# Multi-Noisy-objective Optimization Based on Prediction of Worst-Case Performance

Kiyoharu Tagawa[1] and Shoichi Harada[2]

[1] School of Science and Engineering, Kinki University
Higashi-Osaka 577-8502, Japan
tagawa@info.kindai.ac.jp

[2] Graduate School of Science and Engineering Research, Kinki University
Higashi-Osaka 577-8502, Japan

**Abstract.** This paper proposes a new approach to cope with multi-objective optimization problems in presence of noise. In the first place, since considering the worst-case performance is important in many real-world optimization problems, a solution is evaluated based on the upper bounds of respective noisy objective functions predicted statistically by multiple sampling. Secondary, a rational way to decide the maximum sample size for the solution is shown. Thirdly, to allocate the computing budget of a proposed evolutionary algorithm only to promising solutions, two pruning techniques are contrived to judge hopeless solutions only by a few sampling and skip the evaluation of the upper bounds for them.

**Keywords:** evolutionary computing, multi-objective optimization.

## 1 Introduction

Many real-world Multi-objective Optimization Problems (MOPs) have more than one objective function contaminated by noise. The presence of noise leads to different results for repeated evaluations of the same solution. Therefore, for solving Multi-Noisy-objective Optimization Problems (MNOPs), various Multi-Objective Evolutionary Algorithms (MOEAs) have also been reported. The goal of those MOEAs is to produce a set of distributed solutions that are not only of high quality, but also robust. However, there are many possible notations of robustness. Even among them, the worst-case performance is important in particular if the decision maker is very risk averse, or if the stakes are high.

This paper thinks about a new class of MNOPs in which the predicted upper bounds of respective noisy objective functions are minimized simultaneously. The predicted upper bounds of noisy objective functions provide a proper criterion to measure the worst-case performance. However, the multiple sampling of every solution to predict the upper bounds statistically is still expensive. Therefore, a novel MOEA based on Differential Evolution (DE) [1] is proposed for solving the new class of MNOPs effectively. In order to examine as many solutions as possible within a limited number of function evaluations, the proposed MOEA uses two pruning techniques, which are called U-cut and C-cut respectively, to judge hopeless solutions only by a few sampling and skip their evaluations.

## 2 Related Work on Multi-Noisy-objective Optimization

To date, a number of methods including various MOEAs have been reported to solve MNOPs [2]. As stated above, the goal of those methods is to produce a set of distributed solutions that are not only of high quality, but also robust. There are many possible notations of robustness, including a good expected performance, a good worst-case performance, a low variability in performance, or a large range of disturbance still leading to acceptable performance [3].

In order to evaluate the good expected performance for a solution, averaging over multiple samples is a most fundamental approach. That is because it is applicable even if the properties of uncertainties are completely unknown [4,5]. On the other hand, some assumptions on the probability distribution of objective function values are often introduced into the problem formulation, namely, a normal distribution with constant variance [6,7,8], a normal distribution with variable variance [9,10], a uniform distribution [11], and so on. Thereby, statistical approaches such as Probabilistic dominance [6] can be used to compare two uncertain solutions. Incidentally, for the case that the objective functions are distributed normally with a constant variance, learning algorithms have also been reported to estimate the constant variance during the optimization [7].

In order to evaluate the worst-case performance for a solution, the concept of min-max robustness is introduced into the problem formulation. Thereby, the worst value of each objective function is found by the multiple sampling of the same solution [12]. Another interpretation of the uncertainty in MOPs is based on scenarios instead of noise. The objective function values for a given solution depend on scenarios. A set of objective function values for all possible scenarios is considered. Then the worst-case performance for the solution is obtained as a set of non-dominated objective function values by solving an inversed MOP [13,14]. The objective function values of a solution for all possible scenarios can be also depicted as a polygon in the objective space. Therefore, the worst-case performance of the solution is represented deterministically as a set of extremal points of the polygon. For finding those extremal points one by one, a single objective optimization algorithm is used repeatedly [15].

It can be seen that the previous work that addresses the worst-case performance in MNOPs is relatively limited. Moreover, to the best of our knowledge, the statistical approach based on the predicted upper bounds of noisy objective functions has not yet been reported. Because it is impossible to find the worst value of a stochastic objective function in a finite number of samples, we think that the statistical approach proposed in this paper is practically useful.

## 3 Problem Formulation

### 3.1 Noisy-objective and Prediction Interval

Let $\boldsymbol{x} = (x_1, \cdots, x_j, \cdots, x_D)$ denote a vector of decision variables $x_j \in \Re$ that can be changed by an algorithm. The decision vector $\boldsymbol{x} \in \Re^D$ is often referred to as a solution. An objective vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x}))$

depending on a solution $\boldsymbol{x} \in \Re^D$ is composed of $M$ $(M \geq 2)$ objective functions $f_m(\boldsymbol{x}) \in \Re$, $m \in \mathcal{I}_M = \{1, \cdots, M\}$. The objective vector $\boldsymbol{f}(\boldsymbol{x})$ is minimized in MOPs. Now, we assume that each objective function $f_m(\boldsymbol{x})$ is contaminated with noise in MNOPs. Therefore, every time a solution $\boldsymbol{x} \in \Re^D$ is evaluated, a different objective vector may be returned. Let $f_m^n(\boldsymbol{x}) \in \Re$, $n \in \mathcal{I}_N = \{1, \cdots, N\}$ be observed values of $f_m(\boldsymbol{x})$, which are distributed normally as

$$f_m^n(\boldsymbol{x}) \sim \mathcal{N}(\mu_m(\boldsymbol{x}),\ \sigma_m(\boldsymbol{x})^2) = \mathcal{N}(f_m(\boldsymbol{x}),\ \sigma_m(\boldsymbol{x})^2), \tag{1}$$

where the mean $\mu_m(\boldsymbol{x}) = f_m(\boldsymbol{x})$, $m \in \mathcal{I}_M$ and the variance $\sigma_m(\boldsymbol{x})^2$, $m \in \mathcal{I}_M$ are mutually independent functions that depend on the solution $\boldsymbol{x} \in \Re^D$.

