Evolutionary multi-level robust solution search for noisy multi-objective optimisation problems with different noise levels

Hiroyuki Sato* and Tomohisa Hashimoto

Graduate School of Information and Engineering Sciences,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
Email: sato@hc.uec.ac.jp
Email: hashimoto@hs.hc.uec.ac.jp
*Corresponding author

Abstract: For noisy multi-objective optimisation problems involving multiple noisy objective functions with different noise levels, this work proposes a multiobjective evolutionary algorithm for multi-level robust solution search (MOEAMRS). MOEA-MRS simultaneously finds multi-level robust solutions with different noise levels for each search direction in the objective space. Furthermore, as an extension of MOEA-MRS, we also propose a MOEA for preference-based multi-level robust solution search (MOEA-pMRS) which focuses the solutions search on a specific noise level to consider the case that the decision maker has a preference for the noise level. The experimental results using noisy DTLZ2 and multi-objective knapsack problems shows that the proposed MOEA-MRS is able to obtain multi-level robust solutions with different noise levels for each search direction in a single run of the algorithm, and the proposed MOEA-pMRS can emphasise the solution search for specific noise levels.

1 Introduction

Evolutionary algorithms are particularly suited to solve multi-objective optimisation problems (MOPs) involving multiple objective functions since Pareto optimal solutions (POS) approximating the optimal trade-off among objectives can be obtained from the population in a single run (Deb, 2001; Coello, 2007). So far, multi-objective evolutionary algorithms (MOEAs) have been intensively studied for solving noise-free MOPs that objective values of each solution are uniquely determined by one-time evaluation. However, in some MOPs such as real-world engineering optimisation problems, objective values of each solution are not uniquely determined, and they are varied in every evaluation due to the influence of noise (Hughes, 2001a; Goh and Tan, 2009). This is because, several unknown factors not considered as decision variables affect the objective values. For these noisy multi-objective optimisation problems (NMOPs), several algorithms optimising solutions based on the estimation of the true objective function values by considering the influence of noise have been studied so far (Babbar et al., 2003; Teich, 2001; Basseur and Zitzler, 2006; Bui et al., 2005; Buche et al., 2002; Goh and Tan, 2009). However, in conventional approaches, the noise level of each solution cannot be considered in the decision-making process when a decision maker tries to select the final solution from the obtained solutions. If the noise level of each solution can be considered in the decision-making process, the number of valuable choices is increased for the decision maker. Meanwhile, the decision maker will be able to select a solution by considering not only objective values but also the noise level.

Figure 1 Two-stage MCDM considering the balance among objectives and the noise levels of solutions, (a) 1st stage: make a decision on the balance among objectives (b) 2nd stage: make a decision on the robustness of solution (see online version for colours)

For NMOPs, we aim to develop a multi-criteria decision-making (MCDM) system considering not only the objective values but also the noise level of solutions. Figure 1 shows an overview of the MCDM system. In this system, the decision-making process is divided into two stages. In the first stage, the decision maker selects a solution with a

preferred balance of objective values from the obtained POS without considering the noise level in the objective space. In the second stage, the decision maker selects a solution based on the noise levels of solutions. For the preferred balance of objective values, this system shows solutions with different levels of the noise and guides the decision-making considering the noise level of solutions. In Figure 1, the size of circle around each solution indicates the noise level. Thus, the two-stage MCDM system allows the decision maker to select the final solution by considering not only the objective values but also the noise levels.

For the two-stage MCDM system considering the noise, in this work we propose a MOEA for multi-level robust solution search (MOEA-MRS) which simultaneously searches multi-level robust solutions with different noise levels for each search direction in the multidimensional objective space. The proposed MOEA-MRS is designed as a variant of the conventional MOEA/D (Zhang and Li, 2007) which is a representative MOEA for solving MOPs without the noise. The proposed MOEA-MRS tries to find multi-level robust solutions for a number of search directions in a single run of the algorithm. Furthermore, as an extension of MOEA-MRS, we also propose a MOEA for preference-based multi-level robust solution search (MOEA-pMRS) which focuses the solutions search on a specific noise level to consider the case that the decision maker has a preference for the noise level. In this work we extend the conventional DTLZ2 problem (Deb et al., 2002) and multi-objective knapsack problem (Zitzler and Thiele, 1999) to noisy problems with different noise levels, and use them to verify the effectiveness of the proposed MOEA-MRS and MOEA-pMRS. This paper is the extended version of our paper (Hashimoto and Sato, 2015) presented at IES2015 conference for the special issue, and additionally includes MOEA-pMRS employing the preference-based robust solution search and its effectiveness verification in the noisy DTLZ2 problem.

2   Noisy multi-objective optimisation using evolutionary algorithms

2.1   Noisy multi-objective optimisation problem

A NMOP is defined by

$$\text{Minimise (or maximise) } \boldsymbol{f}^z(\boldsymbol{x}) = \left\{ f_1^z(\boldsymbol{x}), f_2^z(\boldsymbol{x}), \ldots, f_m^z(\boldsymbol{x}) \right\}, \tag{1}$$

where $\boldsymbol{x}$ is the design variable vector, the solution, consists of n kinds of variables ($\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$), and $f_j^z (j = 1, 2, \ldots, m)$ are m kinds of noisy objective functions. The objective values $f_j^z(\boldsymbol{x})(j = 1, 2, \ldots, m)$ including noise are varied in every evaluation of the solution $\boldsymbol{x}$ due to the influence of the probabilistic noise.

For the noisy objective $f_j^z (j = 1, 2, \ldots, m)$, there are several formulations.

2.2   Three types of noisy objective functions

In the noisy multi-objective optimisation using evolutionary algorithms, two types of noisy objective functions have been studied so far (Hughes, 2001a; Goh and Tan, 2009). According to the categorisation of noisy objective functions in the article (Hughes, 2001a), in this paper they are described as *Type A* and *B*, respectively.

Additionally, in this work we define another *Type C* noisy objective function and focus on it. In the following, the three types of noisy objective functions are introduced.

- *Type A:* is the noisy objective function that probabilistic noise values $z = \{z_1, z_2, \ldots, z_n\}$ vary each solution vector $x = \{x_1, x_2, \ldots, x_n\}$ before the original objective functions f_j ($j = 1, 2, \ldots, m$) are calculated (Hughes, 2001a, 2001b; Basseur and Zitzler, 2006). This type of noisy objective function is formulated by

$$f_j^z(x) = f_j(x + z) \quad (j = 1, 2, \ldots, m). \tag{2}$$

  Note that the input for each original objective function f_j is $x + z$ ($= \{x_1 + z_1, x_2 + z_2, \ldots, x_n + z_n\}$) which is varied due to the influence of noise z. That is, in NMOPs with this type of noisy objective functions, the aim is to find robust solutions showing good objective function values even if their variables are slightly varied in the decision variable space due to the probabilistic noise z. Recently, evolutionary multi-objective optimisation is increasingly applied to engineering optimisation problems (Coello and Lamont, 2004). Even though we can obtain an ideal design plan as the result of an engineering optimisation, some errors usually occur in its production. To achieve good product performance on the assumption that some production errors occur, this type of function formulation is available.

- *Type B*: is the noisy objective function that probabilistic noise values $z = \{z_1, z_2, \ldots, z_m\}$ vary the original objective function values $f(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\}$ after the calculation of the original objective functions f (Hughes, 2001a, 2001b; Buche et al., 2002; Babbar et al., 2003; Basseur and Zitzler, 2006; Bui et al., 2005; Park and Ryu, 2011). This type of noisy objective function is formulated by

$$f_j^z(x) = f_j(x) + z_j \quad (j = 1, 2, \ldots, m). \tag{3}$$

  Note that the input for each original objective function f_j is fixed x, and the varying noise z_j is added to the original function value $f_j(x)$ ($j = 1, 2, \ldots, m$). In NMOPs with this type of noisy objective functions, the aim is to find solutions showing good original objective function values f_j ($j = 1, 2, \ldots, m$) while considering the influence of noise z. As an example in engineering optimisation, this type of function is the case that product performance values are varied in every evaluation due to the influence of unknown noise factors even the same product design is used for every evaluation.

