Turning Parent Degradation into an Advantage: Archive-Directed Mating for Archive-Based Evolutionary Multi-Objective Optimization

Kazuma Sato*, Minami Miyakawa*, Keiki Takadama†, Hiroyuki Sato*
**The University of Electro-Communications*, Tokyo, Japan
k.sato@uec.ac.jp, m.miyakawa@hs.hc.uec.ac.jp, h.sato@uec.ac.jp
†*The University of Tokyo*, Chiba, Japan
takadama@g.ecc.u-tokyo.ac.jp

***Abstract*—In multi-objective evolutionary algorithms, archives are widely used to preserve non-dominated solutions during the search process. However, introducing an archive can make parent degradation more explicit, where the parent population is dominated by archive solutions, which has traditionally been regarded as an undesirable phenomenon. This degradation, however, implies a clear dominance relationship between parents and archive solutions and can be viewed as informative for search improvement. In this paper, we propose Archive-Directed Mating, which exploits parent degradation as a source of search guidance rather than suppressing it. When the first parent is dominated by archive solutions, the second parent is selected from the archive, directly incorporating information from existing high-quality solutions into mating. Experiments on multi-objective optimization problems with two to six objectives show that the proposed method consistently outperforms conventional mating and re-selection approaches in terms of Hypervolume, particularly in many-objective settings. Further analysis of dominance relations between offspring and archive solutions reveals that the archive-directed mating increases the number of promising offspring while suppressing poor ones. These results demonstrate that parent degradation can be effectively leveraged for search improvement, and that archive-directed mating provides a general and complementary strategy to existing mating and re-selection mechanisms.**

***Index Terms*—multi-objective evolutionary algorithms, non-dominated solution archive, directed mating, parent degradation**

I. Introduction

Evolutionary multi-objective optimization (EMO) is widely used as a representative framework for simultaneously optimizing multiple conflicting objectives in various fields, including engineering design and decision-making. In this field, following a long-standing convention in evolutionary strategies, the $(\mu + \lambda)$ scheme, in which the solution set consists of a parent population of size μ and an offspring population of size λ, is widely adopted. In addition, to preserve non-dominated solutions obtained during the search, it is common to maintain an archive of non-dominated solutions independently of the parent population [1].

While archives are particularly useful in real-world applications with expensive evaluations, maintaining them in parallel with the parent population can cause parent degradation, where parent individuals are dominated by archive solutions [2]–[4]. This phenomenon raises fundamental questions about the role of the parent population and the design of generational updates, and motivates a re-examination of search behavior and design principles in EMO algorithms.

Previous studies have mainly investigated approaches that restore consistency between the parent population and the archive by replacing degraded parents with non-dominated archive solutions [5]–[7]. These methods aim to directly improve parent quality by suppressing or eliminating degradation. However, the fact that degraded parents are dominated by archive solutions also implies the existence of clear improvement directions in the objective space, and thus can be regarded as containing useful information for search improvement. In this sense, parent degradation is not merely a phenomenon to be avoided, but one that can potentially be exploited.

This perspective is consistent with the concept of directed mating [8] in constrained multi-objective optimization. In constrained problems, selecting infeasible solutions that dominate parent individuals as mating partners has been shown to improve search performance [8]. If infeasible solutions are viewed as solutions from another set that dominate parents, a similar idea can be applied regardless of the presence of constraints. Thus, parent degradation caused by archive solutions is structurally analogous to situations where directed mating is effective in constrained problems, suggesting that it may also contribute to search improvement.

Based on this observation, this paper proposes an archive-directed mating method that actively exploits parent degradation as an information source for search improvement rather than suppressing it as a cause of performance deterioration. In the proposed method, the first parent is selected from the parent population using tournament selection as in conventional approaches. If a set of archive solutions dominating the first parent exists, the second parent is selected from this set to reflect the improvement directions suggested by the archive in the crossover. Otherwise, the method falls back to conventional parent selection. The proposed method is embedded into NSGA-II [9] and evaluated on multi-objective

Algorithm 1 Conventional method: Baseline

Input: μ: the size of the parent population $\mathcal{P}$, λ: the size of the offspring population $\mathcal{Q}$, T_{max}: the total number of generations.
Output: $\mathcal{P}$: the parent population.
1: $\mathcal{P} \leftarrow$ Initialization (μ)
2: **for** $t \leftarrow 1, 2, \ldots, T_{max}$ **do**
3: $\quad \mathcal{Q} \leftarrow \emptyset$
4: $\quad$ **for** $i \leftarrow 1, 2, \ldots, \lambda$ **do**
5: $\quad\quad \boldsymbol{p}^1 \leftarrow$ Parent Selection $(\mathcal{P})$
6: $\quad\quad \boldsymbol{p}^2 \leftarrow$ Parent Selection $(\mathcal{P})$
7: $\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup \boldsymbol{q} \leftarrow$ Crossover and Mutation $(\boldsymbol{p}^1, \boldsymbol{p}^2)$
8: $\quad$ **end for**
9: $\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q}, \mu)$
10: **end for**

knapsack problems [10], where its optimization performance and search behavior are compared with conventional mating and re-selection-based approaches.