Because the mean $\mu_m(\boldsymbol{x})$ and the variance $\sigma_m(\boldsymbol{x})^2$ in (1) are usually unknown, we have to estimate those values, respectively, by the sample mean and the unbiased variance. From a sample set $\{f_m^1(\boldsymbol{x}), \cdots, f_m^n(\boldsymbol{x}), \cdots, f_m^N(\boldsymbol{x})\}$ of an objective function $f_m(\boldsymbol{x})$ for $\boldsymbol{x} \in \Re^D$, the sample mean is calculated as

$$\overline{f}_m(\boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} f_m^n(\boldsymbol{x}). \tag{2}$$

The unbiased variance is also calculated from the sample set and (2) as

$$s_m(\boldsymbol{x})^2 = \frac{1}{N-1} \sum_{n=1}^{N} (f_m^n(\boldsymbol{x}) - \overline{f}_m(\boldsymbol{x}))^2. \tag{3}$$

By using $\overline{f}_m(\boldsymbol{x})$ and $s_m(\boldsymbol{x})^2$ instead of $\mu_m(\boldsymbol{x})$ and $\sigma_m(\boldsymbol{x})^2$ respectively, the normal distribution in (1) is approximated by Student's t-distribution. We have already obtained the sample set $\{f_m^n(\boldsymbol{x}) \in \Re \mid n \in \mathcal{I}_N\}$ of size $N$. Let $f_m^{N+1}(\boldsymbol{x})$ be the $(N+1)$-th sample, or the future observation of $f_m(\boldsymbol{x})$. Then the following statistic yields Student's t-distribution with $N-1$ degrees of freedom [16]:

$$\frac{f_m^{N+1}(\boldsymbol{x}) - \overline{f}_m(\boldsymbol{x})}{s_m(\boldsymbol{x}) \sqrt{1 + \dfrac{1}{N}}} \sim \mathcal{T}(N-1). \tag{4}$$

Let $\alpha$ $(0 < \alpha \leq 0.05)$ be a significance level. The one-side prediction interval in which the future observation $f_m^{N+1}(\boldsymbol{x})$ will fall is derived from (4) as

$$-\infty < f_m^{N+1}(\boldsymbol{x}) \leq \overline{f}_m(\boldsymbol{x}) + t(N-1,\, \alpha)\, s_m(\boldsymbol{x}) \sqrt{1 + \frac{1}{N}} = f_m^U(\boldsymbol{x}), \tag{5}$$

where $t(N-1,\, \alpha)$ is the $\alpha$-quantile of Student's t-distribution with $N-1$ degrees of freedom. The upper bound of the prediction interval is denoted by $f_m^U(\boldsymbol{x})$.

The probability of the future observation $f_m^{N+1}(\boldsymbol{x})$ of the noisy objective function $f_m(\boldsymbol{x})$ falling in the prediction interval shown in (5) is

$$\mathcal{P}(f_m^{N+1}(\boldsymbol{x}) \leq f_m^U(\boldsymbol{x})) = 1 - \alpha. \tag{6}$$

On the other hand, the probability that the future observation $f_m^{N+1}(\boldsymbol{x})$ doesn't fall in the prediction interval in (5) is very small such as

$$\mathcal{P}(f_m^U(\boldsymbol{x}) \leq f_m^{N+1}(\boldsymbol{x})) = \alpha. \tag{7}$$

## 3.2 Multi-Noisy-objective Optimization Problem

Let $\{\boldsymbol{f}^n(\boldsymbol{x}) = (f_1^n(\boldsymbol{x}), \cdots, f_m^n(\boldsymbol{x}), \cdots f_M^n(\boldsymbol{x})) \mid n \in \mathcal{I}_N\}$ be a sample set of an objective vector $\boldsymbol{f}(\boldsymbol{x}) \in \Re^M$ depending on a solution $\boldsymbol{x} \in \Re^D$. From (2), (3), (5), and the sample set $\{\boldsymbol{f}^n(\boldsymbol{x}) \in \Re^M \mid n \in \mathcal{I}_N\}$ of size $N$, we can predict the upper bound $\boldsymbol{f}^U(\boldsymbol{x}) \in \Re^M$ of the future observation $\boldsymbol{f}^{N+1}(\boldsymbol{x}) \in \Re^M$. We also suppose that each of decision variables $x_j \in \Re$, $j \in \mathcal{I}_D = \{1, \cdots, D\}$ is limited to the range between the lower $x_j^L$ and the upper $x_j^U$ bounds. Thereby, a Multi-Noisy-objective Optimization Problem (MNOP) is formulated as

$$\begin{bmatrix} \text{minimize} & \boldsymbol{f}^U(\boldsymbol{x}) = (f_1^U(\boldsymbol{x}), \cdots, f_m^U(\boldsymbol{x}), \cdots, f_M^U(\boldsymbol{x})), \\ \text{subject to} & \boldsymbol{x} = (x_1, \cdots, x_j, \cdots, x_D) \in \boldsymbol{X}, \end{bmatrix} \tag{8}$$

where $\boldsymbol{X} = \{\boldsymbol{x} \in \Re^D \mid \forall j \in \mathcal{I}_D : x_j^L \leq x_j \leq x_j^U\}$ is called the decision space. Furthermore, $\boldsymbol{F} = \{\boldsymbol{f}^U(\boldsymbol{x}) \in \Re^M \mid \boldsymbol{x} \in \boldsymbol{X}\}$ is called the objective space. In order to simplify the notation in this paper, we will sometimes use an objective vector $\boldsymbol{f}^U(\boldsymbol{x}) \in \boldsymbol{F}$ to represent a corresponding solution $\boldsymbol{x} \in \boldsymbol{X}$, and vice versa.

