Multi-factorial Evolutionary Algorithm Using Objective Similarity Based Parent Selection

Shio Kawakami(✉), Keiki Takadama, and Hiroyuki Sato

The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
{s.kawakami,h.sato}@uec.ac.jp, keiki@inf.uec.ac.jp

Abstract. This work proposes a multi-factorial evolutionary algorithm encouraging crossovers among solutions with similar target objective functions and suppressing crossovers among solutions with dissimilar target objective functions. Evolutionary multi-factorial optimization simultaneously optimizes multiple objective functions with a single population, a solution set. Each solution has a target objective function, and sharing solution resources in one population enhances the simultaneous search for multiple objective functions. However, the conventional multi-factorial evolutionary algorithm does not consider similarities among objective functions. As a result, solutions with dissimilar target objectives are crossed, and it deteriorates the search efficiency. The proposed algorithm estimates objective similarities based on search directions of solution subsets with different target objective functions in the design variable space. The proposed algorithm then encourages crossovers among solutions with similar target objectives and suppresses crossovers among solutions with dissimilar objectives. Experimental results using multi-factorial distance minimization problems show the proposed algorithm achieves higher search performance than the conventional evolutionary single-objective optimization and multi-factorial optimization.

Keywords: Multi-factorial optimization · Evolutionary algorithms · Objective function similarity

1 Introduction

In product developments and releases, multiple variant products with different specifications are often developed and released simultaneously. For instance, multiple smartphone models with different display sizes and processors depending on various users' demands are simultaneously developed and released from a device maker. These multiple models involve common parts that can be shared, and the independent and parallel design optimization of each model will not be efficient. This kind of design optimization problem belongs to the class of *multi-factorial optimization problem*. Multi-factorial optimization is that simultaneous

T. Nakano (Ed.): BICT 2021, LNICST 403, pp. 45–60, 2021.
https://doi.org/10.1007/978-3-030-92163-7_5

optimizations of multiple objective functions on common design variables. Evolutionary algorithms are particularly suited for multi-factorial optimization since solutions can share design variable information while parallelly optimizing multiple objective functions [1,2].

The conventional evolutionary approach [1–4] assigns an objective function to each solution as a *skill factor* and simultaneously optimizes multiple objective functions by using a single population, a solution set. To generate new solutions, the conventional MFEA [1,2] uses crossover, which recombines design variable values of solutions in the population. MFDE [3] is based on the differential evolution [5] and uses differential vectors of solutions in the design variable space. MFPSO [4] is based on the particle swarm optimizer [6] and uses velocity vectors obtained by historical and current solutions in the design variable space. These conventional algorithms commonly recombine solutions to generate new solutions. However, the conventional algorithms recombine solutions without considering similarities among objective functions. The recombinations of solutions with skill factors of similar objective functions encourage optimizations of their objective functions. However, the recombinations of solutions with skill factors of dissimilar objective functions deteriorate the simultaneous optimization of their objective functions. The use of similarities among objective functions for the recombinations would enhance evolutionary multi-factorial optimization.

To accelerate evolutionary multi-factorial optimization, in this work, we extend the conventional MFEA that uses the crossover for the recombination. The proposed algorithm estimates similarities among objective functions and utilizes them for crossovers to generate new solutions. The proposed algorithm encourages crossovers of solutions with skill factors of similar objective functions. This enhances the information exchange among these solutions and synergic effects for optimizations of these similar objective functions. Conversely, the proposed algorithm suppresses crossovers of solutions with skill factors of dissimilar objective functions. We use multi-factorial distance minimization problems [7] with several situations of objective function similarities and compare the search performance of the proposed algorithm with the conventional MFEA [1,2] and the single-objective EA without any interactions of solutions with different skill factors.

2 Multi-factorial Optimization Problem

2.1 General Definition

The general definition of multi-factorial optimization problems is as follows. For design variable vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ in the design variable space $\mathcal{X}$, there are K $(= |\mathcal{K}|)$ objective functions f_k $(k \in \mathcal{K})$, where $\mathcal{K}$ is the set of objective function indices (e.g. $\mathcal{K} = \{1, 2, \ldots, K\}$). The task on multi-factorial optimization problems is to acquire K $(= |\mathcal{K}|)$ optimal solutions $\boldsymbol{x}_k^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f_k(\boldsymbol{x})$ $(k \in \mathcal{K})$.

Multiple objective functions f_k $(k \in \mathcal{K})$ exist but design variable $\boldsymbol{x}$ is common among them. Unlike multi-objective optimization to search for the optimal trade-off, the Pareto front, among objective functions, multi-factorial optimization searches K optimal solutions on K objective functions.

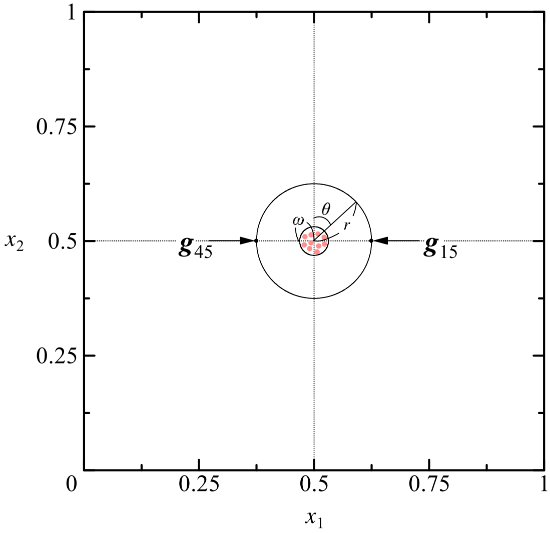


Fig. 1. Design variable space $\mathcal{X}$ of MFDMP with goal points $\boldsymbol{g}_{15}$ and $\boldsymbol{g}_{45}$

2.2 Multi-factorial Distance Minimization Problem

In this work, we use multi-factorial distance minimization problem (MFDMP) [7], which is a benchmark problem framework for multi-factorial optimization. MFDMP is an extension of multi-objective distance minimization problem [8,9] frequently used in multi-objective optimization benchmarking.