II. Evolutionary Multi-Objective Optimization

A. Multi-Objective Optimization Problem

Given a decision vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{S}$ in the decision space $\mathcal{S}$, a multi-objective optimization problem is defined as

$$\text{Minimize} \quad f_i(\boldsymbol{x}) \quad (i = 1, 2, \ldots, M), \tag{1}$$

where f_i $(i = 1, 2, \ldots, M)$ denotes the i-th objective function. For two solutions $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$, $\boldsymbol{x}$ is said to dominate $\boldsymbol{y}$ $(\boldsymbol{x} \preceq \boldsymbol{y})$ if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ for all $i \in \{1, 2, \ldots, M\}$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ for at least one $j \in \{1, 2, \ldots, M\}$. For a solution set $X \subseteq \mathcal{S}$, the set $\{\boldsymbol{x} \in X \mid \nexists \boldsymbol{y} \in X : \boldsymbol{y} \preceq \boldsymbol{x}\}$ is called the non-dominated solution set. The set of non-dominated solutions in $\mathcal{S}$ is referred to as the Pareto optimal set, $\mathcal{PS} = \{\boldsymbol{x} \in \mathcal{S} \mid \nexists \boldsymbol{y} \in \mathcal{S} : \boldsymbol{y} \preceq \boldsymbol{x}\}$. The Pareto front is defined as $\mathcal{PF} = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{PS}\}$. The goal of multi-objective optimization is to approximate the Pareto front with a finite set of solutions.

B. Multi-Objective Evolutionary Algorithm

The pseudocode of the $(\mu + \lambda)$-based EMO algorithm used as the baseline in this paper is shown in **Alg. 1**. The parent population $\mathcal{P}$ has size μ, and the offspring population $\mathcal{Q}$ has size λ, where $\mu = \lambda$ is commonly assumed. In line 1, the parent population $\mathcal{P}$ is initialized. Lines 2–10 are repeated for each generation t. In lines 4–8, the offspring population $\mathcal{Q}$ is generated. In lines 5 and 6, the first and second parents for crossover are selected. In a representative algorithm such as NSGA-II [9], tournament selection is used. In line 9, environmental selection is applied to $\mathcal{P} \cup \mathcal{Q}$ to obtain the parent population $\mathcal{P}$ for the next generation. In NSGA-II [9], the combined population $\mathcal{P} \cup \mathcal{Q}$ is first sorted into fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ based on non-dominance. Solutions within the same front are ranked using the crowding distance, which is computed based on distances to neighboring solutions. Based on the front rank and crowding distance, the parent population $\mathcal{P}$ for the next generation is selected. Finally, the parent population $\mathcal{P}$ is output.

Algorithm 2 Conventional method: Archive

Input: μ: the size of the parent population $\mathcal{P}$, λ: the size of the offspring population $\mathcal{Q}$, γ: the size of the non-dominated solution archive $\mathcal{A}$, T_{max}: the total number of generations.
Output: $\mathcal{A}$: the non-dominated solution archive.
$\mathcal{P} \leftarrow$ Initialization (μ)
$\mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{P}, \gamma)$
for $t \leftarrow 1, 2, \ldots, T_{max}$ **do**
$\quad \mathcal{Q} \leftarrow \emptyset$
$\quad$ **for** $i \leftarrow 1, 2, \ldots, \lambda$ **do**
$\quad\quad \boldsymbol{p}^1 \leftarrow$ Parent Selection $(\mathcal{P})$
$\quad\quad \boldsymbol{p}^2 \leftarrow$ Parent Selection $(\mathcal{P})$
$\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup \boldsymbol{q} \leftarrow$ Crossover and Mutation $(\boldsymbol{p}^1, \boldsymbol{p}^2)$
$\quad$ **end for**
$\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q}, \mu)$
$\quad \mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{A} \cup \mathcal{Q}, \gamma)$
end for

III. Related Work

A. Non-Dominated Solution Archive

In EMO algorithms, a non-dominated solution archive $\mathcal{A}$ is widely used to prevent the loss of non-dominated solutions during generational updates and to maintain the approximation quality of the Pareto front [1]. The pseudocode with an archive is shown in **Alg. 2**, where non-dominated solutions are extracted from the parent population $\mathcal{P}$ and stored in $\mathcal{A}$, and the archive is updated using the offspring population $\mathcal{Q}$. In general, the archive size γ is set larger than μ and λ. In this paper, when $|\mathcal{A}| > \gamma$, the archive is truncated by NSGA-II environmental selection.