**Definition 1.** *A vector $\boldsymbol{v} = (v_1, \cdots, v_m, \cdots, v_M) \in \Re^M$ is said to dominate the other $\boldsymbol{v}' \in \Re^M$ and denoted as $\boldsymbol{v} \succ \boldsymbol{v}'$, if the following condition is true:*

$$(\forall m \in \mathcal{I}_M : v_m \leq v'_m) \;\wedge\; (\exists n \in \mathcal{I}_M : v_n < v'_n). \tag{9}$$

**Definition 2.** *A vector $\boldsymbol{v} = (v_1, \cdots, v_M) \in \Re^M$ is said to weakly dominate the other $\boldsymbol{v}' \in \Re^M$ and denoted as $\boldsymbol{v} \succeq \boldsymbol{v}'$, if the following condition is true:*

$$\forall m \in \mathcal{I}_M : v_m \leq v'_m. \tag{10}$$

From (6), the probability of $\boldsymbol{f}^{N+1}(\boldsymbol{x})$ weakly dominating $\boldsymbol{f}^U(\boldsymbol{x})$ is

$$\mathcal{P}(\boldsymbol{f}^{N+1}(\boldsymbol{x}) \succeq \boldsymbol{f}^U(\boldsymbol{x})) = \prod_{m=1}^{M} \mathcal{P}(f_m^{N+1}(\boldsymbol{x}) \leq f_m^U(\boldsymbol{x})) = (1-\alpha)^M. \tag{11}$$

From (7), the probability of $\boldsymbol{f}^U(\boldsymbol{x})$ weakly dominating $\boldsymbol{f}^{N+1}(\boldsymbol{x})$ is

$$\mathcal{P}(\boldsymbol{f}^U(\boldsymbol{x}) \succeq \boldsymbol{f}^{N+1}(\boldsymbol{x})) = \prod_{m=1}^{M} \mathcal{P}(f_m^U(\boldsymbol{x}) \leq f_m^{N+1}(\boldsymbol{x})) = \alpha^M. \tag{12}$$

## 3.3 Selection of Sample Size

For calculating an objective vector $\boldsymbol{f}^U(\boldsymbol{x})$ in (8), a solution $\boldsymbol{x} \in X$ needs to be evaluated $N$ times. Sampling size selection is actually a burden to balance the quality of the objective vector $\boldsymbol{f}^U(\boldsymbol{x})$ with the computational overhead.

We employ a rational way to determine an appropriate sample size $N$ from the accuracy of the unbiased variance $s_m(\boldsymbol{x})^2$ in (3). The both-side confidence interval of the variance $\sigma_m(\boldsymbol{x})^2$ appeared in (1) is given as follows [16]:

$$\frac{N-1}{\chi^2(N-1,\, \alpha/2)}\, s_m(\boldsymbol{x})^2 \leq \sigma_m(\boldsymbol{x})^2 \leq \frac{N-1}{\chi^2(N-1,\, 1-\alpha/2)}\, s_m(\boldsymbol{x})^2, \tag{13}$$

where $\chi^2(N-1,\,\alpha/2)$ and $\chi^2(N-1,\,1-\alpha/2)$ are the $\alpha/2$-quantile and the $(1-\alpha/2)$-quantile of the $\chi^2$-distribution with $N-1$ degrees of freedom.

Let $\delta$ ($\delta > 1$) be a tolerance for the ratio of the upper bound to the lower bound of the confidence interval in (13). Thereby, the ratio is limited as

$$\frac{\chi^2(N-1,\,\alpha/2)}{\chi^2(N-1,\,1-\alpha/2)} \le \delta. \tag{14}$$

From the condition in (14) and the Fisher's approximation of $\chi^2$-distribution [17], we decide a sample size $N$ for a given tolerance $\delta$ ($\delta > 1$) as follows:

$$N \ge \frac{1}{2}\left(\frac{(1+\sqrt{\delta})\,z_{\alpha/2}}{\sqrt{\delta}-1}\right)^2 + \frac{3}{2}, \tag{15}$$

where $z_{\alpha/2}$ is the $\alpha/2$-quantile of the standard normal distribution: $\mathcal{N}(0,\,1)$. The sample size $N$ increases quickly as we attempt to reduce the tolerance $\delta$.

## 4 Differential Evolution for MNOP

If we calculate the objective vector $\boldsymbol{f}^U(\boldsymbol{x})$ in (8) from the sample set of size $N$ for every examined solution $\boldsymbol{x} \in \boldsymbol{X}$, we can apply conventional MOEAs, such as NSGA-II [18] and DEMO [19], to MNOP without modification. In order to cope with MNOP, we select DEMO as the basic MOEA for its simplicity in coding, fewer control parameters, good accuracy, and fast speed convergence [5].

Algorithm 1 provides the pseudo-code of DEMO applied to MNOP. First of all, an initial population $\boldsymbol{P} \subset \boldsymbol{X}$ of size $N_P$ is generated randomly. Thereafter, the objective vector $\boldsymbol{f}^U(\boldsymbol{x}_i)$ is evaluated for each $\boldsymbol{x}_i \in \boldsymbol{P}$ from a sample set $\{\boldsymbol{f}^n(\boldsymbol{x}_i) \mid n \in \mathcal{I}_N\}$ of size $N$. Every solution $\boldsymbol{x}_i \in \boldsymbol{P}$, $i = 1, \cdots, N_P$ is chosen to be the target vector $\boldsymbol{x}_i$ in turn. By using a basic strategy named "DE/rand/1/exp" [1], a new trial vector $\boldsymbol{u} \in \boldsymbol{X}$ is generated from the target vector $\boldsymbol{x}_i \in \boldsymbol{P}$ and other solutions selected randomly in $\boldsymbol{P}$ at the 7th line.

The search efficiency of DE depends on the control parameters, namely the scale factor $S_F$ and the crossover rate $C_R$, which are used in the strategy. Thus, we introduce a self-adapting mechanism of them [20] into DEMO. A different set of parameter values $S_{F,i}$ and $C_{R,i}$ are assigned to each $\boldsymbol{x}_i \in \boldsymbol{P}$, $i = 1, \cdots, N_P$. The strategy generates $\boldsymbol{u}$ from $\boldsymbol{x}_i \in \boldsymbol{P}$ by using $S_F$ and $C_R$ decided as

$$S_F = \begin{cases} 0.1 + \mathrm{rand}_1[0,\,1]\,0.9, & \text{if } \mathrm{rand}_2[0,\,1] < 0.1, \\ S_{F,i}, & \text{otherwise}, \end{cases} \tag{16}$$

$$C_R = \begin{cases} \mathrm{rand}_3[0,\,1], & \text{if } \mathrm{rand}_4[0,\,1] < 0.1, \\ C_{R,i}, & \text{otherwise}, \end{cases} \tag{17}$$

where $\mathrm{rand}_k[0,\,1] \in [0,\;1]$ denotes a uniformly distributed random number.