- *Type C:* is the noisy objective function that probabilistic noise function values $z(x) = \{z_1(x), z_2(x), \ldots, z_m(x)\}$ vary the original objective function values $f(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\}$ after the calculation of the original objective functions f.

$$f_j^z(x) = f_j(x) + z_j(x) \quad (j = 1, 2, \ldots, m). \tag{4}$$

Compared with *Type B*, note that the last noise term is a function of x. That is, *Type C* is the function that the noise level (amount) is determined by the design variable vector x. For example, in *Type C*, a design variable x showing a large $z(x)$ has a large influence of noise for the objective values f^z. In contrast, a design variable x showing a small $z(x)$ has

a small influence of noise for the objective values $\boldsymbol{f}^{z}$. In this work, we focus on this type of noisy objective function which has a trade-off between the noise level and the objective function values. For example, here we consider two design variable vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Although the expected objective function values of solution $\boldsymbol{a}$ are better than the ones of $\boldsymbol{b}$, $\boldsymbol{a}$ has larger noise than $\boldsymbol{b}$. In contrast, although solution $\boldsymbol{b}$ has smaller noise than $\boldsymbol{a}$, the expected objective function values of $\boldsymbol{b}$ are worse than the ones of $\boldsymbol{a}$. In this case, we cannot determine inferiority or superiority of these two design variable vectors. The decision maker might prefer a solution with a low influence of noise even its expected objective values are bad, or prefer a solution with good expected objective values even its influence of noise is high. Therefore, in NMOPs with this type of noisy objective functions, the aim is to find POSs for multi noise levels. After the optimisation, the decision maker selects one final solution from the obtained solutions by considering both the expected objective values and the noise levels of solutions.

2.3   *Related works and scope of this work*

To solve NMOPs with *Type A* and *B* noisy objective functions by using evolutionary algorithms, several noise-handlings have been proposed so far (Goh and Tan, 2009). For each solution $\boldsymbol{x}$, most of algorithms use multiple samples of the objective values $\boldsymbol{f}^{z}(\boldsymbol{x})$ including noise by repeating function evaluation of $\boldsymbol{f}^{z}$. In one approach to utilise the multiple samples of $\boldsymbol{f}^{z}(\boldsymbol{x})$, the average objective vector of the multiple samples of $\boldsymbol{f}^{z}(\boldsymbol{x})$ is used to determine inferiority or superiority of solutions in the population (Babbar et al., 2003). In another approach, probabilistic rankings of solutions based on several extensions of Pareto dominance have been proposed (Hughes, 2001a, 2001b; Teich, 2001; Park and Ryu, 2011). Also, to extend the idea of the indicator-based MOEA (Zitzler and Künzli, 2004) originally proposed for MOPs without noise, the estimation of the expected indicator value was proposed for solving NMOPs (Basseur and Zitzler, 2006). To estimate the true (original) objective values in noisy multi-objective optimisation, the multiple samples of the objective values are needed but it is time-consuming. To reduce computational time while maintain the estimation quality of the true (original) objective values, the fitness inheritance assigning an offspring a weighted sum of the objective values of parents was introduced for solving NMOPs (Bui et al., 2005). Furthermore, although the archive of non-dominated solutions is generally effective for solving MOPs without noise, the existence of solutions with a large estimation error of objective values in the archive has a negative effect for the solution search in NMOPs. To avoid the negative effect, the idea of life time for solutions in the archive population was introduced (Buche et al., 2002). The life time value of each solution is gradually decreased during the solution search. When the life time value of a solution becomes zero, the solution is discarded from the archive and reevaluated in the population.

The aforementioned conventional MOEAs are designed for solving NMOPs with *Type A* and *B* noisy objective functions, and their aims are to find solutions showing good objective function values even if their variables are slightly varied in the variable space or solutions showing good original objective function values while considering the influence of noise. As mentioned before, this work focuses on NMOPs with *Type C* noisy objective function. These conventional algorithms cannot be easily applied to NMOPs with *Type C* because these conventional algorithms cannot considers different noise levels. Also, in the decision-making process, the noise levels of solutions cannot be considered when the

decision maker tries to select the final solution from the obtained solutions. For NMOPs with different noise levels, if the noise level of each solution can be considered in the decision-making process, the number of valuable choices is increased for the decision maker. Meanwhile, the decision maker will be able to select the final solution by considering not only the objective values but also the robustness. In this work, we propose a MOEA to consider different noise levels in NMOPs with *Type C* noisy objective function.

3 MOEA/D

In this work, we design a MOEA for NMOPs based on the algorithm framework of MOEA/D (Zhang and Li, 2007). MOEA/D is a representative MOEA for solving MOPs without the noise. In this section, we introduce the conventional MOEA/D. Figure 2(a) shows a conceptual figure of the conventional MOEA/D in an $m = 2$ dimensional minimisation problem.

3.1 Algorithm

MOEA/D decomposes a MOP into a number of single-objective optimisation problems. The single-objective optimisation problems are defined by scalarising functions s using uniformly distributed weight vectors $\mathcal{L} = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$. Each weight vector λ^i determines a search direction in the m dimensional objective space. Each element $\lambda^i_j (j = 1, 2, \ldots, m)$ is one of $\{0/H, 1/H, \ldots, H/H\}$ based on the decomposition parameter H, and $N = C^{m-1}_{H+m-1}$ kinds of weight vectors satisfying $\sum_{j=1}^{m} \lambda^i_j = 1.0$ are used for the solution search. A similar idea was also proposed in the article (Murata et al., 2001). In the following, the algorithm of MOEA/D is briefly described.

Step 1	Initialisation:	
	Step 1-1	Calculate distances between any two weight vectors and find the T-nearest weight vectors to each weight vector. For each $i \in \{1, 2, \ldots, N\}$, set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where $\mathcal{L} = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$. are the T-nearest weight vectors to λ^i.
	Step 1-2	Randomly generate the population $\{x^1, x^2, \ldots, x^N\}$.
Step 2	Solution Search:	
	For each $i \in \{1, 2, \ldots, N\}$, perform the following procedure.	
	Step 2-1	Randomly choose two indices k and l from $B(i)$, and then generate an offspring y from parents x^k and x^l by applying genetic operators.
	Step 2-2	For each index $j \in B(i)$, if $s(y\|\lambda^j)$ is better than $s(x^j\|\lambda^j)$, then the current solution x^j is replaced by the generated offspring y ($x^j = y$).
Step 3	Termination:	
	If the termination criterion is satisfied, then stop and pick POS from the population $\{x^1, x^2, \ldots, x^N\}$ as the output of the optimisation. Otherwise, go to Step 2.	

3.2 Scalarising function

In MOEA/D, there are several scalarising approaches to aggregate *m* kinds of objective function values (Miettinen, 1999; Zhang and Li, 2007; Li et al., 2013). In this work, we employ the weighted Tchebycheff approach. In this approach, there are two function formulations. They are the conventional weighted Tchebycheff (Miettinen, 1999) and the reciprocal weighted Tchebycheff (Li et al., 2013). Since the search points specified by the reciprocal weighted Tchebycheff with the weight vectors in the objective space and the weight vectors in the weight space are matched, in this work we employ the reciprocal weighted Tchebycheff scalarising function. The scalar optimisation problem of the reciprocal weighted Tchebycheff function *s* is defined by

$$\text{Minimise } s\left(\boldsymbol{x}\,|\,\boldsymbol{\lambda}\right) \;=\; \max_{1\le j\le m}\left\{\,|\,f_j(\boldsymbol{x})-I_j\,|\,/\lambda_j\,\right\}, \tag{5}$$

where *I* is the obtained ideal point. In this work, each element I_j ($j = 1, 2,…,m$) is set to the best objective function value f_j in the population. To avoid division by zero, $\lambda_j = 0$ is exceptionally replaced by $\lambda_j = 10^{-6}$ before the calculation of *s*. The reciprocal weighted Tchebycheff approach searches a solution minimising *s* toward *I*. The weighted Tchebycheff approach has an advantage that both convex and concave Pareto front can be approximated.

4 Proposal: MOEA-MRS

4.1 Concept

For the two-stage MCDM considering the noise level, in this work we propose a MOEA for multi-level robust solution search (MOEA-MRS) which simultaneously searches multilevel robust solutions with different noise levels for each search direction in the objective space. The proposed MOEA-MRS is designed as a variant of MOEA/D. Figure 2(b) shows a conceptual figure of the proposed MOEA-MRS. In this figure, the size of circle around each solution indicates the noise level. As shown in Figure 2(a), the conventional MOEA/D only obtain blue solutions since the noise level is not considered. For the two-stage MCDM considering the noise level, as shown in Figure 2(b), the proposed MOEA-MRS tries to simultaneously obtain multiple red solutions with different noise levels for each search direction.