Archive methods are commonly classified into size-limited and size-unlimited approaches [1], [11]. In addition, archive usage can be categorized into methods that actively exploit the archive in the optimization process [5]–[7] and those that use it only for performance evaluation [12]–[14]. This paper focuses on size-limited archives that are actively used for optimization.

Despite their advantages, the introduction of archives makes parent degradation more explicit, where part of the parent population $\mathcal{P}$ is dominated by archive solutions [2]–[4]. An example is illustrated in **Fig. 1**. When parent degradation occurs, offspring are generated from inferior parents, implying that the parent population does not always maintain the best solutions and raising concerns about the design of generational updates.

To mitigate this issue, re-selection methods that reconstruct the parent population $\mathcal{P}$ from the archive $\mathcal{A}$ when domination occurs have been proposed [5]–[7], and are reviewed in IV. While re-selection improves parent quality and optimization performance, existing approaches mainly aim to suppress degradation. The exploitation of the dominance relationship between degraded parents and archive solutions as search guidance has not yet been sufficiently explored.

B. Directed Mating

This subsection reviews the concept of directed mating in constrained optimization as the theoretical background of the archive-directed mating proposed later. Directed mating has

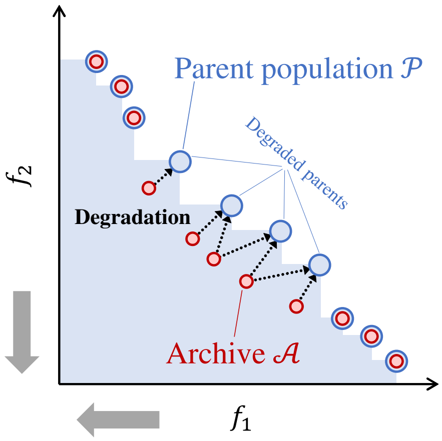


Fig. 1: Parent degradation

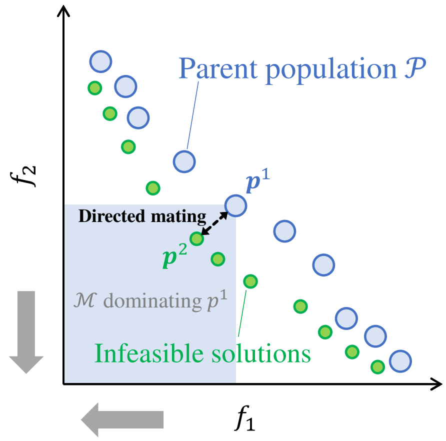


Fig. 2: Directed mating [8]

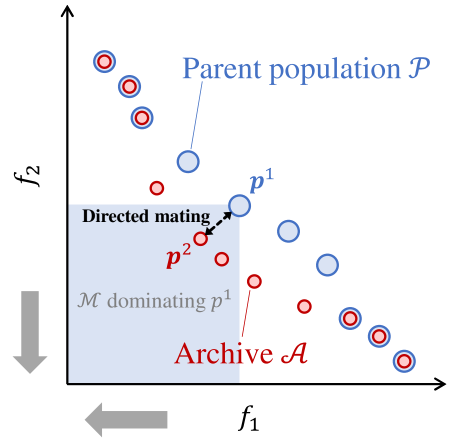


Fig. 3: Proposed archive-directed mating

been proposed as a parent selection strategy for constrained multi-objective optimization problems [8]. Instead of simply discarding infeasible solutions, directed mating actively exploits them when they have better objective values than feasible solutions, and reflects the implied improvement directions in the objective space through crossover.

An example of directed mating is shown in **Fig. 2**. When selecting a pair of parents for crossover, the first parent $\boldsymbol{p}^1$ is selected from the parent population $\mathcal{P}$. Next, a solution set $\mathcal{M}$ that dominates $\boldsymbol{p}^1$ is extracted. A key feature is that $\mathcal{M}$ may include infeasible solutions. If $\mathcal{M}$ is non-empty, the second parent $\boldsymbol{p}^2$ is selected from $\mathcal{M}$; otherwise, $\boldsymbol{p}^2$ is selected from $\mathcal{P}$. By crossing $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$, the improvement directions suggested by solutions dominating $\boldsymbol{p}^1$ are reflected in offspring generation, thereby enhancing convergence.