As shown in Fig. 1, MFDMP has a $d = 2$ dimensional design variable space $\mathcal{X} = [0, 1]^2$. $\boldsymbol{g}_k$ is a goal point indicating the optimal point $\boldsymbol{x}_k^*$ of objective function f_k. Goal point $\boldsymbol{g}_k$ is set on the circle of the center position $\boldsymbol{c} = (0.5, 0.5)$ and the radius r. The maximum number of goal positions is limited to 60 in this work. We treat the circle with the radius r as an analog clock and set goal points on minute positions. The clock-wise angle θ based on the center position $\boldsymbol{c} = (0.5, 0.5)$ and $x_1 = 0.5$ is used to set each goal point $\boldsymbol{g}_k$. Goal point $\boldsymbol{g}_k$ at k minute position of the analog clock is given by

$$\boldsymbol{g}_k = \left(r \cdot \sin\left(k \cdot \left(\frac{2\pi}{60}\right)\right) + 0.5,\ r \cdot \cos\left(k \cdot \left(\frac{2\pi}{60}\right)\right) + 0.5 \right). \tag{1}$$

The minimizing objective function f_k corresponding to the goal point $\boldsymbol{g}_k$ is given by

$$f_k(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{g}_k||, \tag{2}$$

where $||\cdot||$ is the Euclidean distance. Figure 1 shows a case with the goal point $\boldsymbol{g}_{15}$ at 15 min position and the goal point $\boldsymbol{g}_{45}$ at 45 min position on the analog clock circle. We minimize two objective functions $f_{15}(\boldsymbol{x})$ and $f_{45}(\boldsymbol{x})$ in this case. The initial population is generated inside the circle of the center position $\boldsymbol{c} = (0.5, 0.5)$ and the radius ω in this work.

Characteristics of MFDMPs are as follows. The first is the scalability in the number of objectives since it can be easily increased by adding goal points. The second is the scalability in the similarity of objective functions since the distance of goal points in the design variable space $\mathcal{X}$ directly affects the similarity of objective functions. The third is visual analyzability since the design variable space is $d = 2$ dimensional.

3 Evolutionary Multi-factorial Optimization

3.1 Method

This work focuses on MFEA [1,2] as the representative evolutionary multi-factorial optimization algorithm. Algorithm 1 shows the pseudo-code of the conventional MFEA.

MFEA first randomly generates N solutions and composes of the parent population $\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$. For each solution $\boldsymbol{x}^i$, MFEA assigns a skill factor s^i indicating target objective function f_{s^i} and composes of the skill factor set $\mathcal{S} = \{s^1, s^2, \ldots, s^N\}$. MFEA evaluates each solution $\boldsymbol{x}^i$ on the target objective function f_{s^i} specified by its skill factor s^i. To generate new solutions, MFEA randomly choose two parents $\boldsymbol{x}^p$ and $\boldsymbol{x}^q$ from the parent population $\mathcal{P}$. MFEA then applies a crossover to them when the two parents $\boldsymbol{x}^p$ and $\boldsymbol{x}^q$ have the same skill factor (i.e. $s^p = s^q$), or under the probability rmp even if they do not have the same skill factor (i.e. $s^p \neq s^q$). A mutation to perturb design variable values is also applied to the obtained offspring $\boldsymbol{x}^{p\prime}$ and $\boldsymbol{x}^{q\prime}$, and they are added to the offspring population $\mathcal{P}'$. Skill factors $s^{p\prime}$ and $s^{q\prime}$ of offspring are inherited from the skill factors s^p and s^q of their parents. The above offspring generation is repeated until the size of the offspring population $\mathcal{S}'$ reaches to N, which is the same size of the parent population $\mathcal{P}$. For each solution in the combined population $\mathcal{P} \cup \mathcal{P}'$, we calculate scalar fitness, which is the rank value on each targeting objective function specified by skill factor. MFEA then selects the best N solutions from the combined population $\mathcal{P} \cup \mathcal{P}'$ as the new parent population $\mathcal{P}$ in terms of scalar fitness.

MFEA repeats the above process and simultaneously optimizes multiple objective functions with the single population $\mathcal{P}$.

3.2 Issue Focus

The crossover to recombine solutions with different skill factors has an important role in evolutionary multi-factorial optimization since it enhances the simultaneous optimization of multiple objective functions. Therefore, the selection of

Algorithm 1. Conventional MFEA [1,2]

$\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\} \leftarrow$ Randomly generate the population ▷ Parent population
$\mathcal{S} = \{s^1, s^2, \ldots, s^N\} \leftarrow$ Evenly assign skill factors ▷ Skill factors of $\mathcal{P}$
Evaluate each $\boldsymbol{x}^i \in \mathcal{P}$ on objective function f_{s^i}
for $g \leftarrow 1, 2, \ldots, G$ **do**
    $\mathcal{P}' \leftarrow \emptyset$ ▷ Offspring population
    $\mathcal{S}' \leftarrow \emptyset$ ▷ Skill factors of $\mathcal{P}'$
    while $|\mathcal{P}'| < N$ **do**
        $p, q \leftarrow$ Randomly choose $(1, 2, \ldots, N)$
        if $s^p = s^q \vee$ rand(0,1) $< rmp$ **then**
            $\boldsymbol{x}^{p\prime}, \boldsymbol{x}^{q\prime} \leftarrow$ Crossover $(\boldsymbol{x}^p, \boldsymbol{x}^q)$
        else
            $\boldsymbol{x}^{p\prime}, \boldsymbol{x}^{q\prime} \leftarrow$ Copy $(\boldsymbol{x}^p, \boldsymbol{x}^q)$
        end if
        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\boldsymbol{x}^{p\prime}, \boldsymbol{x}^{q\prime}\} =$ Mutation $(\boldsymbol{x}^{p\prime}, \boldsymbol{x}^{q\prime})$
        $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{s^{p\prime}, s^{q\prime}\} =$ Inherit (s^p, s^q)
    end while
    Evaluate each $\boldsymbol{x}^{i\prime} \in \mathcal{P}'$ on objective function $f_{s^{i\prime}}$
    Calculate scalar fitness $(\mathcal{P} \cup \mathcal{P}')$
    $\mathcal{P}, \mathcal{S} \leftarrow$ Select best N solutions $(\mathcal{P} \cup \mathcal{P}', \mathcal{S} \cup \mathcal{S}')$
end for
return $\boldsymbol{x}_k^* = \arg\min_{x \in \mathcal{P}} f_k(\boldsymbol{x})$ $(k \in \mathcal{K})$