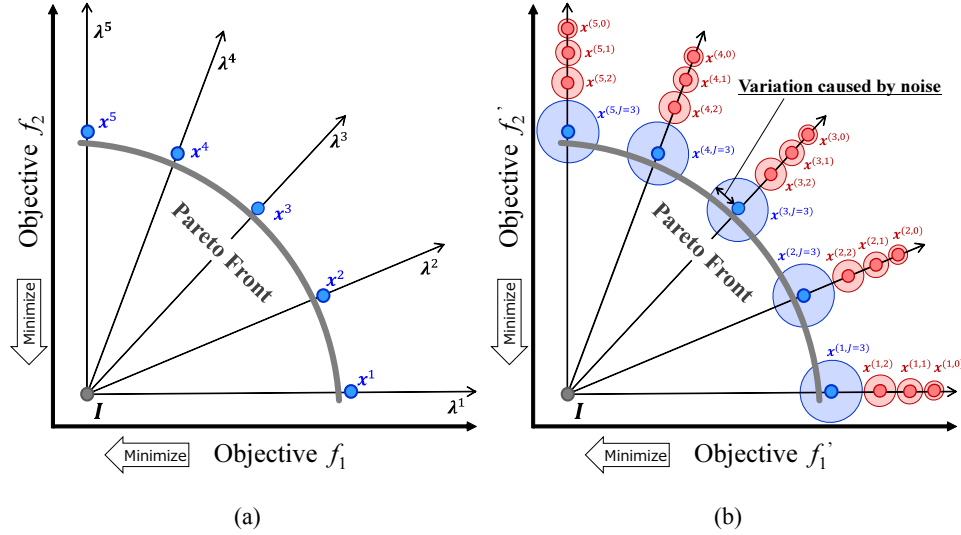### 4.2 *m + 1 fitness values considering noise*

The proposed MOEA-MRS solves a noisy *m* objective optimisation problem as an *m + 1* objective optimisation problem considering not only the optimisation of *m* objectives but also the minimisation of the noise. The proposed MOEA-MRS evaluates a solution based on $m + 1$ kinds of fitness values f'_j ($j = 1, 2, ...,m + 1$). The first *m* fitness values f'_j ($j = 1, 2, ...,m$) are the average objective function values calculated by repeatedly evaluating a solution *r* times. For a solution *x*, the fitness values $f'_j(\boldsymbol{x})$ ($j = 1, 2, ...,m$) are calculated by

$$f_j'(\boldsymbol{x}) = \frac{1}{r} \cdot \sum_{i=1}^{r} f_j^z \quad (j = 1, 2, \ldots, m) \tag{6}$$

The last fitness value f_{m+1}' is the noise level. In this work, we use the total standard deviation. For a solution $\boldsymbol{x}$, the fitness value $f_{m+1}'(\boldsymbol{x})$ is calculated by

$$f_{m+1}'(\boldsymbol{x}) = \sum_{j=1}^{m} \sqrt{\frac{1}{r} \cdot \sum_{i=1}^{r} \left\{ f_j^z(\boldsymbol{x}) - f_j'(\boldsymbol{x}) \right\}^2}. \tag{7}$$

Figure 2 Conceptual figures of the conventional MOEA/D and the proposed MOEA-MRS in a $m = 2$ dimensional minimisation problem, (a) the conventional MOEA/D (b) the proposed MOEA-MRS (see online version for colours)
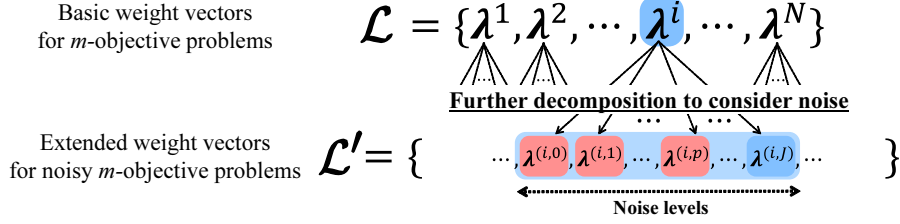

(a)

(b)

4.3 *m + 1 dimensional weight vectors considering noise*

The proposed MOEA-MRS needs $m + 1$ dimensional weight vectors since a NMOP with m objectives is solved as a MOP with $m + 1$ objectives. The proposed MOEA-MRS generates $m + 1$ dimensional weight vectors in a different way from the conventional MOEA/D (Zhang and Li, 2007).

For a NMOP with m objectives, first we generate m dimensional basic weight vectors $\mathcal{L}$ in the same manner as the conventional MOEA/D (Zhang and Li, 2007). The number of elements in each of basic weight vectors $\boldsymbol{\lambda}^i$ ($i = 1, 2, \ldots, N$) is equivalent to the number of objectives m. Next, we generate the extended weight vectors $\mathcal{L}'$ from the basic weight vectors $\mathcal{L}$ as shown in Figure 3. Each extended weight vector in $\mathcal{L}'$ is represented as $\boldsymbol{\lambda}^{(i,p)}$, where i is the basic weight vector index, and p is the extended weight vector index. $\boldsymbol{\lambda}^{(i,p)}$ ($p = 0, 1, \ldots, J$) are weight vectors extended from the basic weight vector $\boldsymbol{\lambda}^i$ to consider the noise in J levels, and the number of elements in $\boldsymbol{\lambda}^{(i,p)}$ becomes $m + 1$. We generate $\boldsymbol{\lambda}^{(i,p)}$ ($p = 0, 1, \ldots, J$) from $\boldsymbol{\lambda}^i$ by the following equation.
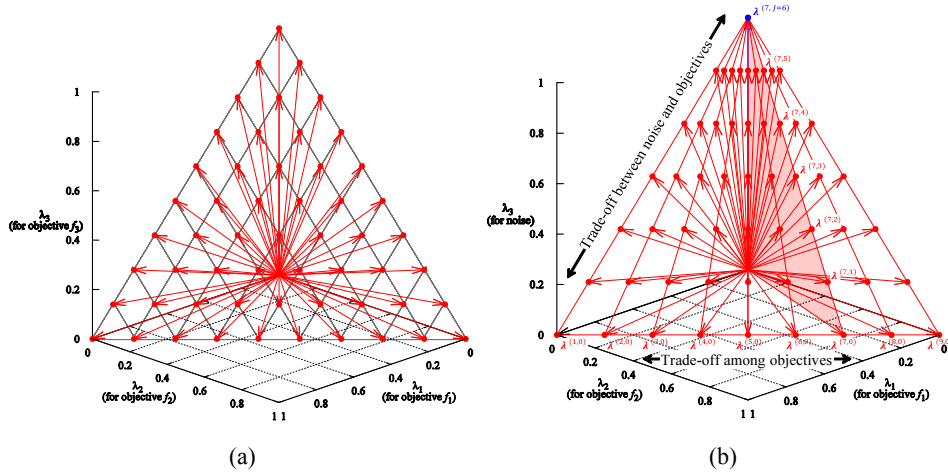
$$\lambda_j^{(i,p)} = \begin{cases} \lambda_j^i \cdot \left(1.0 - \dfrac{p}{J+\delta}\right) & \text{for } j = 1, 2, \ldots, m \\[2ex] \left(\dfrac{p}{J+\delta}\right) & \text{for } j = m+1 \end{cases} \quad (j = 1, 2, \ldots, m, m+1) \tag{8}$$

Figure 3   The basic weights $\mathcal{L}$ and the extended weights $\mathcal{L}'$ in the proposed MOEA-MRS (see online version for colours)

To avoid $\lambda^{(1,J)} = \lambda^{(2,J)} = \cdots = \lambda^{(N,J)}$, we use a very small value δ in equation (8). In this work δ is set to 0.01. In this way, the extended weight vectors $\lambda^{(i,p)}$ ($p = 0, 1, \ldots, J$) consider the additional dimension for the noise while maintaining the balance of m kinds of elements in the basic weight vector λ^i. The number of the basic weight vectors in $\mathcal{L}$ is $N = C_{H+m-1}^{m-1}$ as described in Section 3.1. The number of the extended weight vectors in $\mathcal{L}'$ becomes $N' = N \times (J+1)$ since each of the basic weight vectors is further decomposed into $J+1$ kinds of the extended weight vectors.