Although directed mating has mainly been studied for constrained problems, situations where infeasible solutions dominate feasible ones are structurally similar to parent degradation, where parent individuals are dominated by archive solutions. In both cases, a set of solutions superior to the parent exists, and their dominance relations can be used to infer improvement directions. This similarity suggests that archive solutions can play a role analogous to infeasible solutions in directed mating.

Based on this observation, as illustrated in **Fig. 3**, this paper extends the idea of directed mating to unconstrained multi-objective optimization and investigates archive-directed mating, which actively exploits the dominance relationship between degraded parents and archive solutions to reflect improvement directions in crossover.

IV. Conventional Archive Re-Selection

A. Overview

The archive re-selection method maintains the quality of the parent population by replacing degraded parent individuals with non-dominated solutions from the archive, thereby avoiding situations where parents are dominated by archive solutions. This subsection reviews the archive-based re-selection method proposed in [7].

Algorithm 3 Conventional method: Archive with Re-selection

Input: μ: the size of the parent population $\mathcal{P}$, λ: the size of the offspring population $\mathcal{Q}$, γ: the size of the non-dominated solution archive $\mathcal{A}$, T_{max}: the total number of generations, r: the interval of generations to re-select the parent population $\mathcal{P}$ from the archive $\mathcal{A}$.
Output: $\mathcal{A}$: the non-dominated solution archive.

1: $\mathcal{P} \leftarrow$ Initialization (μ)
2: $\mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{P}, \gamma)$
3: **for** $t \leftarrow 1, 2, \dots, T_{max}$ **do**
4: $\quad \mathcal{Q} \leftarrow \emptyset$
5: $\quad$ **for** $i \leftarrow 1, 2, \dots, \lambda$ **do**
6: $\quad\quad \boldsymbol{p}^1 \leftarrow$ Parent Selection $(\mathcal{P})$
7: $\quad\quad \boldsymbol{p}^2 \leftarrow$ Parent Selection $(\mathcal{P})$
8: $\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup \boldsymbol{q} \leftarrow$ Crossover and Mutation $(\boldsymbol{p}^1, \boldsymbol{p}^2)$
9: $\quad$ **end for**
10: $\quad$ **if** $t \bmod r = 0$ **then**
11: $\quad\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q} \cup \mathcal{A}, \mu)$
12: $\quad$ **else**
13: $\quad\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q}, \mu)$
14: $\quad$ **end if**
15: $\quad \mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{A} \cup \mathcal{Q}, \gamma)$
16: **end for**

B. Method

The block diagram is shown in **Fig. 4**, and the pseudocode is given in **Alg. 3**. The baseline is **Alg. 2**, and the differences are highlighted in red. The re-selection interval is controlled by a parameter r. In line 10, if generation t is a re-selection generation, environmental selection is applied in line 11 to the union $\mathcal{P} \cup \mathcal{Q} \cup \mathcal{A}$ to construct the next parent population $\mathcal{P}$. This directly avoids parent degradation, where parent individuals are dominated by archive solutions [2]–[4]. Otherwise, in line 13, environmental selection is applied to $\mathcal{P} \cup \mathcal{Q}$, as in **Alg. 1** and **Alg. 2**.

The re-selection method is effective in reducing inconsistency between the parent population $\mathcal{P}$ and the archive $\mathcal{A}$,

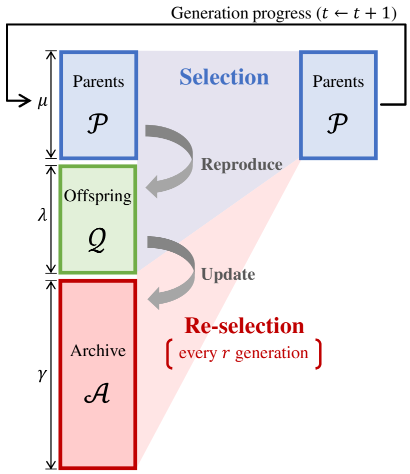


Fig. 4: Archive with re-selection

and has been reported to improve optimization performance [7].

V. Proposed Method: Archive-Directed Mating

A. Overview

This paper proposes archive-directed mating, which exploits situations where parent individuals are dominated by archive solutions as a useful source of information for search improvement. The key idea is to focus on the relationship between degraded parents and the archive solutions that dominate them, and to reflect the improvement directions implied by this dominance relationship in crossover. In this way, the dominance structure between degraded parents and the archive, which has not been sufficiently utilized so far, is actively incorporated into the search process.