The objective vector $\boldsymbol{f}^U(\boldsymbol{u})$ is evaluated for the trial vector $\boldsymbol{u}$ from a sample set $\{\boldsymbol{f}^n(\boldsymbol{u}) \mid n \in \mathcal{I}_N\}$ of size $N$ at the 8th line. In lines 9-15, the trial vector $\boldsymbol{u}$

**Algorithm 1.** DEMO APPLIED TO MNOP

```
1: $\boldsymbol{P}$ := GENERATE_INITIAL_POPULATION($N_P$);
2: for $i := 1$ to $N_P$ do
3:    $\boldsymbol{f}^U(\boldsymbol{x}_i)$ := PREDICT_UPPER_BOUND($\boldsymbol{f}^n(\boldsymbol{x}_i),\ n \in \mathcal{I}_N$);
4: end for
5: repeat
6:    for $i := 1$ to $N_P$ do
7:       $\boldsymbol{u}$ := STRATEGY($\boldsymbol{x}_i \in \boldsymbol{P}$);   /* Generate a new trial vector $\boldsymbol{u} \in \boldsymbol{X}$ */
8:       $\boldsymbol{f}^U(\boldsymbol{u})$ := PREDICT_UPPER_BOUND($\boldsymbol{f}^n(\boldsymbol{u}),\ n \in \mathcal{I}_N$);
9:       if $\boldsymbol{f}^U(\boldsymbol{u}) \succeq \boldsymbol{f}^U(\boldsymbol{x}_i)$ then
10:         $\boldsymbol{x}_i := \boldsymbol{u}$;   /* Replace $\boldsymbol{x}_i \in \boldsymbol{P}$ by $\boldsymbol{u}$. */
11:      else
12:         if $\boldsymbol{f}^U(\boldsymbol{x}_i) \not\succ \boldsymbol{f}^U(\boldsymbol{u})$ then
13:            $\boldsymbol{P} := \boldsymbol{P} \cup \{\boldsymbol{u}\}$;   /* Add $\boldsymbol{u}$ to $\boldsymbol{P}$. Thus, $|\boldsymbol{P}| > N_P$ holds. */
14:         end if
15:      end if
16:   end for
17:   $\boldsymbol{P}$ := TRUNCATION_METHOD#1($\boldsymbol{P}$, $N_P$);   /* $|\boldsymbol{P}| = N_P$ holds. */
18: until a termination condition is satisfied;
19: Output the non-dominated solution set $\check{\boldsymbol{P}} \subseteq \boldsymbol{P}$;
```

is compared to the target vector $\boldsymbol{x}_i \in \boldsymbol{P}$. If $\boldsymbol{f}^U(\boldsymbol{u})$ weakly dominates $\boldsymbol{f}^U(\boldsymbol{x}_i)$, $\boldsymbol{u}$ replaces $\boldsymbol{x}_i$. However, when they are non-dominated each other, $\boldsymbol{u}$ is added to $\boldsymbol{P}$. Otherwise, $\boldsymbol{u}$ is discarded. As a result, if $\boldsymbol{u}$ survives, the control parameters $S_F$ and $C_R$ used for $\boldsymbol{u}$ are assigned to the new solution $\boldsymbol{u} \in \boldsymbol{P}$. The number of solutions in $\boldsymbol{P}$ becomes $N_P \le |\boldsymbol{P}| \le 2\,N_P$ at the 17th line. In order to return the population size to $N_P$, the following truncation method is applied to $\boldsymbol{P}$.

**[truncation method #1]**

**Step 1** Decide the non-domination rank [18] for each solution $\boldsymbol{x}_i \in \boldsymbol{P}$ and then select $N_P$ solutions from $\boldsymbol{P}$ in the ascending order on the rank.

**Step 2** If some solutions need to be selected from $\boldsymbol{P}_r \subseteq \boldsymbol{P}$ with the same rank, evaluate $\epsilon$-DOM criterion [21] for $\boldsymbol{x}_i \in \boldsymbol{P}_r$. Thereafter, select the necessary number of solutions from $\boldsymbol{P}_r$ in the descending order on the criterion.

For sorting non-dominated solutions, some secondary criteria that can replace the crowding-distance [18] have been reported. From the result of comparative study, $\epsilon$-DOM was the best in the average among examined secondary criteria [21]. Therefore, $\epsilon$-DOM is adopted in Step 2 of the truncation method #1.

## 5 Proposed Approach to MNOP

Multiple sampling of every examined solution is very expensive in most of real-world optimization problems. To allocate the computing budget of DEMO only to promising solutions of MNOP, we propose two novel pruning techniques of hopeless solutions, which are called U-cut and C-cut respectively. First of all, we restrict the value of each $f_m^U(\boldsymbol{x})$ in (8) to be less than $\gamma_m \in \Re$ because

1. in real-world applications, every solution has to meet absolute standards,
2. a part of the Pareto-front is usually sufficient for decision making,
3. expensive evaluation may be omitted for unacceptable solutions.

Let $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_M\} \in \Re^M$ be a cutoff point specified by the designer. A Multi-Noisy-Hard-objective Optimization Problem (MNHOP) is formulated as

$$\left[\begin{array}{l} \text{minimize} \quad \boldsymbol{f}^U(\boldsymbol{x}) = (f_1^U(\boldsymbol{x}), \cdots, f_m^U(\boldsymbol{x}), \cdots, f_M^U(\boldsymbol{x})), \\ \text{subject to} \ \ (\boldsymbol{x} \in \boldsymbol{X}) \wedge (\boldsymbol{f}^U(\boldsymbol{x}) \succeq \boldsymbol{\gamma}), \end{array}\right. \tag{18}$$

where a solution $\boldsymbol{x} \in \boldsymbol{X}$ is feasible if the solution satisfies all constraints. The feasible space $\boldsymbol{G} \subseteq \boldsymbol{F}$ is defined as $\boldsymbol{G} = \{\boldsymbol{f}^U(\boldsymbol{x}) \in \boldsymbol{F} \mid \forall m \in \mathcal{I}_M : f_m^U(\boldsymbol{x}) \leq \gamma_m\}$.