parents to be crossed strongly affects crossover effects. However, the conventional MFEA randomly chooses parents without considering similarities among objective functions. As a result, solutions with skill factors of dissimilar objective functions become pairs of parents and are crossed. This would be a barrier to evolutionary multi-factorial optimization.

Evolutionary multi-factorial optimization would be further enhanced if we could estimate similarities among objective functions, encourage crossovers of solutions with similar skill factors and suppress crossovers of solutions with dissimilar skill factors.

4 Proposal: MFEA Using Objective Similarity Based Parent Selection

4.1 Summary

In this work, we propose an MFEA-based algorithm that estimates similarities of objective functions and reflects them to parent selection probabilities to enhance evolutionary multi-factorial optimization. The proposed method encourages crossovers of solutions with skill factors of similar objective functions and suppresses crossovers of solutions with skill factors of dissimilar objective functions. Algorithm 2 shows the pseudo-code of the proposed algorithm. Differences from the conventional MFEA [1,2] is highlighted with blue.

Algorithm 2. Proposed MFEA

1: $\mathcal{P} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\} \leftarrow$ Randomly generate the population ▷ Parent population
2: $\mathcal{S} = \{s^1, s^2, \ldots, s^N\} \leftarrow$ Evenly assign skill factors ▷ Skill factors of $\mathcal{P}$
3: Evaluate each $\boldsymbol{x}^i \in \mathcal{P}$ on objective function f_{s^i}
4: **for** $g \leftarrow 1, 2, \ldots, G$ **do**
5:     **for each** $k \in \mathcal{K}$ **do**
6:         $\boldsymbol{m}_k \leftarrow \left(\frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_1}{|\mathcal{P}_k|}, \frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_2}{|\mathcal{P}_k|}, \ldots, \frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_d}{|\mathcal{P}_k|}\right)$ ▷ Mean position, Eq. (3)
7:         $\boldsymbol{v}_k \leftarrow \boldsymbol{m}_k - \boldsymbol{c}$ ▷ Search direction, Eq. (4)
8:     **end for**
9:     **for each** $k \in \mathcal{K}$ **do**
10:         **for each** $l \in \mathcal{K}$ **do**
11:             $S_{k,l} \leftarrow \frac{\cos(\boldsymbol{v}_k, \boldsymbol{v}_l)+1}{2}$ ▷ Objective similarity, Eq. (5)
12:             $P_{k,l} \leftarrow \frac{S_{k,l}^{\alpha}}{\sum_{m \in \mathcal{K}} S_{k,m}^{\alpha}}$ ▷ Selection probability, Eq. (6)
13:         **end for**
14:     **end for**
15:     $\mathcal{P}' \leftarrow \emptyset$ ▷ Offspring population
16:     $\mathcal{S}' \leftarrow \emptyset$ ▷ Skill factors of $\mathcal{P}'$
17:     **while** $|\mathcal{P}'| < N$ **do**
18:         $k \leftarrow$ Randomly choose a skill factor ($\mathcal{K}$)
19:         $\boldsymbol{x}^p \leftarrow$ Randomly choose first parent ($\mathcal{P}_k$)
20:         $l \leftarrow$ Probabilistically choose a skill factor ($\mathcal{K}$, $P_{k,1}, P_{k,2}, \ldots, P_{k,K}$)
21:         $\boldsymbol{x}^q \leftarrow$ Randomly choose second parent ($\mathcal{P}_l$)
22:         $\boldsymbol{x}^{p'}, \boldsymbol{x}^{q'} \leftarrow$ Crossover $(\boldsymbol{x}^p, \boldsymbol{x}^q)$
23:         $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\boldsymbol{x}^{p'}, \boldsymbol{x}^{q'}\} =$ Mutation $(\boldsymbol{x}^{p'}, \boldsymbol{x}^{q'})$
24:         $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{s^{p'}, s^{q'}\} =$ Inherit (s^p, s^q)
25:     **end while**
26:     Evaluate each $\boldsymbol{x}^{i'} \in \mathcal{P}'$ on objective function $f_{s^{i'}}$
27:     Calculate scalar fitness $(\mathcal{P} \cup \mathcal{P}')$
28:     $\mathcal{P}, \mathcal{S} \leftarrow$ Select best N solutions $(\mathcal{P} \cup \mathcal{P}', \mathcal{S} \cup \mathcal{S}')$
29: **end for**
30: **return** $\boldsymbol{x}_k^* = \arg\min_{\boldsymbol{x} \in \mathcal{P}} f_k(\boldsymbol{x})$ $(k \in \mathcal{K})$

4.2 Search Direction

For each skill factor k indicating target objective function f_k $(k \in \mathcal{K})$, the proposed algorithm estimates the search direction $\boldsymbol{v}_k$ in the design variable space $\mathcal{X}$. 5–8 lines of Algorithm 2 corresponds to this process.

