F,i}\) and \(C_{R,i}\) is assigned to each \(x_i \in P\), \(i = 1, \cdots, N_p\).

4.2 Strategy

In order to generate a candidate for a new \(x_i \in P\), DE uses a reproduction process called “strategy” [8]. The strategy of DE is defined exactly by a series of three genetic operators, namely 1) reproduction selection, 2) differential mutation, and 3) crossover. Even though various strategies have been proposed for DE [8, 10], a basic strategy named “DE/rand/1/bin” [8] is described and used by the proposed optimization method.

\begin{algorithm}
\begin{algorithmic}
  \State \textbf{for} \(i := 1\) to \(N_p\) \textbf{do}
  \State \hspace{1em} Generate \(x_i \in P\);
  \State \hspace{1em} Evaluate \(y_a(x_i)\);
  \State \hspace{1em} \((S_{F,i}, C_{R,i}) := (0.5, 0.9)\);
  \State \hspace{1em} \textbf{end for}
  \State \textbf{repeat}
  \State \hspace{1em} \textbf{for} \(i := 1\) to \(N_p\) \textbf{do}
  \State \hspace{2em} \((S_F, C_R) := \text{SELF-ADAPTIVE}(S_{F,i}, C_{R,i})\);
  \State \hspace{2em} \(u := \text{STRATEGY}(x_i, P, S_F, C_R)\);
  \State \hspace{2em} Evaluate \(y_a(u)\);
  \State \hspace{2em} \textbf{if} \(y_a(u) \leq y_a(x_i)\) \textbf{then}
  \State \hspace{3em} \(x_i := u\); /\* \(x_i \in P\) */
  \State \hspace{2em} \textbf{end if}
  \State \hspace{1em} \textbf{end for}
  \State \hspace{1em} \textbf{end for}
  \State \hspace{1em} \textbf{until} a termination condition is satisfied;
  \State \hspace{1em} \(x_b := \text{SELECT\_BEST}(P)\);
\end{algorithmic}
\end{algorithm}

In the reproduction selection, each individual \(x_i \in P\) is assigned to “target vector” in turn. Except for the target vector \(x_i\), three other distinct individuals, say \(x_{r1}, x_{r2}, \text{and} x_{r3} \in P\) \((i \neq r1 \neq r2 \neq r3)\), are selected randomly from the current population \(P\).

The scale factor \(S_F\) and the crossover rate \(C_R\) are decided adaptively from the parameters \(S_{F,i}\) and \(C_{R,i}\) accompanying the current target vector \(x_i \in P\) as

\[
S_F = \begin{cases} 
0.1 + \text{rand}_1 \times 0.9, & \text{if } \text{rand}_2 < 0.1, \\
\text{S}_F; & \text{otherwise,}
\end{cases}
\]

\[
C_R = \begin{cases} 
\text{rand}_3; & \text{if } \text{rand}_4 < 0.1, \\
\text{C}_{R,i}; & \text{otherwise}
\end{cases}
\]

where \(\text{rand}_1 \text{ to rand}_4\) are random values in \([0, 1]\).

By using the above three individuals, the differential mutation generates a new real-parameter vector called “mutated vector” \(v = (v_1, \cdots, v_D) \in \mathbb{R}^D\) as

\[
v = x_{r1} + S_F (x_{r2} - x_{r3}).
\]

The binomial crossover between the mutated vector \(v\) and the target vector \(x_i\) generates a candidate for a new individual \(u = (u_1, \cdots, u_D) \in \mathbb{R}^D\) called “trial vector”. Each component \(u_j \in \mathbb{R}\) of the trial vector \(u \in \mathbb{R}^D\) is inherited from either \(v \in \mathbb{R}^D\) or \(x_i \in P\) as

\[
u_j = \begin{cases} 
\text{v}_j; & \text{if } \text{rand}_j \leq C_R \text{ v } (j = j_r), \\
x_{j,i}; & \text{otherwise}
\end{cases}
\]

where the subscript \(j_r \in [1, D]\) is selected randomly, which ensures that \(u \in \mathbb{R}^D\) differs from \(x_i \in P\).