**Algorithm 2.** DEUC APPLIED TO MNHOP

```
1: $\boldsymbol{P}$ := GENERATE_INITIAL_POPULATION($N_P$);
2: for $i := 1$ to $N_P$ do
3:   if $\forall n \in \mathcal{I}_N : \boldsymbol{f}^n(\boldsymbol{x}_i) \succeq \boldsymbol{\gamma}$ then
4:     $\boldsymbol{g}(\boldsymbol{x}_i) := (\boldsymbol{f}^U(\boldsymbol{x}_i) :=$ PREDICT_UPPER_BOUND($\boldsymbol{f}^n(\boldsymbol{x}_i),\ n \in \mathcal{I}_N$));
5:   else
6:     $\boldsymbol{g}(\boldsymbol{x}_i) := \boldsymbol{f}^{\hat{n}}(\boldsymbol{x}_i)$;   /* $\exists \hat{n} \in \mathcal{I}_N : \boldsymbol{f}^{\hat{n}}(\boldsymbol{x}_i) \not\succeq \boldsymbol{\gamma}$ */
7:   end if
8: end for
9: repeat
10:   for $i := 1$ to $N_P$ do
11:     $\boldsymbol{u}$ := STRATEGY($\boldsymbol{x}_i \in \boldsymbol{P}$);   /* Generate a new trial vector $\boldsymbol{u} \in \boldsymbol{X}$ */
12:     if $\forall n \in \mathcal{I}_N : (\boldsymbol{g}(\boldsymbol{x}_i) \not\succ \boldsymbol{f}^n(\boldsymbol{u})) \wedge (\boldsymbol{f}^n(\boldsymbol{u}) \succeq \boldsymbol{\gamma})$ then
13:       $\boldsymbol{g}(\boldsymbol{u}) := (\boldsymbol{f}^U(\boldsymbol{u}) :=$ PREDICT_UPPER_BOUND($\boldsymbol{f}^n(\boldsymbol{u}),\ n \in \mathcal{I}_N$));
14:     else
15:       $\boldsymbol{g}(\boldsymbol{u}) := \boldsymbol{f}^{\hat{n}}(\boldsymbol{u})$;   /* $\exists \hat{n} \in \mathcal{I}_N : (\boldsymbol{g}(\boldsymbol{x}_i) \succ \boldsymbol{f}^{\hat{n}}(\boldsymbol{u})) \vee (\boldsymbol{f}^{\hat{n}}(\boldsymbol{u}) \not\succeq \boldsymbol{\gamma})$ */
16:     end if
17:     if $\boldsymbol{g}(\boldsymbol{u}) \succeq \boldsymbol{g}(\boldsymbol{x}_i)$ then
18:       $\boldsymbol{x}_i := \boldsymbol{u}$;   /* Replace $\boldsymbol{x}_i \in \boldsymbol{P}$ by $\boldsymbol{u}$. */
19:     else
20:       if $\boldsymbol{g}(\boldsymbol{x}_i) \not\succ \boldsymbol{g}(\boldsymbol{u})$ then
21:         $\boldsymbol{P} := \boldsymbol{P} \cup \{\boldsymbol{u}\}$;   /* Add $\boldsymbol{u}$ to $\boldsymbol{P}$. Thus, $|\boldsymbol{P}| > N_P$ holds. */
22:       end if
23:     end if
24:   end for
25:   $\boldsymbol{P}$ := TRUNCATION_METHOD#2($\boldsymbol{P}$, $N_P$, $\eta$);   /* $|\boldsymbol{P}| = N_P$ holds. */
26: until a termination condition is satisfied;
27: Output the non-dominated feasible solution set $\check{\boldsymbol{Q}} \subseteq \boldsymbol{Q} \subseteq \boldsymbol{P}$;
```

Differential Evolution with U-cut & C-cut (DEUC) is an extended DEMO and applied to MNHOP in (18) instead of MNOP in (8). Algorithm 2 provides the pseudo-code of DEUC. The proposed DEUC evaluates solutions $\boldsymbol{x}_i \in \boldsymbol{P}$ by the fitness vectors $\boldsymbol{g}(\boldsymbol{x}_i) \in \Re^M$ instead of the objective vectors $\boldsymbol{f}^U(\boldsymbol{x}_i) \in \Re^M$. The fitness vectors are initialized for $\boldsymbol{x}_i \in \boldsymbol{P}$, $i = 1, \cdots, N_P$ in lines 2-8 as

$$\boldsymbol{g}(\boldsymbol{x}_i) = \begin{cases} \boldsymbol{f}^U(\boldsymbol{x}_i) = (f_1^U(\boldsymbol{x}_i), \cdots, f_M^U(\boldsymbol{x}_i)), \text{ if } \forall n \in \mathcal{I}_N : \boldsymbol{f}^n(\boldsymbol{x}_i) \succeq \boldsymbol{\gamma}, \\ \boldsymbol{f}^{\hat{n}}(\boldsymbol{x}_i) = (f_1^{\hat{n}}(\boldsymbol{x}_i), \cdots, f_M^{\hat{n}}(\boldsymbol{x}_i)), \text{ otherwise,} \end{cases} \tag{19}$$

where $(\forall n \in \{1, \cdots, \hat{n}-1\} \subset \mathcal{I}_N : \boldsymbol{f}^n(\boldsymbol{x}_i) \succeq \boldsymbol{\gamma}) \ \wedge \ (\boldsymbol{f}^{\hat{n}}(\boldsymbol{x}_i) \not\succeq \boldsymbol{\gamma})$ holds.