As with the conventional MFEA, the proposed algorithm also assigns a skill factor s^i to each solution $\boldsymbol{x}^i$ in the parent population $\mathcal{P}$. That is, solution $\boldsymbol{x}^i$ is evaluated and responsible for objective function f_{s^i}. For each skill factor $k \in \mathcal{K}$, we calculate the mean position vector $\boldsymbol{m}_k$ of $\mathcal{P}_k = \{\boldsymbol{x}^i \in \mathcal{P} | s^i = k\}$, which is the solution set with skill factor k in the parent population $\mathcal{P}$. The mean position vector $\boldsymbol{m}_k$ of skill factor k is given by

$$\boldsymbol{m}_k = \left(\frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_1}{|\mathcal{P}_k|}, \frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_2}{|\mathcal{P}_k|}, \ldots, \frac{\sum_{\boldsymbol{x} \in \mathcal{P}_k} x_d}{|\mathcal{P}_k|}\right), \tag{3}$$

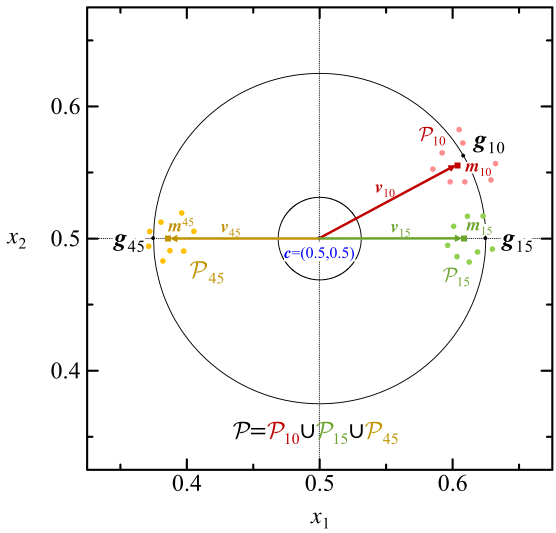


Fig. 2. Three search directions for three objective functions, goal points (Color figure online)

where $\mathcal{P}_k \subseteq \mathcal{P}$ and $\mathcal{P} = \bigcup_{k \in \mathcal{K}} \mathcal{P}_k$.

Next, for each skill factor $k \in \mathcal{K}$, we calculate the search direction vector $\boldsymbol{v}_k$ in the design variable space $\mathcal{X}$. The search direction vector $\boldsymbol{v}_k$ is calculated with the central position $\boldsymbol{c}$ of the design variable space $\mathcal{X}$ by

$$\boldsymbol{v}_k = \boldsymbol{m}_k - \boldsymbol{c}. \tag{4}$$

In the case of MFDMP, we use $\boldsymbol{c} = (0.5, 0.5)$. $\boldsymbol{c}$ indicates the mean position of the initial solutions. In this way, each skill factor k is characterised by the search direction $\boldsymbol{v}_k$ from the central position $\boldsymbol{c}$ to the mean position $\boldsymbol{m}_k$ of the current solutions $\mathcal{P}_k$ with skill factor k in the design variable space $\mathcal{X}$.

Figure 2 shows an example. The central position $\boldsymbol{c} = (0.5, 0.5)$ is shown as the blue point. There are three goals $\boldsymbol{g}_{10}, \boldsymbol{g}_{15}$, and $\boldsymbol{g}_{45}$, which respectively correspond to objective functions f_{10}, f_{15}, and f_{45}. Three subset populations $\mathcal{P}_{10}, \mathcal{P}_{15}$, and $\mathcal{P}_{45}$ with skill factors 10, 15, and 45 are shown with different colored circles. Also, the mean position vectors $\boldsymbol{m}_{10}, \boldsymbol{m}_{15}$, and $\boldsymbol{m}_{45}$ are shown as rectangles, and the search direction vectors $\boldsymbol{v}_{10}, \boldsymbol{v}_{15}$, and $\boldsymbol{v}_{45}$ are also depicted. Note that the three goal positions $\boldsymbol{g}_{10}, \boldsymbol{g}_{15}$, and $\boldsymbol{g}_{45}$ are unknown for optimizer. However, we see the similarity of f_{10} and f_{15} from the their search directions $\boldsymbol{v}_{10}$ and $\boldsymbol{v}_{15}$. Also, we see the dissimilarity of f_{15} and f_{45} from their search directions $\boldsymbol{v}_{15}$ and $\boldsymbol{v}_{45}$.

4.3 Objective Similarity and Selection Probability

Next, the proposed algorithm estimates objective similarities among objective functions and calculates selection probabilities based on the objective similarities. 9–14 lines of Algorithm 2 correspond to this process.

The proposed algorithm quantitatively estimates objective similarities among objective functions f_k $(k \in \mathcal{K})$ by using the search direction vectors $\boldsymbol{v}_k$ $(k \in \mathcal{K})$. The objective similarity $S_{k,l}$ between objective functions f_k and f_l is given by

$$S_{k,l} = \frac{\cos(\boldsymbol{v}_k, \boldsymbol{v}_l) + 1}{2} \quad (k \in \mathcal{K}, l \in \mathcal{K}). \tag{5}$$

Thus, we use the cosine similarity of the search directions $\boldsymbol{v}_k$ and $\boldsymbol{v}_l$. $S_{k,l}$ is in the value range $[0, 1]$. The higher $S_{k,l}$, the more similar objective functions.

We then calculate selection probabilities based on the objective similarities obtained above. For skill factor k $(\in \mathcal{K})$, the probabilities $P_{k,l}$ $(l \in \mathcal{K})$ to select the skill factor l are given by

$$P_{k,l} = \frac{S_{k,l}^{\alpha}}{\sum_{m \in \mathcal{K}} S_{k,m}^{\alpha}} \quad (k \in \mathcal{K}, l \in \mathcal{K}), \tag{6}$$

where $\alpha = [0, \infty]$ is the exponent parameter to bias the selection probability. $P_{k,l}$ is in the value range $[0, 1]$, and $\sum_{l \in \mathcal{K}} P_{k,l} = 1$. $\alpha = 0$ indicates that the selection probabilities of any skill factors are equal and is equivalent to the random selection. The probability of selecting solutions with similar objective functions increases as the increase of α.