Figure 4   Difference of the weights distribution between the conventional MOEA/D considering $m = 3$ objectives and the proposed MOEA-MRS considering $m = 2$ objectives plus one noise, (a) the conventional MOEAD with $H = 9$ (b) the proposed MOEA-MRS with $H = 8$ and $J = 6$ (see online version for colours)

(a)   (b)

Although $m + 1$ dimensional weight vectors can also be generated by the same way of the conventional MOEA/D, the proposed MOEA-MRS generates $m + 1$ dimensional weight vectors by using the above procedure for the two-stage MCDM system shown in Figure 1. Figure 4 shows examples of weight vector distributions in the weight space. $m + 1 = 3$ dimensional weights generated by the conventional MOEA/D are shown in Figure 4(a). Also, $m + 1 = 3$ dimensional weights generated by the proposed MOEA-MRS is shown in Figure 4(b). Thus, weight distributions of the conventional MOEA/D and the proposed MOEA-MRS are different. For the two-stage MCDM, the proposed MOEA-MRS varies the extended weight λ_3 for the noise while maintaining the balance of the elements in the basic weight vector. For example, in Figure 4(b), after the balance of $\lambda^{(7,\cdot)}$ is selected in the first stage of the decision-making process, in the second stage the decision maker can select a solution from solutions obtained with $\lambda^{(7,0)}$, $\lambda^{(7,1)}$,...,$\lambda^{(7,J)}$ while maintaining the balance of the objectives selected in the first stage of the decision-making process.

4.4 Algorithm

The proposed MOEA-MRS is designed as a variant of MOEA/D. The proposed MOEAMRS has two main differences from the conventional MOEA/D.

The first difference is the way to determine neighbourhood weight vectors. MOEA/D selects parents from solutions of neighbourhood weight vectors and generates offspring. As shown in Figure 4(a), the weight vectors of the conventional MOEA/D are uniformly distributed in the weight space. Therefore, in the conventional MOEA/D, neighbourhood weight vectors are T-nearest neighbours to each weight vector. That is, the number of neighbourhood weight vectors is T for any weight vectors. On the other hand, as shown in Figure 4(b), the weight vectors of the proposed MOEA-MRS is not uniformly distributed in the weight space. Therefore, the proposed MOEA-MRS determines neighbourhood weight vectors based on an user-defined Euclidean distance D in the weight space. That is, in the proposed MOEA-MRS, the number of neighbourhood weight vectors depends on each weight vector.

The second difference is the way to generate offspring. Since the weight vectors of the conventional MOEA/D are uniformly distributed in the weight space, the objective space can be uniformly searched by equally giving the chance to generate offspring to each weight vector. On the other hand, since the weight vectors of the proposed MOEA-MRS is not uniformly distributed in the weight space, the objective space cannot be uniformly searched by equally giving the chance to generate offspring to each weight vector. Therefore, the proposed MOEA-MRS introduces a counter $c^{(i,p)}$ for each weight vector and controls the number of offspring generations.

The entire algorithm of the proposed MOEA-MRS is described in the following.

Step 1	Initialisation:	
	Step 1-1	Calculate distances between any two weight vectors in $\mathcal{L}'$ and find the nearest weight vectors within the user-defined distance D to each weight vector. For each $\lambda_{(i,p)}$ in $\mathcal{L}'$, set $B(i, p) = \{(i_1, p_1), (i_2, p_2), \dots \}$, where $\lambda^{(i_1,p_1)}$, $\lambda^{(i_2,p_2)}$,... are the weight vectors within the user-defined distance D from $\lambda^{(i,p)}$.
	Step 1-2	For each $i \in \{1, 2,\dots,N\}$, randomly generate one solution and initialise all $x^{(i,p)}$ ($p = 0, 1,\dots,J$) by the generated solution,

i.e., $x^{(i,0)} = x^{(i,1)} = \ldots = x^{(i,J)}$.

Step 1-3   Initialise each counter $c^{(i,p)}$ of the weight vector $\lambda^{(i,p)}$ by zero.

Step 2   Solution Search:

Perform the following procedure N times.

Step 2-1   Find the index $(i^{\min}, p^{\min})$ of the weight vector $\lambda^{(i^{\min}, p^{\min})}$ which has the minimum counter value $c^{(i^{\min}, p^{\min})}$ among all weight vectors in $\mathcal{L}'$. Then, select $x^{(i^{\min}, p^{\min})}$ as the first parent.

Step 2-2   Randomly choose an index $(i_q^{\min}, p_q^{\min})$ from $B(i^{\min}, p^{\min})$ and select $x^{(i_q^{\min}, p_q^{\min})}$ as the second parent. Then, generate an offspring y from two parents $x^{(i^{\min}, p^{\min})}$ and $x^{(i_q^{\min}, p_q^{\min})}$ by applying genetic operators.

Step 2-3   For each index $(i_j^{\min}, p_j^{\min}) \in B(i^{\min}, p^{\min})$, if $s(y \mid \lambda^{(i_j^{\min}, p_j^{\min})})$ is better than $s(x^{(i_j^{\min}, p_j^{\min})} \mid \lambda^{(i_j^{\min}, p_j^{\min})})$, then the current solution $x^{(i_j^{\min}, p_j^{\min})}$ is replaced by the generated offspring y, i.e., $x^{(i_j^{\min}, p_j^{\min})} = y$.

Step 2-4   For each index $(i_j^{\min}, p_j^{\min}) \in B(i^{\min}, p^{\min})$, $c^{(i_j^{\min}, p_j^{\min})}$ is incremented by one.

Step 3   Termination:

If the termination criterion is satisfied, then stop and pick non-dominated solutions on fitness vector f' from $\{x^{(i,0)}, x^{(i,1)}, \ldots, x^{(i,J)}\}$ for each $i \in \{1, 2, \ldots, N\}$ as the output of the optimisation. Otherwise, go to Step 2.

4.5   MOEA-pMRS: preference-based multi-level robust solution search

MOEA-MRS described in the previous section tries to find multi-level robust solutions from the minimum noise to the best average objective values in a single run of the algorithm. For the case that the decision maker has a preference for the noise level, as an extension of MOEA-MRS, we also propose a MOEA-pMRS. MOEA-MRS described in the previous section controls the amount of solution search for each area of $m + 1$ dimensional objective space by counting $c^{(i,p)}$. $m + 1$ dimensional objective space can be uniformly searched by incrementing $c^{(i,p)}$ one by one. For the preference-based search of the noise level, the proposed MOEA-pMRS varies the increased amount to each $c^{(i,p)}$. In MOEA-pMRS, the increased amount to $c^{(i,p)}$ is increased by increasing the distance from the preferred noise level. Since the increased amount of the preferred noise level becomes lower than the ones of other noise levels, solutions with the preferred noise level gets many chances to generate offspring and update current solutions. For the preferred noise level $F (= [0, J])$, MOEA-pMRS is the algorithm that Step 2-4) of MOEA-MRS described in the previous section is replaced by the following Step 2-4').
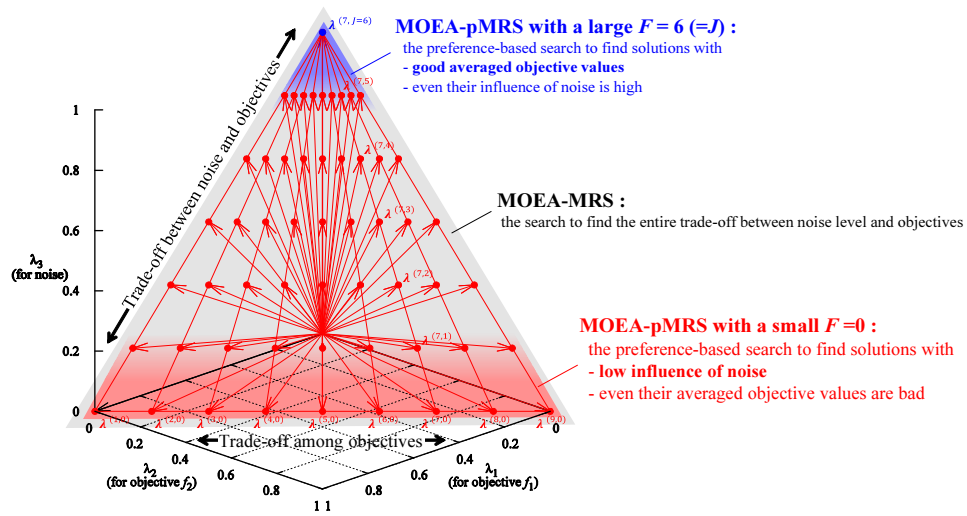
Step 2 – 4')

For each $(i_j^{\min}, p_j^{\min}) \in B(i^{\min}, p^{\min})$, $|F - p_j^{\min}|^{\alpha} + 1$ is added to $c^{(i_j^{\min}, p_j^{\min})}$,

i.e., $c^{(i_j^{\min}, p_j^{\min})} = c^{(i_j^{\min}, p_j^{\min})} + |F - p_j^{\min}|^{\alpha} + 1$.

α is the parameter to control the increased amount to $c^{(i,p)}$, and $\alpha = 2$ is used in this work.