DEUP generates the trial vector $\boldsymbol{u} \in \boldsymbol{X}$ at the 11th line in the same way with DEMO in Algorithm 1. The fitness vector $\boldsymbol{g}(\boldsymbol{u})$ of $\boldsymbol{u}$ is evaluated in lines 12-16. From (11), if $\boldsymbol{g}(\boldsymbol{x}_i)$ dominates $\boldsymbol{f}^{\hat{n}}(\boldsymbol{u})$, $\boldsymbol{g}(\boldsymbol{x}_i)$ also dominates $\boldsymbol{f}^U(\boldsymbol{u})$ in short odds. Therefore, the U-cut based on the upper bounds of objective functions skips additional sampling of $\boldsymbol{u}$ and set $\boldsymbol{f}^{\hat{n}}(\boldsymbol{u})$ to $\boldsymbol{g}(\boldsymbol{u})$. On the other hand, if $\boldsymbol{f}^{\hat{n}}(\boldsymbol{u})$ doesn't dominate $\boldsymbol{\gamma} \in \Re^M$, $\boldsymbol{u}$ is probably infeasible. Therefore, the C-cut based on the cut -off point also skips additional sampling of $\boldsymbol{u}$ and set $\boldsymbol{f}^{\hat{n}}(\boldsymbol{u})$ to $\boldsymbol{g}(\boldsymbol{u})$. The objective vector $\boldsymbol{f}^U(\boldsymbol{u})$ is evaluated and substituted for $\boldsymbol{g}(\boldsymbol{u})$ at the 13th line only if $\boldsymbol{u}$ is feasible and $\boldsymbol{g}(\boldsymbol{x}_i)$ doesn't dominate every $\boldsymbol{f}^n(\boldsymbol{u})$, $n \in \mathcal{I}_N$.

The trial vector $\boldsymbol{u}$ is compared to the target vector $\boldsymbol{x}_i \in \boldsymbol{P}$ in lines 17-23 based on their fitness vectors. The feasibility of solutions doesn't need to be considered in the comparison between $\boldsymbol{u}$ and $\boldsymbol{x}_i \in \boldsymbol{P}$, because it is proven that feasible solutions $\boldsymbol{f}^U(\boldsymbol{x}) \in \boldsymbol{G}$ are not dominated by any infeasible ones [22]:

$$\boldsymbol{f}^U(\boldsymbol{x}) \in \boldsymbol{F} \ \wedge \ \boldsymbol{f}^U(\boldsymbol{x}') \in \boldsymbol{G} \ \wedge \ \boldsymbol{f}^U(\boldsymbol{x}) \succeq \boldsymbol{f}^U(\boldsymbol{x}') \Rightarrow \boldsymbol{f}^U(\boldsymbol{x}) \in \boldsymbol{G}. \tag{20}$$

In order to return the population size to $N_P$ at the 25th line, the following truncation method #2 is applied to $\boldsymbol{P}$. The truncation method #2 was proposed for multi-hard-objective optimization problems in our previous paper [22]. Hard-objective differs from constrained objective because the former has no conflict with its constraint. If an objective function $f_m^U(\boldsymbol{x})$ is minimized in MNHOP, its constraint $f_m^U(\boldsymbol{x}) \leq \gamma_m$ will be satisfied sooner or later. Let $\boldsymbol{Q} \subseteq \boldsymbol{P}$ be a set of feasible solutions defined as $\boldsymbol{Q} = \{\boldsymbol{x}_i \in \boldsymbol{P} \mid \boldsymbol{f}^U(\boldsymbol{x}_i) \in \boldsymbol{G}\}$. Feasible solutions $\boldsymbol{x}_i \in \boldsymbol{Q}$ have priority over infeasible ones in $\boldsymbol{P}$. For sorting infeasible solutions, alternative schemes are chosen by a control parameter $\eta$ $(0 \leq \eta \leq 1)$.

**[truncation method #2]**

**Step 1** If $|\boldsymbol{Q}| \geq N_P$ then apply truncation method #1 to $\boldsymbol{Q} \subseteq \boldsymbol{P}$.
**Step 2** If $|\boldsymbol{Q}| < N_P$ then select all feasible solutions $\boldsymbol{x}_i \in \boldsymbol{Q}$. Thereafter, the shortage is selected from the set of infeasible solutions $\boldsymbol{Q}^c = \boldsymbol{P} \setminus \boldsymbol{Q}$ as
**Step 2.1** If $|\boldsymbol{Q}| \leq \eta\, N_P$ then apply truncation method #1 to $\boldsymbol{Q}^c \subseteq \boldsymbol{P}$.
**Step 2.2** Otherwise, select the necessary number of solutions $\boldsymbol{x}_i \in \boldsymbol{Q}^c \subseteq \boldsymbol{P}$ in the ascending order on the violation distance $d(\boldsymbol{x}_i) \in \Re$ defined as

$$d(\boldsymbol{x}_i) = \sum_{m=1}^{M} \max\{0, \ (g_m(\boldsymbol{x}_i) - \gamma_m)\}. \tag{21}$$

## 6 Numerical Experiments

### 6.1 Experimental Setup

In most real-world MNOPs, the higher objective function values are usually expected to have more errors than lower ones [10]. Therefore, by using a deterministic function $f_m(\boldsymbol{x}) \in \Re$, the noisy objective function is defined as

**Table 1.** Number of obtained solutions

(a) DEMO applied to MNOP

| $M$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| DTLZ1 | 85.7 | 99.9 | 100.0 | 100.0 |
| DTLZ2 | 98.3 | 100.0 | 100.0 | 100.0 |
| DTLZ3 | 98.7 | 100.0 | 100.0 | 100.0 |
| DTLZ4 | 98.5 | 100.0 | 100.0 | 100.0 |
| DTLZ5 | 98.3 | 100.0 | 100.0 | 100.0 |
| DTLZ6 | 79.7 | 100.0 | 100.0 | 100.0 |

(b) DEUC applied to MNHOP

| $M$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| DTLZ1 | 92.0 | 100.0 | 100.0 | 76.6 |
| DTLZ2 | 99.0 | 100.0 | 100.0 | 100.0 |
| DTLZ3 | 99.4 | 100.0 | 100.0 | 63.3 |
| DTLZ4 | 98.7 | 100.0 | 100.0 | 100.0 |
| DTLZ5 | 99.0 | 100.0 | 100.0 | 100.0 |
| DTLZ6 | 95.1 | 100.0 | 100.0 | 100.0 |

**Table 2.** Comparison of DEMO and DEPC by Wilcoxon test

(a) Convergence (CM)

| $M$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| DTLZ1 | △ | △ | △ | △ |
| DTLZ2 | — | △ | △ | △ |
| DTLZ3 | △ | △ | △ | △ |
| DTLZ4 | ▲ | △ | △ | △ |
| DTLZ5 | — | △ | ▲ | △ |
| DTLZ6 | △ | △ | △ | △ |