4.4 Parent Selection

The proposed algorithm selects pairs of parents based on the selection probabilities obtained above. 18–21 lines of Algorithm 2 correspond to this process.

To select a pair of parents $\boldsymbol{x}^p$ and $\boldsymbol{x}^q$ from the parent population $\mathcal{P}$, we randomly choose one skill factor k among the set of skill factor indices $\mathcal{K}$. We then randomly choose the first parent $\boldsymbol{x}^p$ among $\mathcal{P}_k = \{\boldsymbol{x}^i \in \mathcal{P} | s^i = k\}$, which is the solution set with skill factor k in the parent population $\mathcal{P}$. Next, we select the skill factor l of the second parent solution based on the probabilities $P_{k,j}$ $(j \in \mathcal{K})$. For the selected skill factor l, we randomly choose the second parent $\boldsymbol{x}^q$ among $\mathcal{P}_l = \{\boldsymbol{x}^i \in \mathcal{P} | s^i = l\}$, which is the solution set with skill factor l in the parent population $\mathcal{P}$.

In this way, the proposed algorithm makes pairs of parents according to the selection probabilities based on the objective similarities. In contrast, the conventional MFEA makes pairs of parents randomly without considering the objective similarities. The proposed selection encourages crossovers of parents with skill factors of similar objective functions and suppresses crossovers of parents with skill factors of dissimilar objective functions.

Table 1. MFDMPs used in this paper

Problem name	Number of objectives $K(= \|\mathcal{K}\|)$	Goal positions (objective indices) $\mathcal{K}$
Uniform MFDMP-4	4	{0, 15, 30, 45}
Uniform MFDMP-10	10	{0, 6, 12, 18, 24, 30, 36, 42, 48, 54}
Uniform MFDMP-12	12	{0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55}
Biased MFDMP-A	12	{0, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40}
Biased MFDMP-B	12	{0, 1, 2, 3, 4, 5, 30, 31, 32, 33, 34, 35}
Biased MFDMP-C	12	{0, 1, 2, 15, 16, 17, 30, 31, 32, 45, 46, 47}

5 Experimental Setting

5.1 Algorithm

We compare the search performances of three algorithms: (i) the conventional MFEA [1,2], (ii) the conventional single-objective EA (SOEA), and (iii) the proposed MFEA using the objective similarity based parent selection.

For all algorithms, we use the SBX crossover (the distribution index $\eta_c = 2$ and the crossover ratio 1.0) and the polynomial mutation (the distribution index $\eta_m = 5$ and the mutation rate 0.5). The population size is set to 100, and the total number of generations is set to $G = 80$. Each algorithm is executed 100 times, and we compare the average result.

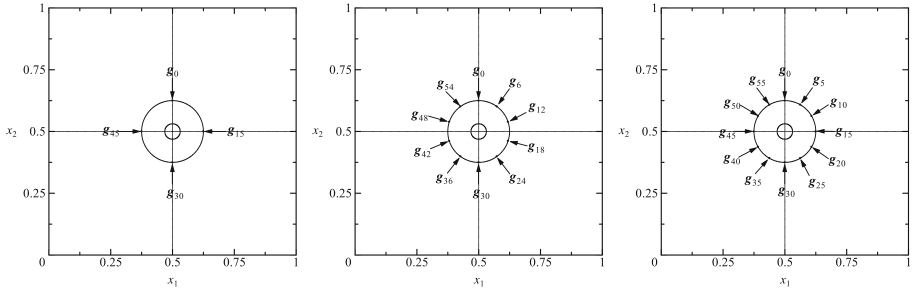
For the conventional MFEA, *rmp* is set to 0.3. The conventional SOEA also uses the single population and assigns a skill factor to each solution. However, any interactions among solutions with different skill factors are not performed. That is, a solution for a skill factor is always crossed with a solution with the same skill factor.

5.2 Metric

As the metric to evaluate the multi-factorial optimization performance, we use D metric [7] given by

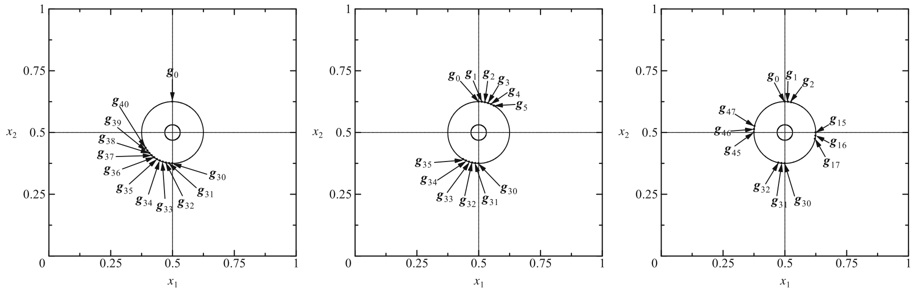
$$D = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \min_{\boldsymbol{x} \in \mathcal{P}} f_k(\boldsymbol{x}). \tag{7}$$

D is the average objective function value of the best objective function values on $K(= |\mathcal{K}|)$ objective functions in the population $\mathcal{P}$. The smaller D, the better multi-factorial optimization performance.