Figure 5 shows a conceptual figure of focused search areas of MOEA-MRS, MOEA-pMRS with a large *F* and MOEA-pMRS with a small *F*. MOEA-MRS searches the entire trade-off between the noise level and objective values while MOEA-pMRSs focus the search on specific noise levels. MOEA-pMRS with a large *F* searches solutions with good average objective values even their influence of noise is high. On the other hand, MOEA-pMRS with a small *F* searches solutions with low influence of noise even their average objective values are bad. In this way, if the decision maker have a preference for specific noise level, MOEA-pMRS can focus the solution search on the preferred noise level.

Figure 5 Difference of focused search areas of MOEA-MRS and two MOEA-pMRSs with different values of *F* (see online version for colours)

5 Experimental setup

5.1 Test problems

To verify the effectiveness of the proposed MOEA-MRS and MOEA-pMRS, we use two test problems. The first one is the noisy DTLZ2 which is a continuous problem, and the second one is the noisy multi-objective knapsack problem which is a discrete problem. Both of them are categorised as *Type C* described in Section 2.2.

5.1.1 Continuous test problem: nDTLZ2

We extend DTLZ2 problem (Deb et al., 2002) to a noisy continuous test problem. The noisy DTLZ2 (nDTLZ2) is simply formulated by

$$\text{Minimise } f_j^{\text{DTLZ2}}(\boldsymbol{x}) + z(\boldsymbol{x}) \, (j = 1, 2, \dots, m), \qquad (9)$$

where f_j^{DTLZ2} $(j = 1, 2, \ldots, m)$ are the original objective functions of DTLZ2 (Deb et al., 2002), and $z(\boldsymbol{x})$ is the noise function. Note that nDTLZ2 is categorised as *Type C* described in Section 2.2 because the noise level depends on $\boldsymbol{x}$. More precisely, nDTLZ2 is formulated by

Minimise

$$f_1^z(\boldsymbol{x}) = \left(1 + g(\boldsymbol{x}^M)\right)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\cos\left(x_{m-1}\frac{\pi}{2}\right) + z(\boldsymbol{x}^M),$$

$$f_2^z(\boldsymbol{x}) = \left(1 + g(\boldsymbol{x}^M)\right)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\cos\left(x_{m-2}\frac{\pi}{2}\right)\cos\left(x_{m-1}\frac{\pi}{2}\right) + z(\boldsymbol{x}^M),$$

$$f_3^z(\boldsymbol{x}) = \left(1 + g(\boldsymbol{x}^M)\right)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\ldots\sin\left(x_{m-2}\frac{\pi}{2}\right) + z(\boldsymbol{x}^M),$$

$$\vdots$$

$$f_m^z(\boldsymbol{x}) = \left(1 + g(\boldsymbol{x}^M)\right)\sin\left(x_1\frac{\pi}{2}\right) + z(\boldsymbol{x}^M),$$

(10)

where g is the distance function, and z is the noise function. z is the only difference from the original DTLZ2. A solution $\boldsymbol{x}$ consists of n variables $\{x_1, x_2, \ldots, x_{m-1}, x_m^M, x_{m+1}^M, \ldots, x_n^M\}$, and all the elements are real values in the range [0, 1]. The last part $\boldsymbol{x}^M = \{x_m^M, x_{m+1}^M, \ldots, x_n^M\}$ in $\boldsymbol{x}$ determines the distance $g(\boldsymbol{x}^M)$ between $\boldsymbol{x}$ and the true Pareto front in the objective space. $g(\boldsymbol{x}_M)$ is defined by the following equation.

$$g(\boldsymbol{x}^M) = \sum_{x_i^M \in \boldsymbol{x}^M} \left(x_i^M - 0.5\right)^2.$$

(11)

A solution with $\boldsymbol{x}^M = \{0.5, 0.5, \ldots, 0.5\}$ achieves the minimum $g(\boldsymbol{x}^M) = 0$ and becomes a true POS.

In the nDTLZ2, the last part $\boldsymbol{x}^M$ also determines the noise level $z(\boldsymbol{x}^M)$. $z(\boldsymbol{x}^M)$ is formulated by

$$g(\boldsymbol{x}^M) = R \cdot N(0,1) \cdot \sum_{x_i^M \in \boldsymbol{x}^M} \left(x_i^M - 0.7\right)^2,$$

(12)

where R is the scaling parameter to control the scale of noise level, and N is the zero-mean Gaussian probability distribution function with the standard deviation 1. In the nDTLZ2, a solution with $\boldsymbol{x}^M = \{0.7, 0.7, \ldots, 0.7\}$ has the smallest noise.

Figure 6(a) shows the objective space of the nDTLZ2 with $m = 2$ objectives, and Figure 6(b) shows g and z functions for a variable $x_i^M (\in \boldsymbol{x}^M)$. In the nDTLZ2, the distance to the true Pareto front in the objective space is determined by g. For example, the solution $\boldsymbol{a}$ with the smallest g in Figure 6(b) is the closest to the true Pareto front among three solutions in Figure 6(a). Contrary, the solution $\boldsymbol{c}$ with the largest g in Figure 6(b) is the farthest to the true Pareto front among three solutions in Figure 6(a). Next, in the nDTLZ2, the noise level is determined by z. In Figure 6(a), the size of circle around each solution indicates the noise level. Although the noise of $\boldsymbol{b}$ is the smallest, its distance to the true Pareto front is larger than $\boldsymbol{a}$. Contrary, although $\boldsymbol{a}$ is closest to the true Pareto front in the objective space, its noise is larger than $\boldsymbol{b}$. Additionally, $\boldsymbol{c}$ is the farthest

from the true Pareto front and its noise is the largest among three solutions. That is, there is the optimal trade-off between the distance g and the noise z in the nDTLZ2, and the task of the proposed MOEA-MRS in the nDTLZ2 is to find solutions distributed in the range $0.5 \le x_i^M \le 0.7$ (grey region) by the solution search with the extended weight vectors.

Figure 6 The noisy DTLZ2 problem (nDTLZ2), (a) the objective space (b) the distance g and the noise z (see online version for colours)

(a)

(b)

5.1.2 Discrete test problem: nMOKP

We also extend multi-objective knapsack problem (MOKP) (Zitzler and Thiele, 1999) to a noisy discrete test problem. The noisy MOKP (nMOKP) is simply formulated by

$$\begin{cases} \text{Maximise } f_j^z(\boldsymbol{x}) = f_j^{\text{MOKP}}(\boldsymbol{x}) + z(\boldsymbol{x}) \\ \text{Subject to } \sum_{l=1}^n w_{l,j} \cdot x_l \le c_j \end{cases} \quad (j = 1, 2, \dots, m), \tag{13}$$

where f_j^{MOKP} $(j = 1, 2, \dots, m)$ are the original objective functions of MOKP (Zitzler and Thiele, 1999), and $z(\boldsymbol{x})$ is noise the function. The noise function $z(\boldsymbol{x})$ is the only difference from the original MOKP. Note that nMOKP is also categorised as *Type C* because the noise level depends on $\boldsymbol{x}$. More precisely, the noisy objective functions f_j^z $(j = 1, 2, \dots, m)$ are formulated by

$$f_j^z(\boldsymbol{x}) = \sum_{l=1}^n p_{l,j} \cdot x_l + \sum_{l=1}^n R \cdot N(0, l / 3n) \cdot x_l \quad (j = 1, 2, \dots, m) \tag{14}$$

where, the first term is the original objective function f_j^{MOKP}, and the second term is the noise function $z(\boldsymbol{x})$. In the nMOKP, there are n items and m knapsacks (objectives). Each item l has m kinds of profits $p_{l,j}$ $(j = 1, 2, \dots, m)$ and m kinds of weights $w_{l,j}$ $(j = 1, 2, \dots, m)$. The task is to find combinations of items $\boldsymbol{x} = \{x_1, x_2, \dots, x_n\} \in \{0, 1\}^n$

which maximises the total of profits on m kinds of objectives subject to the total of weights does not exceed m kinds of knapsack capacities c_j. The capacities of knapsacks c_j are defined as

$$c_j = \phi \cdot \sum_{l=1}^{n} w_{l,j} \quad (j = 1, 2, ..., m) \tag{15}$$

where ϕ is the feasibility ratio for each knapsack (constraint). In this work, infeasible solutions are repaired by repeatedly removing a randomly chosen item from the solution until all constraints are satisfied. Also, R is the scaling parameter to control the scale of noise level, and N is the zero-mean Gaussian probability distribution function with the standard deviation $l/3n$. Note that the noise value, the second term of equation (14), is increased by increasing item number l.