If a component \(u_j \in \mathbb{R}\) of \(u \in \mathbb{R}^D\) is made out of the range \([\underline{\xi}_j, \overline{\xi}_j]\), it is returned to the range as

\[
u_j = \begin{cases} 
x_{j,i} + \text{rand}_j (\overline{\xi}_j - x_{j,i}); & \text{if } u_j < \underline{\xi}_j, \\
x_{j,i} + \text{rand}_j (\overline{\xi}_j - x_{j,i}); & \text{if } u_j > \overline{\xi}_j.
\end{cases}
\]

where \(x_{r1} \in P\) is the individual used in (11).
4.3 Survival Selection

The newborn trial vector \( \mathbf{u} \in \mathbf{X} \) is evaluated \( N \) times such as \( Y^n = g(\mathbf{u}, \xi^n), n = 1, \cdots, N \). From the set of samples, the objective value \( y_\alpha(\mathbf{u}) \) is estimated by using either ECDF in (5) or \( W_{\text{ECDF}} \) in (6) as stated above. Then the trial vector \( \mathbf{u} \in \mathbf{X} \) is compared with the target vector \( \mathbf{x}_i \in \mathbf{P} \). If \( y_\alpha(\mathbf{u}) \leq y_\alpha(\mathbf{x}_i) \), \( \mathbf{x}_i \in \mathbf{P} \) is replaced by \( \mathbf{u} \in \mathbf{X} \) immediately. As a result, the excellent trial vector can be used soon to generate succeeding trial vectors. Algorithm 1 provides the pseudo-code of the proposed DE for CCP in (3), which returns the best solution \( \mathbf{x}_\star \in \mathbf{P} \) in the end.

5 Numerical Experiments

5.1 Experiment 1

The following one-dimensional deterministic function [13] is used to make the instances of CCP in (2).

\[
g_1(x) = \begin{cases} 
1 - e(x) |\sin(5 \pi x)|^{0.5}; & \text{if } 0.4 < x \leq 0.6, \\
1 - e(x) \sin(5 \pi x); & \text{otherwise} 
\end{cases}
\]  
(14)

where \( x \in [0, 1] \subseteq \mathbb{R} \) and \( e(x) \) is defined as

\[
e(x) = \exp \left(-2 \ln_2 \left(\frac{x - 0.1}{0.8}\right)^2 \right).
\]  
(15)

By adding disturbance to the decision variable \( x \in \mathbb{R} \), we extend the deterministic function to a stochastic cost function. Besides, we suppose that the disturbance is a normally distributed random number. Specifically, the cost function for CCP is defined as

\[
g_1(x, \xi) = g_1(x + \xi), \quad \xi \sim \mathcal{N}(0, \sigma^2)
\]  
(16)

By using the cost function in (16), two instances of CCP in (2) are formulated as follows. For all instances of CCP, the probability is specified as \( \alpha = 0.95 \).

**CCP1a:** \( \Pr(g_1(x, \xi) \leq y_\alpha), x \in [0, 1], \sigma = 0.01 \)

**CCP1b:** \( \Pr(g_1(x, \xi) \leq y_\alpha), x \in [0, 1], \sigma = 0.012 \)

In order to solve CCP in (2) with DE, it has to be transformed into the equivalence problem in (3).

We compared two variants of DE, namely DE(E) and DE(W). DE(W) uses \( W_{\text{ECDF}} \) to estimate \( y_\alpha(x) \in \mathbb{R} \), while DE(E) uses ECDF. Both DE(E) and DE(W) were coded by MATLAB@. The population size and the sample size were chosen respectively as \( N_P = 20 \) and \( N = 20 \). The maximum number of generations was fixed to 20 as the termination condition of DE.