(b) Diversity (MS)

| $M$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| DTLZ1 | ▽ | ▽ | ▽ | ▽ |
| DTLZ2 | — | ▽ | ▽ | ▽ |
| DTLZ3 | ▽ | ▽ | ▽ | ▽ |
| DTLZ4 | — | ▽ | ▽ | — |
| DTLZ5 | — | ▽ | ▽ | ▽ |
| DTLZ6 | ▽ | ▽ | ▽ | ▽ |

(c) Hypervolume (Hv)

| $M$ | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| DTLZ1 | △ | △ | △ | △ |
| DTLZ2 | — | — | — | — |
| DTLZ3 | △ | △ | △ | △ |
| DTLZ4 | — | — | — | — |
| DTLZ5 | — | △ | — | — |
| DTLZ6 | △ | △ | △ | △ |

$$f_m^n(\boldsymbol{x}) = f_m(\boldsymbol{x}) + \lambda_m\, f_m(\boldsymbol{x})\, \varepsilon_m^a + \kappa_m\, \varepsilon_m^b, \tag{22}$$

where $\varepsilon_m^a \sim \mathcal{N}(0,\ 1)$, $\varepsilon_m^b \sim \mathcal{N}(0,\ 1)$, $\lambda_m > 0$, and $\kappa_m > 0$.

According to a model of noise [10], the noise in (22) is also composed of two components: variable one $\lambda_m\, f_m(\boldsymbol{x})$ and constant one $\kappa_m$. From the reproductive property of the normal distribution [16], $f_m^n(\boldsymbol{x})$ is distributed normally as

$$f_m^n(\boldsymbol{x}) \sim \mathcal{N}(f_m(\boldsymbol{x}),\ \sigma_m(\boldsymbol{x})^2) = \mathcal{N}(f_m(\boldsymbol{x}),\ \lambda_m^2\, f_m(\boldsymbol{x})^2 + \kappa_m^2). \tag{23}$$

The scalable test MOPs [23] with $M$ objectives are employed for providing $f_m(\boldsymbol{x})$ in (22) with $\lambda_m = 0.01$ and $\kappa_m = 0.05$, $m \in \mathcal{I}_M$. From (15), the minimum sample size $N = 40$ is calculated for $\alpha = 0.05$ and $\delta = 2.5$. DEMO is applied to each instance of MNOP in (8) 30 times. The population size is chosen as $N_P = 100$. As the termination condition, the total number of function evaluations is limited to $8 \times 10^5$. Similarly, DEUC is applied to each instance of MNHOP in (18), where a cutoff point $\boldsymbol{\gamma} \in \Re^M$ is given for all cases as $\gamma_m = 2.0$, $m \in \mathcal{I}_M$. A recommended value $\eta = 0.2$ [22] is used for the truncation method #2.

### 6.2 Results and Discussion

Table 1 compares the average numbers of solutions obtained by DEMO and DEUC. Because the solutions obtained by DEUC have to be feasible, DEUC finds fewer solutions than DEMO in two cases: DTLZ1 and DTLZ3 ($M = 8$).

To evaluate the solutions in Table 1, we use three metrics: 1) Convergence Measure (CM) of the original test MOPs [23], 2) Maximum Spread (MS) [24], and 3) Hypervolume (Hv). MS is a metric to evaluate the diversity of solutions. Hv is a comprehensive metric evaluating both convergence and diversity.

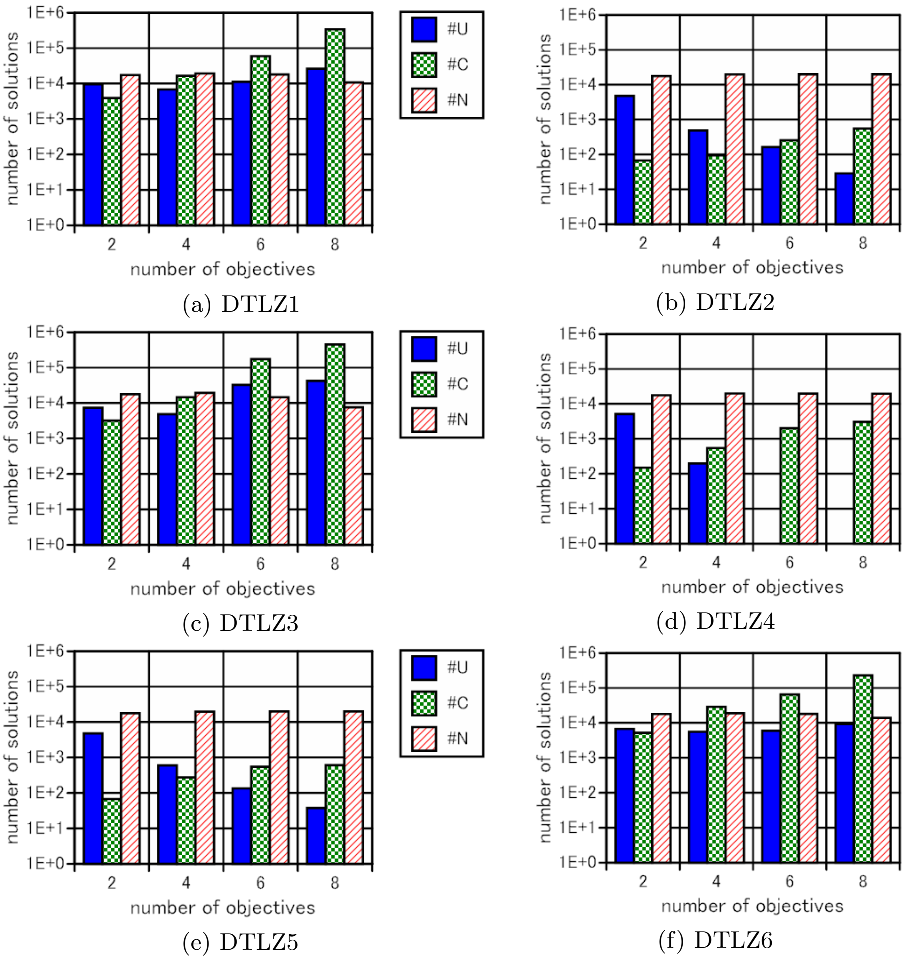

**Fig. 1.** Histogram of the number of solutions examined by DEUC

Table 2 compares DEUC with DEMO by using Wilcoxon test about CM, MS, and Hv: △ (▽) means that DEUC is significantly better (worse) than DEMO with risk 1[%]; ▲ (▼) means that DEUC is better (worse) than DEMO with risk 5[%]; and "—" means that there is no significant difference between DEUC and DEMO. From Table 2, DEUC is not defeated by DEMO in CM for all cases. On the other hand, DEUC can't beat DEMO in MS. Comparing to DEMO in Hv, DEUC is not defeated for all cases and significantly better for many cases.