All profits $p_{l,j}$ and weights $w_{l,j}$ are generated by random integers in the range of [10, 100] in the same manner as the original MOKP (Zitzler and Thiele, 1999). However, for analysis of the experimental result, a special reordering of pairs of profit $p_{l,1}$ and weight $w_{l,1}$ ($l = 1, 2,…,n$) is performed in this work. We calculate ratios $p_{l,1}/w_{l,1}$ ($l = 1, 2,…,n$) and reorder pairs of profit $p_{l,1}$ and weight $w_{l,1}$ to satisfy $p_{1,1}/w_{1,1} > p_{n,1}/w_{n,1} > p_{2,1}/w_{2,1} > p_{n-1,1}/w_{n-1,1…}$. That is, the ratio $p_{l,1}/w_{l,1}$ is increased by increasing the difference $|l - n/2|$ between item number l and the intermediate item number $n/2$.

5.2 Parameters

In this work, we use the nDTLZ2 problem with $m = 2$ objectives, $n = 11$ variables and the scaling parameter $R = 0.125$. For solving the nDTLZ2, we use SBX crossover with the crossover ratio 0.8 and the distribution parameter $\eta_c = 15$ and the polynomial mutation with the mutation ratio $1/n$ and the distribution parameter $\eta_m = 20$ (Deb and Goyal, 1996).

Also, we use the nMOKP with $m = 2$ objectives, $n = 500$ items (bits) and the scaling parameter $R = 15$. For solving then MOKP, we use the uniform crossover with the crossover ratio 0.8 and the bit-flip mutation with the mutation ratio $4/n$.

For both problems, as the common parameters, the basic decomposition parameter and the extended decomposition parameter are set to $H = 200$ and $J = 40$, respectively. The distance to determine the neighbourhood weight vectors is set to $D = 0.06$. To calculate the average objective function values f'_j ($j = 1, 2,…,m$) and the total standard deviation f'_{m+1}, the number of evaluation for each generated solution is set to $r = 100$.
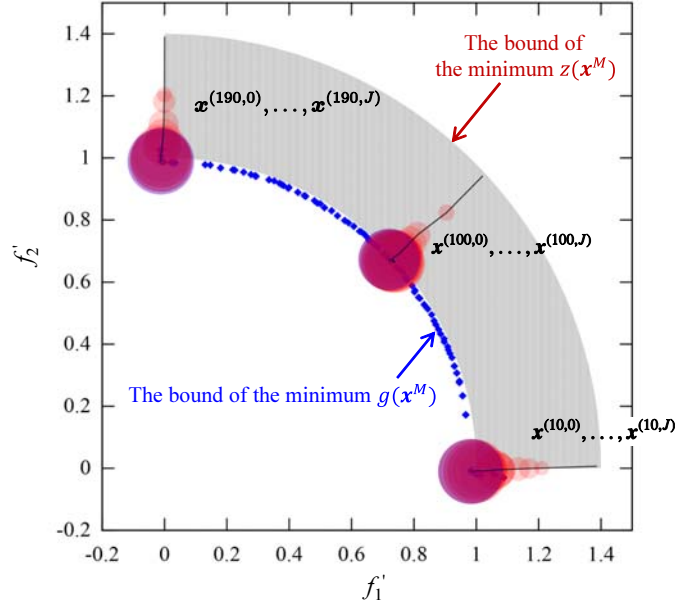
6 Experimental results and discussion

6.1 Results of MOEA-MRS in nDTLZ2

First, we verify the effectiveness of the proposed MOEA-MRS on the nDTLZ2 problem. In this section, the termination criterion to stop MOEA-MRS is set to 8,000 generations. Figure 7 shows the obtained solutions in the objective space. Note that the both objective functions of the nDTLZ2 should be minimised. All solutions are plotted based on their average objective function values f'_1 and f'_2. First, the blue points are the obtained solutions $x^{(1,J)}$, $x^{(2,J)}$,…,$x^{(N,J)}$ without considering the noise. Next, we focus only on three

basic weight vector indices $i = \{10, 100, 190\}$, solutions $\boldsymbol{x}^{(10,J)}$, $\boldsymbol{x}^{(100,J)}$, $\boldsymbol{x}^{(190,J)}$ without considering the noise are plotted with blue circles, and solutions $\boldsymbol{x}^{(10,p)}$, $\boldsymbol{x}^{(100,p)}$, $\boldsymbol{x}^{(190,p)}$ ($p = 0, 1,\ldots,J - 1$) considering the noise are plotted with red circles. The size of circle indicates the noise level.

Figure 7 The obtained solutions in the nDTLZ2 problem (see online version for colours)

From the result, first we can see that the blue solutions $\boldsymbol{x}^{(1,J)}$, $\boldsymbol{x}^{(2,J)}$,…,$\boldsymbol{x}^{(N,J)}$ without considering the noise approximate the trade-off between two objective functions. These blue solutions are used for the first stage of the decision-making shown in Figure 1. Next, for each of three basic weight vector indices $i = \{10, 100, 190\}$, we can see that the proposed MOEA-MRS can simultaneously obtain multi-level robust solutions with different levels of noise in a single run of the algorithm. These red solutions are used for the second stage of the decision-making shown in Figure 1.

Next, we analyse the obtained solutions in the variable space. Here, we focus on the obtained solutions $\boldsymbol{x}^{(190,p)}$ ($p = 0, 1,\ldots,J$) for the basic weight vector index $i = 190$. Figure 8 shows the distance function g and the noise function z for the variable $x_2^M (\in \boldsymbol{x}^M)$, and the obtained solutions $\boldsymbol{x}^{(190,p)}$ ($p = 0, 1,\ldots,J$) are plotted on the horizontal axis. The solution $\boldsymbol{x}^{(190,J)}$ without considering the noise is plotted as the blue point, and the solutions $\boldsymbol{x}^{(190,p)}$ ($p = 0, 1,\ldots,J-1$) considering the noise are plotted as the red points.

From the result, we can see that the obtained solutions are distributed in or near the grey region ($0.5 \leq x_2^M \leq 0.7$) showing the optimal trade-off between the distance g and the noise z. Also, Figure 9 shows the obtained solutions $\boldsymbol{x}^{(190,p)}$ ($p = 0, 1,\ldots,J$) on all variables $\boldsymbol{x}^M = \{x_2^M, x_3^M,\ldots,x_{n=11}^M\}$. From the result, we can see that the obtained solutions are distributed in or near the optimal grey regions of all variables in $\boldsymbol{x}^M$.

Figure 8   The obtained solutions $x^{(190,p)}$ ($p = 0, 1,\ldots,J$) on the variable x_2^M (see online version for colours)