DE(E) and DE(W) were applied respectively to CCP1a 20 times. As a result, each DE found 20 best solutions for CCP1a. Fig. 4 compares the landscapes of the deterministic function value \( g_1(x) \in \mathbb{R} \) in (14) and the stochastic function value \( y_\alpha(x) \in \mathbb{R} \) of CCP1a. The optimal solution of CCP1a is known as \( x^\star = 0.100 \). From the landscape of CCP1a in Fig. 4, there are five valleys or local optima. We name those valleys A, B, C, D, and E from the left. As you can see in Fig. 4, the optimal solution \( x^\star = 0.100 \) exists in the valley “A”.

Table 1 shows the numbers of the best solutions found by DE(E) and DE(W) in the five valleys of CCP1a. From Table 1, DE(W) found more the best solutions than DE(E) in the correct valley “A”. From now on, we call the best solution \( x^\dagger \in \mathbf{P} \) found in the correct valley the correct solution. Table 1 also compares DE(W) with DE(H) in the rate of the correct solutions.

Table 2 also compares DE(W) with DE(E) in their correct solutions for CCP1a. \( E(x^\dagger) \) is the average value of the correct solutions. We define the accuracy of the correct solution \( x^\dagger \in \mathbf{P} \) as \( \delta = |x^\star - x^\dagger| \). \( E(\delta) \) is the average accuracy. Besides, Table 2 shows the result of Wilcoxon test about the accuracy with \( p \)-value. From the \( p \)-value in Table 2, DE(W) is significantly better than DE(E) in the accuracy of the correct solution.

Similarly, DE(E) and DE(W) were applied to CCP1b 20 times. Fig. 5 compares the landscapes of \( g_1(x) \) in (14) and \( y_\alpha(x) \) of CCP1b. From Fig. 4 and Fig. 5, we can observe that the objective value \( y_\alpha(x) \) depends on both the decision variable \( x \in \mathbb{R} \) and the variance \( \sigma^2 \).
From the landscape of CCP1b in Fig. 5, there are five valleys or local optima. We name those valleys A, B, C, D, and E from the left. The optimal solution $x^* = 0.487$ exists in the valley “C”. Table 3 shows the numbers of the best solutions found by DE(E) and DE(W) in the five valleys. From Table 3, DE(W) found all the best solutions $x_b \in P$ in the correct valley “C”.

In the same way with Table 2, Table 4 compares DE(W) with DE(E) in their correct solutions for CCP1b. From Table 1 to Table 4, we can conclude that the proposed DE(W) is superior to DE(E).

### 5.2 Experiment 2

The following two-dimensional deterministic function is used to make the instances of CCP in (2).

$$g_2(x) = g_2(x_1, x_2) = \sqrt{g_1(x_1)g_1(x_2)}$$  \quad (17)

where $g_1(x_j), j = 1, 2$ is defined by (14).

By adding disturbances to the decision variables, we extend the above deterministic function to a stochastic cost function for CCP as

$$g_2(x, \xi) = g_2(x_1 + \xi_1, x_2 + \xi_2)$$  \quad (18)

where $\xi_1, \xi_2 \sim N(0, \sigma^2)$ are mutually independent.

By using the cost function in (18), two instances of CCP in (2) are formulated as follows. For all instances of CCP, the probability is specified as $\alpha = 0.95$.

**CCP2a:** $Pr(g_2(x, \xi) \leq y_0), x \in [0, 1]^2, \sigma = 0.01$

**CCP2b:** $Pr(g_2(x, \xi) \leq y_0), x \in [0, 1]^2, \sigma = 0.02$

In order to solve CCP in (2) with DE, it has to be transformed into the equivalence problem in (3). We assessed the effects of the population size $N_P$ and the sample size $N$ on the performance of DE(W). The maximum number of generations was fixed to 60.