(a) Uniform MFDMP-4 (b) Uniform MFDMP-10 (c) Uniform MFDMP-12

Fig. 3. Uniform MFDMPs



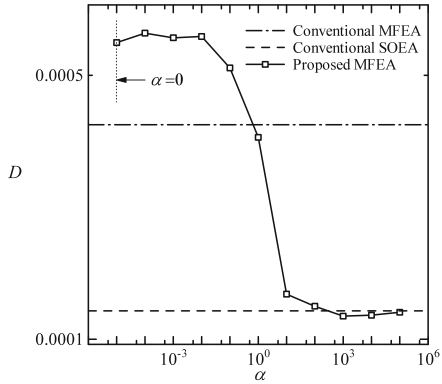
(a) Biased MFDMP-A (b) Biased MFDMP-B (c) Biased MFDMP-C

Fig. 4. Biased MFDMPs

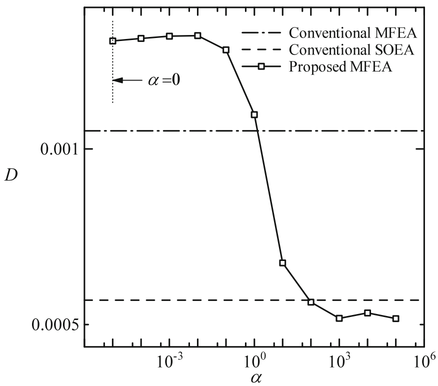
5.3 Problems

For MFDMPs, we use the radius $r = 0.125$ to set goal positions on the circle. Also, we use the radius $\omega = 0.03125$ to set the circle for the solution initialization.

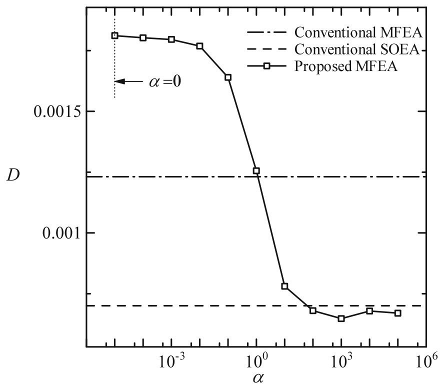
Table 1 shows settings of six MFDMPs used in this work. As mentioned before, the goal-setting circle is equally divided into 60 intervals, such as minute intervals of the analog clock. The first three MFDMPs are uniform MFDMPs, in which the goal positions are uniformly distributed on minute positions. We use the uniform MFDMP-4, -10, and -12 with $K = \{4, 10, 12\}$ goal points, respectively. Goal positions are visually shown in Fig. 3. The other three MFDMPs are biased MFDMPs, in which the goal positions are biasedly distributed on minute positions. We use the biased MFDMP-A, -B, and -C with $K = 12$ goal points. Goal positions are visually shown in Fig. 4. In the biased MFDMP-A, one goal is isolatedly positioned from others. The biased MFDMP-B has two groups of goals, and the two groups are positioned on the opposite side. Each group has six goals, and they are densely distributed. The biased MFDMP-C has four groups of goals, and the four groups are uniformly positioned. Each group has three goals, and they are densely distributed.



(a) Uniform MFDMP-4



(b) Uniform MFDMP-10



(c) Uniform MFDMP-12

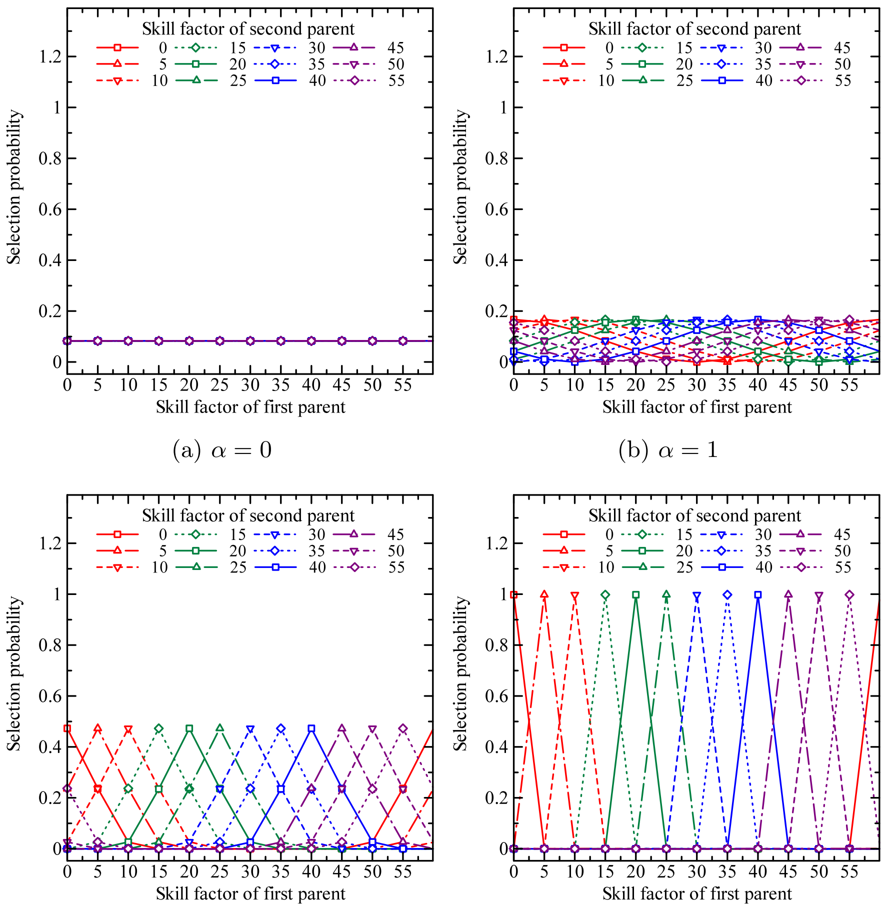
Fig. 5. Results of D on uniform MFDMPs

6 Experimental Results and Discussion

6.1 Results on Uniform MFDMPs

Figure 5 shows results of the proposed algorithm on uniform MFDMPs when we vary the parameter α. The increase of α emphasizes the bias of the selection probabilities based on objective similarities. Note that the horizontal axis is the logarithmic scale, and results with $\alpha = 0$ are exceptionally plotted on $\alpha = 10^{-5}$. Each figure also involves results of the conventional SOEA and MFEA as horizontal lines.