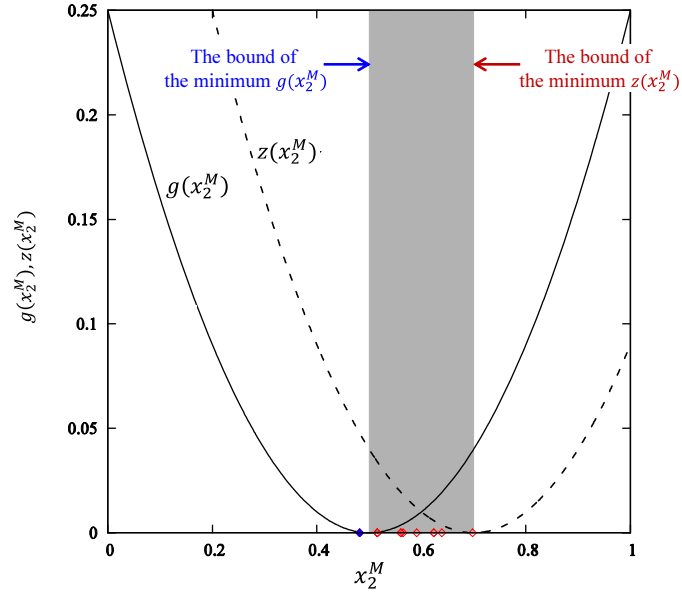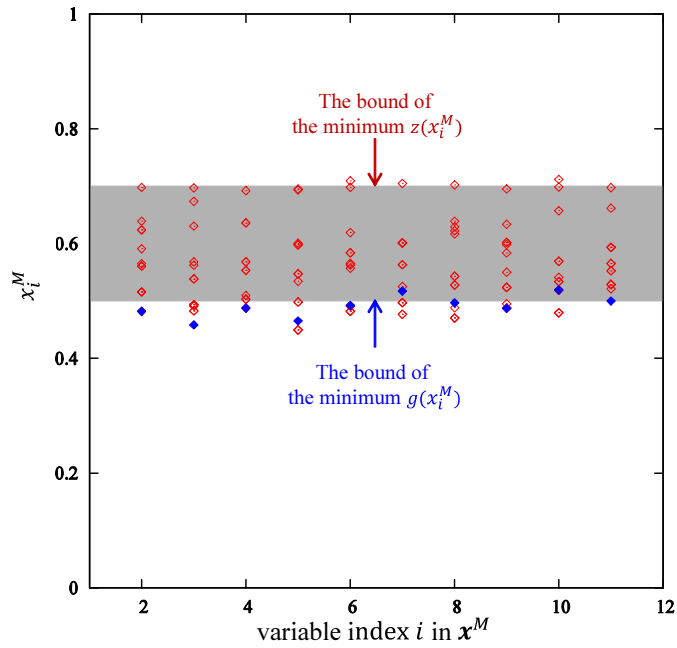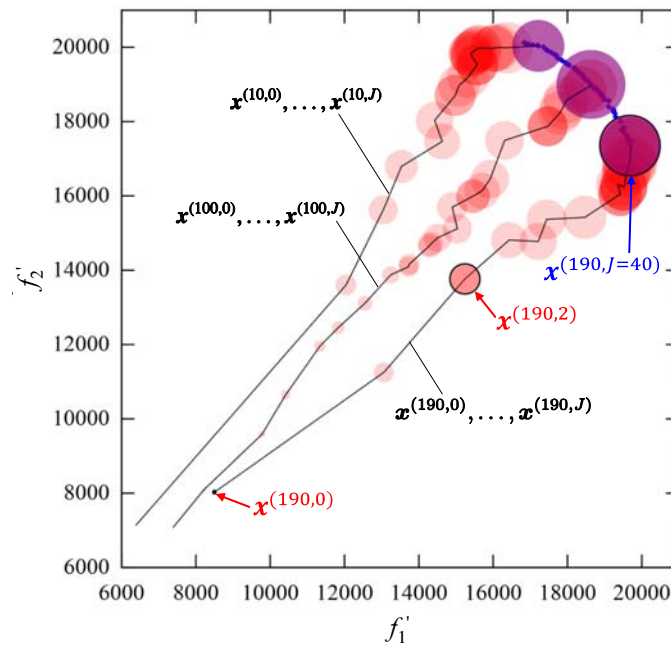**Figure 9**   The obtained of solutions $x^{(190,p)}$ ($p = 0, 1,\ldots,J$) on all variables in x^M (see online version for colours)

6.2 Results of MOEA-MRS in nMOKP

Next, we verify the effectiveness of the proposed MOEA-MRS in the nMOKP. In this section, the termination criterion of MOEA-MRS is set to 10^4 generations. Figure 10 shows the obtained solutions in the objective space. Note that the both objective functions of the nMOKP should be maximised. The blue points are the obtained solutions $x^{(1,J)}$, $x^{(2,J)}$,...,$x^{(N,J)}$ without considering the noise. We focus on three basic weight vector indices $i = \{10, 100, 190\}$, solutions $x^{(10,J)}$, $x^{(100,J)}$, $x^{(190,J)}$ without considering the noise are plotted with blue circles, and solutions $x^{(10,p)}$, $x^{(100,p)}$, $x^{(190,p)}$ ($p = 0, 1,...,J - 1$) considering the noise are plotted with red circles.

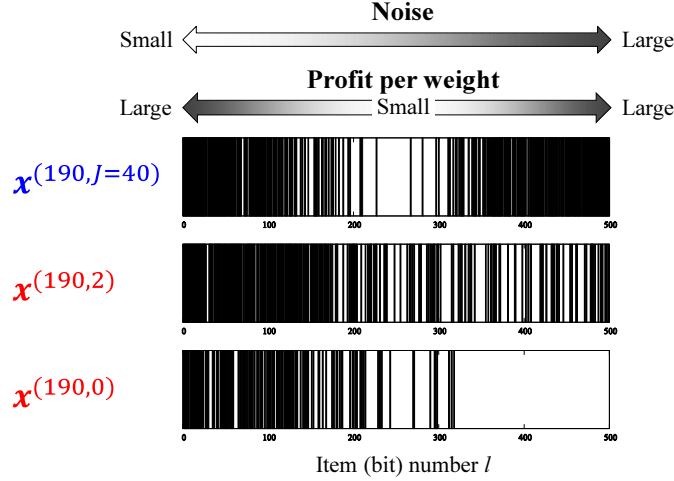**Figure 10** The obtained solutions in the nMOKP problem (see online version for colours)

From the result, as a general tendency, we can see that solutions with large objective values f_1' and f_2' have large noises with large circles. This result in the nMOKP also reveals that the proposed MOEA-MRS is able to simultaneously find multi-level robust solutions with different levels of the noise for each search direction in the objective space in the single run of the algorithm.

Next, we focus on the basic weight vector index $i = 190$ and observe the selected items of the obtained solutions in the nMOKP. The selected items of three solutions $x^{(190,0)}$, $x^{(190,2)}$ and $x^{(190,J=40)}$ are shown in Figure 11. For each item (bit) l, selected item (1) is shown in black, and unselected item (0) is shown in white in this figure. As described in Section 5.1, the noise level is increased by increasing the item (bit) number l. Also, the ratio $p_{l,1}/w_{l,1}$ is increased by increasing the difference $|l - 250|$ between item number l and the intermediate item number 250. From this result, we can see that the solution $x^{(190,J=40)}$ with a large noise selects many items with large item number l. That is, $x^{(190,J=40)}$ selects items with high $p_{l,1}/w_{l,1}$ even if their noises are high. On the other hand, the robust

solution $x^{(190,0)}$ with a small noise tends to not select items with large l. That is, $x^{(190,0)}$ avoids to select items with high noise even if their $p_{l,1}/w_{l,1}$ are high. This result reveals that the proposed MOEA-MRS is able to extract items (variables) with low influences of noise.

Figure 11 Selected items (bits) of three solutions with different noise levels (see online version for colours)

Item (bit) number l
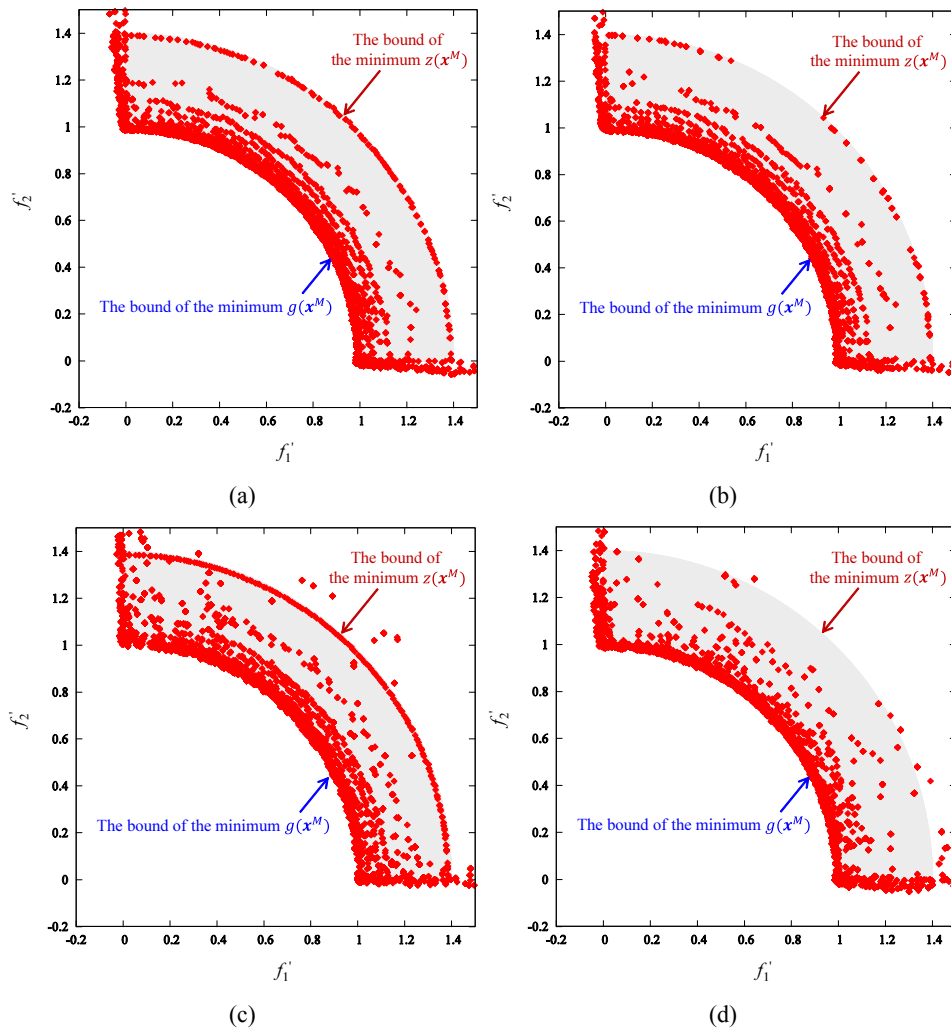
6.3 Results of MOEA-pMRS in nDTLZ2