By using a different pair of the population size $N_P$ and the sample size $N$, DE(W) was applied to CCP2a 20 times. Table 5 shows the objective value $y_0(x_b) \in \mathbb{R}$ of the best solution $x_b \in P$ averaged over 20 runs. Table 6 summarizes the result of the two-way Analysis of Variance (ANOVA) about the objective value shown in Table 5. From the result of ANOVA, the population size $N_P$ has a significant effect on the objective value of the best solution obtained by DE(W). It seems that the effect of the sample size $N$ is small. Besides, the interaction between $N_P$ and $N$ has no effect.

Fig. 6 shows the contour of the objective value $y_0(x) \in \mathbb{R}$ for CCP2a. As you can see in Fig. 6, there are twenty five valleys or local optima. The optimal solution $x^* = (0.10, 0.10)$ exists in the narrow valley at the corner of the search space. Table 7 shows the rate of the correct solutions which are found by DE(W) with different pairs of $N_P$ and $N$. From Table 7, we can confirm that the performance of DE(W) depends on the population size rather than the sample size.

Similarly, DE(W) was applied to CCP2b 20 times.
Table 5: Objective value of best solution (CCP2a)

<table>
<thead>
<tr>
<th>N_P</th>
<th>N</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.195</td>
<td>0.177</td>
<td>0.175</td>
<td>0.158</td>
<td>0.163</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.151</td>
<td>0.141</td>
<td>0.126</td>
<td>0.118</td>
<td>0.136</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.150</td>
<td>0.129</td>
<td>0.136</td>
<td>0.122</td>
<td>0.126</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: ANOVA about objective value (CCP2a)

<table>
<thead>
<tr>
<th>factor</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_P</td>
<td>2</td>
<td>0.105</td>
<td>0.052</td>
<td>15.222</td>
<td>0.000</td>
</tr>
<tr>
<td>N</td>
<td>4</td>
<td>0.033</td>
<td>0.008</td>
<td>2.403</td>
<td>0.050</td>
</tr>
<tr>
<td>N_P × N</td>
<td>8</td>
<td>0.005</td>
<td>0.000</td>
<td>0.212</td>
<td>0.989</td>
</tr>
</tbody>
</table>

DF: Degree of Freedom  SS: Sum of Square  MS: Mean Square

Table 7: Rate of correct solutions (CCP2a)

<table>
<thead>
<tr>
<th>N_P</th>
<th>N</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>45</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>80</td>
<td>85</td>
<td>90</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>75</td>
<td>75</td>
<td>90</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Objective value of best solution (CCP2b)

<table>
<thead>
<tr>
<th>N_P</th>
<th>N</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.334</td>
<td>0.331</td>
<td>0.333</td>
<td>0.332</td>
<td>0.332</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.334</td>
<td>0.331</td>
<td>0.333</td>
<td>0.332</td>
<td>0.332</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.334</td>
<td>0.331</td>
<td>0.333</td>
<td>0.332</td>
<td>0.332</td>
<td></td>
</tr>
</tbody>
</table>

Table 8 shows the objective value \( y_a(x_b) \in \mathbb{R} \) of the best solution \( x_b \in P \) averaged over 20 runs.

Fig. 7 shows the contour of \( y_a(x) \in \mathbb{R} \) for CCP2b. From Fig. 7, there are twenty five valleys. The optimal solution \( x^* = (0.49, 0.49) \) exists in the biggest valley at the center of the search space. Since the correct valley is huge, DE(W) has found the correct solutions in any case. Therefore, the rates of them are 100\%.

From Table 5 to Table 8, the sample size doesn’t effect on the performance of DE(W). Therefore, if we control the sample size appropriately in the process, we may reduce the total number of samples spent by DE(W).

6 Conclusions

We have proposed a DE-based optimization method for CCP. We have evaluated the objective value of CCP by using an approximation of CDF instead of the Monte Carlo simulation. In order to approximate CDF, we have presented two techniques, namely ECDF and \( W_{ECDF} \). Through numerical experiments, we have shown that DE combined with \( W_{ECDF} \) is superior to DE combined with ECDF for solving CCP.

In our future work, we need to verify the performance of the proposed method in real-world CCPs.

References