$\alpha = 0$ indicates the random selection without considering the objective similarities. We see that D decreases as α increases from $\alpha = 0$. That is, the selection bias based on objective similarities improves the multi-factorial optimization performance. For three uniform MFDMPs, $\alpha = 10^3$ achieves the best D values, which are smaller than ones of the conventional SOEA and MFEA.



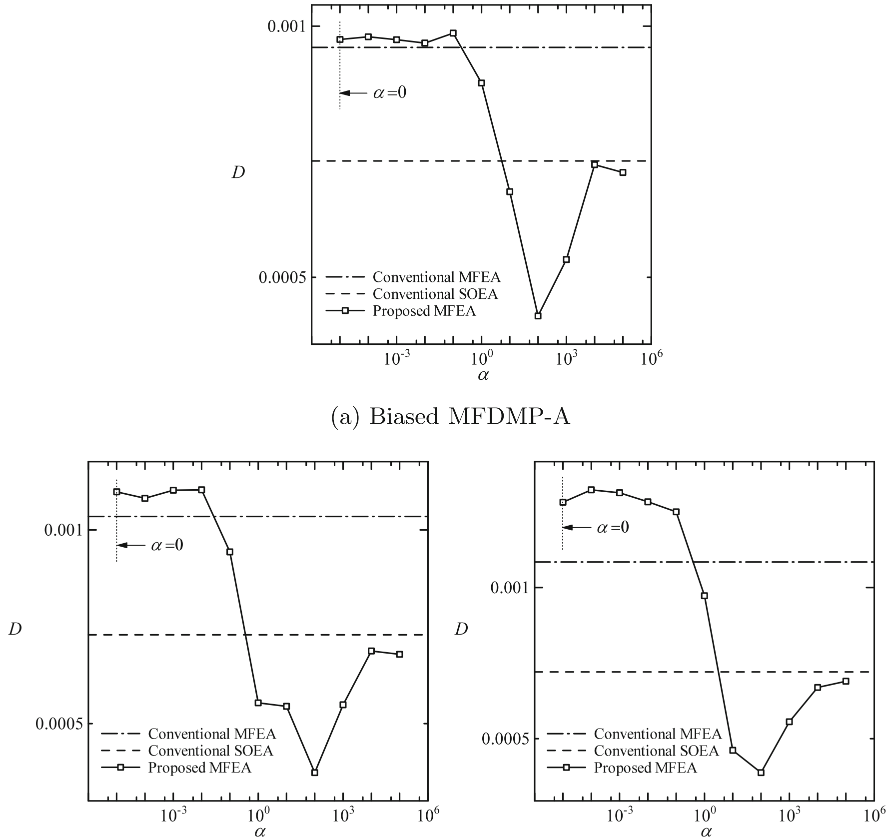
(a) $\alpha = 0$ (b) $\alpha = 1$

(c) $\alpha = 10$ (d) $\alpha = 100$

Fig. 6. Selection probabilities on the uniform MFDMP-K12 (Color figure online)

Since we limited the initialization area of the population, the cooperative search among solutions with different skill factors does not work well on the conventional MFEA. As a result, the conventional MFEA is worse than the conventional SOEA. Even in this problem situation, the cooperative search on the proposed algorithm works well, and the proposed algorithm achieves the best performance by controlling the solution selection bias.

Next, we focus only on the uniform MFDMP-12 and observe the selection probabilities obtained by the proposed algorithm. Figure 6 shows the selection probabilities when we vary the parameter α. The horizontal axis indicates the skill factors of the first parent. Each figure has twelve plots, which are skill factors of second parents. That is, for each skill factor of the first parent on



(a) Biased MFDMP-A

(b) Biased MFDMP-B

(c) Biased MFDMP-C

Fig. 7. Results of D on biased MFDMPs

the horizontal axis, we see that the selection probabilities of skill factors of the second parent.

In Fig. 6(a) with $\alpha = 0$, we see that the selection probabilities for all objective functions are flat, which brings the random parent selection without considering the objective similarities. In Fig. 6(b) with $\alpha = 1$, we see a selection bias based on the objective similarities. Note that the horizontal axis indicates skill factor indices from 0 to 59, which are minute positions of the analog clock shown before. Here, we focus on the solid red line with the rectangle marker showing the selection probabilities of the second parent with the skill factor 0. The highest selection probability can be seen when the first parent is also with the skill factor 0. The second highest probability can be seen when the first parent is with the skill factor 5 or 55, which is the neighborhood of the skill factor 0. Thus, the selection probabilities of the second parent with the skill factor 0 decrease as increasing the distance from the skill factor 0. From Fig. 6(c) and Fig. 6(d), we see that the selection probabilities are biased by increasing α. That is, the making pairs of parents with similar skill factors are emphasized by increasing α.

These results reveal that the control of the selection probabilities based on the objective similarities works well and contributes to improving multi-factorial optimization performance.

6.2 Results on Biased MFDMPs

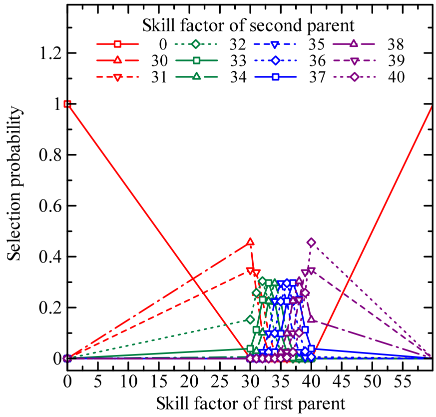
Figure 7 shows results of the proposed algorithm on biased MFDMPs when we vary the parameter α. We see that D decreases by increasing α from $\alpha = 0$. For the three biased MFDMPs, $\alpha = 10^2$ achieves the best search performance, which is better than the conventional SOEA and MFEA. We also see that the further increase of α from $\alpha = 10^2$ deteriorates the search performance. The over-biased selection probability is not appropriate since even the search cooperation among similar objective functions is blocked with too large α.