Finally, we verify the effectiveness of the proposed MOEA-pMRS in nDTLZ2. Figure 12(a) shows all solutions obtained by MOEA-MRS at 8,000 generation, and these solutions are obtained in the same experiment shown in Section 6.1. Figures 12(b) to 12(d) show all solutions obtained by MOEA-MRS and two MOEA-pRMSs with $F = \{0, J = 40\}$ at 3,000 generation. Note the difference between Figure 12(a) and Figures 12(b) to 12(d) is the total number of generations. MOEA-MRS searches the entire trade-off between the noise level and the objective values. MOEA-pMRS with $F = 0$ focuses the solution search on the minimum noise level, and MOEA-pMRS with $F = 40$ focuses the solution search on the best (minimum) average objective values.

From the result in Figure 12(a), we can see that the solutions obtained by MOEA-MRS at 8,000 generation are widely distributed in the grey region which is the entire optimal trade-off between the noise level and the objective values in the objective space of nDTLZ2. However, from the result in Figure 12(b), the solutions obtained by MOEA-MRS at 3,000 generation have large gaps especially around the border of the minimum noise. That is, to approximate the entire trade-off between the noise level and the average objective values, 3,000 generations are not enough for MOEA-MRS.

If the decision maker has a preference for a specific noise level, MOEA-pMRS can be applied and reduce the total computational time. Figure 12(c) shows solutions obtained by MOEA-pMRS with $F = 0$ which has a preference for the lowest noise level at 3,000 generation. From the result, we can see that the distribution density of solutions

around the border of the minimum noise value is higher than the ones of not only MOEA-MRS at 3,000 generation but also MOEA-MRS at 8,000 generation. Similarly, Figure 12(d) shows solutions obtained by MOEA-pMRS with $F = 40$ which has a preference for the best (minimum) objective values at 3,000 generations. From the result, we can see that the distribution density of solutions around the border of the best (minimum) objective values is higher than the ones of not only MOEA-MRS at 3,000 generation but also MOEA-MRS at 8,000 generation. These results reveal that the proposed MOEA-pMRS can emphasise the solution search for a preferred noise level with a small number of generations if the decision maker has a preference for the noise level.

Figure 12 All solutions obtained by two MOEA-MRSs with different total number of generations and two MOEA-pMRS with $F = \{0, J = 40\}$, (a) MOEA-MRS (8,000 generation) (b) MOEA-MRS (3,000 generation) (c) MOEA-pMRS with $F = 0$ (3,000 generation) (d) MOEA-pMRS with $F = 40$ (3,000 generation) (see online version for colours)

7   Conclusions

For the two-stage MCDM considering the noise level, in this work we proposed MOEAMRS which simultaneously searches multi-level robust solutions with different noise levels for each search direction in the multi-dimensional objective space of NMOPs. As an extension of MOEA-MRS, we also proposed MOEA-pMRS which focuses the solutions search on a specific noise level to consider the case that the decision maker has a preference for the noise level. To verify the effectiveness of the proposed MOEA-MRS and MOEA-pMRS, we used continuous nDTLZ2 and discrete nMOKP problems extended from the conventional DTLZ2 and MOKP, respectively. The experimental results showed that the proposed MOEA-MRS was able to obtain multi-level robust solutions with different noise levels for each search direction in a single run of the algorithm. In the nDTLZ2 problem, we showed that the proposed MOEA-MRS obtained solutions in the optimal region showing the optimal trade-off between the distance to the true Pareto front and noise levels. Also, in then MOKP, we showed that the proposed MOEA-MRS was able to extract items (variables) with low influences of noise. Furthermore, we showed that the proposed MOEA-pMRS was able to emphasise the solution search for specific noise levels.

As future work, we will try to decrease the total number of evaluations in the proposed MOEA-MRS and MOEA-pMRS since the current algorithm needs many evaluations to assign fitness values for each solution.

Acknowledgements

References

Babbar, M., Lakshmikantha, A. and Goldberg, D.E. (2003) 'Modified NSGA-II to solve noisy multi-objective problems', *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, Late-Breaking Papers, pp.21–27.

Basseur, M. and Zitzler, E. (2006) 'Handling uncertainty in indicator-based multi-objective optimization', *Int'l Journal of Computational Intelligence Research*, Vol. 2, No. 3, pp.255–272.

Buche, D., Stoll, P., Dornberger, R. and Koumoutsakos, P. (2002) 'Multiobjective evolutionary algorithm for the optimization of noisy combustion processes', *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, Vol. 32, No. 4, pp.460–473.

Bui, L.T., Abbass, H.A. and Essam, D. (2005) 'Fitness inheritance for noisy evolutionary multi-objective optimization', *Proc. of the Genetic and Evolutionary Computation Congress (GECCO)*, pp.779–785.

Coello, C.A.C. and Lamont, G.B. (2004) *Applications of Multi-Objective Evolutionary Algorithms*, World Scientific, Singapore.

Coello, C.A.C., Van Veldhuizen, D.A. and Lamont, G.B. (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition*, Kluwer Academic Publishers, Boston.

Deb, K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK.

Deb, K. and Goyal, M. (1996) 'A combined genetic adaptive search (GeneAS) for engineering design', *Computer Science and Informatics*, Vol. 26, No. 4, pp.30–45.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2002) 'Scalable multi-objective optimization test problems', *Proc. of CEC*, pp.825–830.

Goh, C. and Tan, K.C. (2009) *Evolutionary Multi-objective Optimization in Uncertain Environments*, Springer-Verlag, Berlin, Heidelberg.

Goh, C.K. and Tan, K.C. (2007) 'An investigation on noisy environments in evolutionary multiobjective optimization', *IEEE Trans. on Evolutionary Computation*, Vol. 11, No. 3, pp.354–381.

Hashimoto, T. and Sato, H. (2015) 'A study on multi-level robust solution search for noisy multi-objective optimization problems', *Proc. of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, *Proc. in Adaptation, Learning and Optimization*, Volume 2, Springer, pp.239–253.

Hughes, E.J. (2001a) 'Evolutionary multi-objective ranking with uncertainty and noise', *Proc. of the 1st Conference on Evolutionary Multi-Criterion Optimization (EMO)*, pp.329–343.

Hughes, E.J. (2001b) 'Constraint handling with uncertain and noisy multi-objective evolution', *Proc. of the Congress on Evolutionary Computation (CEC)*, Vol. 2, pp.963–970.

J. Teich, (2001) 'Pareto-front exploration with uncertain objectives', *Proc. of the 1st Conference on Evolutionary Multi-Criterion Optimization (EMO*, Springer, pp.314–328.

Li, K., Zhang, Q., Kwong, S., Li, M. and Wang, R. (2013) 'Stable matching based selection in evolutionary multiobjective optimization', *IEEE Trans. on Evolutionary Computation*, Vol. 18, No. 6, pp.909–923.

Miettinen, K. (1999) *Nonlinear Multiobjective Optimization*, K luwer, Norwell, MA.

Murata, T., Ishibuchi, H. and Gen, M. (2001) 'Specification of genetic search directions in cellular multi-objective genetic algorithm', *Proc. of EMO*, LNCS, Vol. 1993, pp.82–95.

Park, T. and Ryu, K. (2011) 'Accumulative sampling for noisy evolutionary multi-objective optimization', *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, pp.793–800.

Zhang, Q. and Li, H. (2007) 'MOEA/D: a multiobjective evolutionaly algorithm based on decomposition', *IEEE Trans. on Evolutionary Computation*, Vol. 11, No. 6, pp.712–713.

Zitzler, E. and Künzli, S. (2004) 'Indicator-based selection in multiobjective search', *Proc. of the Parallel Problem Solving from Nature (PPSN VIII)*, Springer, pp.832–842.

Zitzler, E. and Thiele, L. (1999) 'Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach', *IEEE Trans. on Evolutionary Computation*, Vol. 3, No. 4, pp.257–271.