From Table 2(c), DEMO is competitive with DEUC in DTLZ2 and DTLZ4. However, DEUC defeats DEMO in DTLZ1, DTLZ3, and DTLZ6. Fig. 1 shows the histogram of the number of solutions examined by DEUC in each problem. Category #N in Fig. 1 denotes the number of solutions evaluated $N = 40$ times. Categories #U and #C denote the numbers of solutions evaluated less than $N = 40$ times due to U-cut and C-cut respectively, where DEUC applies U-cut to each solution before C-cut. Because DEMO evaluates every solution $N = 40$ times, the number of examined solutions is always $(8 \times 10^5)/40 = 2 \times 10^4$. Contrarily, DEUC examines more solutions than DEMO as shown in Fig. 1.

From Fig. 1, both U-cut and C-cut work effectively in DTLZ1, DTLZ3, and DTLZ6. Especially, C-cut becomes more effective as the number of objective functions increases. Because a lot of hopeless solutions are evaluated only a few times due to U-cut and C-cut, the total numbers of solutions examined by DEUC become very large in those problems. On the other hand, neither U-cut nor C-cut is effective for the other problems. Especially, U-put doesn't work when the number of objective functions is large. Consequently, the total numbers of examined solutions don't increase so much in DTLZ2, DTLZ4, and DTLZ5.

## 7 Conclusion

It is important to consider the worst-case performance for real-world MNOPs. Therefore, we have predicted statistically the upper bounds of noisy objective functions from a finite number of samples. In order to omit useless multiple sampling, we have also proposed an extended DEMO named DEUC that uses two pruning techniques of hopeless solutions. DEUC was not defeated by DEMO in all test problems. Besides, DEUC outperformed DEMO in many test problems. Even though DEUC requires a new control parameter, namely the cutoff point, an appropriate cutoff point can be decided easily for real-world MNOPs from specifications or an existing solution. Actually, a sufficient improvement over an existing solution should be acceptable in real-world applications.

Future work will include an in-depth evaluation of the proposed DEUP on a broad range of practical MNHOPs with various cutoff points. Furthermore, handling non-Gaussian noise efficiently remains as an active area of research.

**Acknowledgment.** This work was supported by a grant from Japan Society for the Promotion of Science (JSPS) (Project No. 24560503).

## References

1. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution - A Practical Approach to Global Optimization. Springer (2005)
2. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments - a survey. IEEE Trans. on Evolutionary Computation 9(3), 303–317 (2005)
3. Gunawan, S., Azarm, S.: Multi-objective robust optimization using a sensitivity region concept. Structural and Multidisciplinary Optimization 29(1), 50–60 (2005)
4. Voß, T., Trautmann, H., Igel, C.: New uncertainty handling strategies in multi-objective evolutionary optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 260–269. Springer, Heidelberg (2010)
5. Rakshit, P., Konar, A., Das, S., Jain, L.C., Nagar, A.K.: Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise. IEEE Trans. on Systems, Man, and Cybernetics: Systems 44(7), 922–937 (2013)

6. Hughes, E.J.: Evolutionary multi-objective ranking with uncertainty and noise. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 329–342. Springer, Heidelberg (2001)
7. Fieldsend, J.E., Everson, R.M.: Multi-objective optimization in the presence of uncertainty. In: Proc. IEEE CEC 2005, pp. 243–250 (2005)
8. Shim, V.A., Tan, K.C., Chia, J.Y., Mamun, A.A.: Multi-objective optimization with estimation of distribution algorithm in a noisy environment. Evolutionary Computation 21(1), 149–177 (2013)
9. Bui, L.T., Abbass, H.A., Essam, D.: Localization for solving noisy multi-objective optimization problems. Evolutionary Computation 17(3), 379–409 (2009)
10. Eskandari, H., Geiger, C.D.: Evolutionary multiobjective optimization in noisy problem environments. Journal of Heuristics 15(6), 559–595 (2009)
11. Teich, J.: Pareto-front exploration with uncertain objectives. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 314–328. Springer, Heidelberg (2001)
12. Kuroiwa, D., Lee, G.M.: On robust multiobjective optimization. Vietnam Journal of Mathematics 40(2&3), 305–317 (2012)
13. Avigad, G., Branke, J.: Embedded evolutionary multi-objective optimization for worst case robustness. In: Proc. GECCO 2008, pp. 617–624 (2008)
14. Branke, J., Avigad, G., Moshaiov, A.: Multi-objective worst case optimization by means of evolutionary algorithms. Working Paper, Coventry UK: WBS, University of Warwick (2013), http://wrap.warwick.ac.uk/55724
15. Ehrgott, M., Ide, J., Schöbel, A.: Minmax robustness for multi-objective optimization problems. European Journal of Operation Research (2014), http://dx.doi.org/10.1016/j.ejor.2014.03.013
16. Wackerly, D.D., Mendenhall, W., Scheaffer, R.L.: Mathematical Statistics with Applications, 7th edn. Thomson Learning, Inc. (2008)
17. Fisher, R.A.: On the interpretation of $\chi^2$ from contingency tables, and calculation of $\mathcal{P}$. Journal of the Royal Statistical Society 85(1), 87–94 (1922)
18. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(2), 182–197 (2002)
19. Robič, T., Filipič, B.: DEMO: Differential evolution for multiobjective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 520–533. Springer, Heidelberg (2005)
20. Brest, J., Greiner, S., Bošković, B., Merink, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. on Evolutionary Computation 10(6), 646–657 (2006)
21. Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)
22. Tagawa, K., Imamura, A.: Many-hard-objective optimization using differential evolution based on two-stage constraint-handling. In: Proc. GECCO 2013, pp. 671–678 (2013)
23. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. TIK-Technical Report, 112, 1–27 (2001)
24. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, PhD thesis, Swiss Federal Institute of Technology, Zurich (1999)