Figure 8 shows the selection probability of the proposed algorithm with $\alpha = 10^2$. Figure 8(a) shows results on the biased MFDMP-A. As shown in Fig. 4(a), one goal $\boldsymbol{g}_0$ is isolatedly positioned, and others $\boldsymbol{g}_{30}$–$\boldsymbol{g}_{40}$ are closely positioned. From the result, we see that almost all first parents with skill factor 0 targeting goal $\boldsymbol{g}_0$ are paired with second parents with skill factor 0 since other solutions have quite different search directions in the design variable space. On the other hand, we see that solutions with skill factors targeting closely distributed goals $\boldsymbol{g}_{30}$–$\boldsymbol{g}_{40}$ are cooperating.

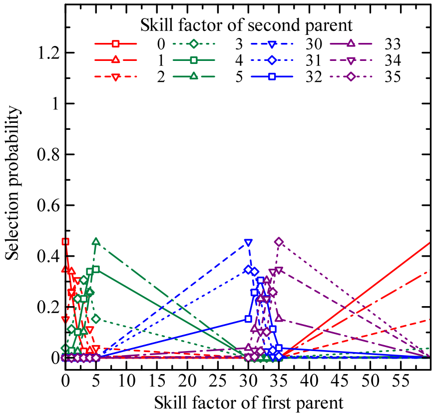
Figure 8(b) shows results on the biased MFDMP-B. As shown in Fig. 4(b), a group of goals $\boldsymbol{g}_0$–$\boldsymbol{g}_5$ and another group of goals $\boldsymbol{g}_{30}$–$\boldsymbol{g}_{35}$ are separately positioned. From the result, we see the parents with different skill factors are frequently paired in each of the closely distributed two groups. However, the probabilities of mating parents from different groups are nearly zero.

Figure 8(c) shows results on the biased MFDMP-C. As shown in Fig. 4(c), there are four groups of goals. The first one involves $\boldsymbol{g}_0$–$\boldsymbol{g}_2$, the second one involves $\boldsymbol{g}_{15}$–$\boldsymbol{g}_{17}$, the third one involves $\boldsymbol{g}_{30}$–$\boldsymbol{g}_{32}$, and the fourth one involves $\boldsymbol{g}_{45}$–$\boldsymbol{g}_{47}$. From the result, we see that the selection probabilities are shared in each group. In other words, the solution resources are shared in each group even if their skill factors are different.

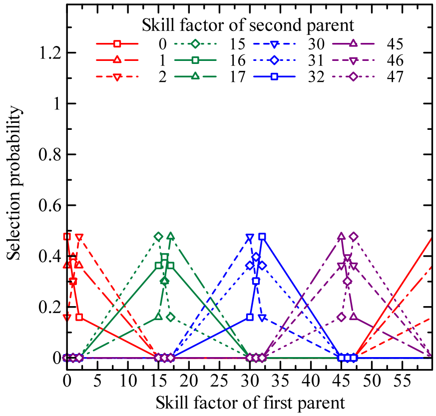
These results verified that the proposed method encouraged pairing for solutions with similar objective functions and suppressed pairing for solutions with dissimilar objective functions. The results also clarified that the proposed similarity-based parent selection improved the multi-factorial optimization performance.



(a) Biased MFDMP-A



(b) Biased MFDMP-B



(c) Biased MFDMP-C

Fig. 8. Selection probabilities on the biased MFDMPs

7 Conclusions

To accelerate the evolutionary multi-factorial optimization by encouraging crossovers of solutions with similar target objective functions and suppressing crossovers of solutions with dissimilar target objective functions, in this work, we proposed an evolutionary algorithm estimating the similarities of objective functions and utilizing them for the parent selection determining pairs of solutions to be crossed. We used the multi-factorial distance minimization problems to verify the effectiveness of the proposed algorithm. Experimental results showed that the distance relations among objective functions are matched to the similarities among objective functions estimated by the proposed method. Also, experimental results showed that the proposed algorithm achieves better search performance than the conventional single-objective and multi-factorial EAs.

As future work, we will address real-world multi-factorial design optimization problems by using the proposed algorithm.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number 19K12135.

References

1. Gupta, A., Ong, Y.S., Feng, L.: Multifactorial evolution: towards evolutionary multitasking. IEEE Trans. Evol. Comput. **20**(3), 343–357 (2015)
2. Tang, J., Chen, Y., Deng, Z., Xiang, Y., Joy, C.P.: A group-based approach to improve multifactorial evolutionary algorithm. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence Main Track, pp. 3870–3876 (2018)
3. Jin, C.C., Tsai, P.W., Qin, A.K.: A study on knowledge reuse strategies in multitasking differential evolution. In: Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC 2019), pp. 1564–1571 (2019)
4. Song, D.H., Qin, A.K., Tsai, P.W., Liang, J.J.: Multitasking multi-swarm optimization. In: Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC 2019), pp. 1937–1944 (2019)
5. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997). https://doi.org/10.1023/A:1008202821328
6. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
7. Kawakami, S., Takagi, T., Takadama, K., Sato, H.: Distance minimization problems for multi-factorial evolutionary optimization benchmarking. In: Abraham, A., Hanne, T., Castillo, O., Gandhi, N., Nogueira Rios, T., Hong, T.-P. (eds.) HIS 2020. AISC, vol. 1375, pp. 710–719. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73050-5_69
8. Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., Nojima, Y.: Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6239, pp. 91–100. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15871-1_10
9. Ishibuchi, H., Yamane, M., Akedo, N., Nojima, Y.: Many-objective and many-variable test problems for visual examination of multiobjective search. In: Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC 2013), pp. 1491–1498 (2013)