B. Method

The pseudocode is shown in **Alg. 4**. The baseline is **Alg. 3**, and the differences are highlighted in blue.

In lines 6 and 7, the first parent $\boldsymbol{p}^1$ and the second parent $\boldsymbol{p}^2$ are selected in the same manner as in **Alg. 1–3**. In lines 8–13, archive-directed mating is attempted when the non-dominated archive $\mathcal{A}$ has reached its predefined size γ, at which point parent degradation may occur. The condition $|\mathcal{A}| = \gamma$ is used because parent degradation becomes explicit after the archive is sufficiently filled. In line 9, a solution set $\mathcal{M}$ that dominates the first parent $\boldsymbol{p}^1$ is extracted from the archive $\mathcal{A}$. If $\mathcal{M}$ is non-empty, indicating that parent degradation has occurred, the second parent $\boldsymbol{p}^2$ is replaced with a solution from $\mathcal{M}$ in line 11. This constitutes the parent selection mechanism of the proposed archive-directed mating.

An illustrative example is shown in **Fig. 3**. Suppose that the blue solution $\boldsymbol{p}^1$ is selected as the first parent. From the red archive $\mathcal{A}$, two red non-dominated solutions that dominate $\boldsymbol{p}^1$ are extracted as $\mathcal{M}$. One of these solutions is then selected as the second parent $\boldsymbol{p}^2$. Archive-directed mating thus performs crossover between the degraded first parent $\boldsymbol{p}^1$ and the dominating archive solution $\boldsymbol{p}^2$.

Algorithm 4 Proposed method: Archive with Re-selection and Archive-Directed Mating

Input: μ: the size of the parent population $\mathcal{P}$, λ: the size of the offspring population $\mathcal{Q}$, γ: the size of the non-dominated solution archive $\mathcal{A}$, T_{max}: the total number of generations, r: the interval of generations to re-select the parent population $\mathcal{P}$ from the archive $\mathcal{A}$.
Output: $\mathcal{A}$: the non-dominated solution archive.

1: $\mathcal{P} \leftarrow$ Initialization (μ)
2: $\mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{P}, \gamma)$
3: **for** $t \leftarrow 1, 2, \dots, T_{max}$ **do**
4: $\quad \mathcal{Q} \leftarrow \emptyset$
5: $\quad$ **for** $i \leftarrow 1, 2, \dots, \lambda$ **do**
6: $\quad\quad \boldsymbol{p}^1 \leftarrow$ Parent Selection $(\mathcal{P})$
7: $\quad\quad \boldsymbol{p}^2 \leftarrow$ Parent Selection $(\mathcal{P})$
8: $\quad\quad$ **if** $|\mathcal{A}| = \gamma$ **then**
9: $\quad\quad\quad \mathcal{M} \leftarrow \{ a \in \mathcal{A} \mid \boldsymbol{a} \preceq \boldsymbol{p}^1 \}$
10: $\quad\quad\quad$ **if** $|\mathcal{M}| \geq 1$ **then**
11: $\quad\quad\quad\quad \boldsymbol{p}^2 \leftarrow$ Parent Selection $(\mathcal{M})$
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end if**
14: $\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup \boldsymbol{q} \leftarrow$ Crossover and Mutation $(\boldsymbol{p}^1, \boldsymbol{p}^2)$
15: $\quad$ **end for**
16: $\quad$ **if** $t \bmod r = 0$ **then**
17: $\quad\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q} \cup \mathcal{A}, \mu)$
18: $\quad$ **else**
19: $\quad\quad \mathcal{P} \leftarrow$ Environmental Selection $(\mathcal{P} \cup \mathcal{Q}, \mu)$
20: $\quad$ **end if**
21: $\quad \mathcal{A} \leftarrow$ Extract non-dominated solutions $(\mathcal{A} \cup \mathcal{Q}, \gamma)$
22: **end for**

It should be noted that **Alg. 4** also includes the re-selection mechanism. When $r = 1$, the optimality of the parent population $\mathcal{P}$ is guaranteed at every generation, and no parent is dominated by the archive $\mathcal{A}$. In this case, archive-directed mating does not occur, and the algorithm becomes equivalent to the conventional re-selection method [7]. When $r > 1$, opportunities for archive-directed mating arise. As r decreases, the parent population $\mathcal{P}$ is re-selected from the archive more frequently, resulting in fewer archive-directed mating events. Conversely, larger values of r increase the likelihood that parents are dominated by the archive, leading to more frequent archive-directed mating. When $r > T_{max}$, re-selection never occurs, and only the effect of archive-directed mating can be observed.

VI. Experimental Setup

A. Problem

The multi-objective knapsack problem was used. Following [10], a binary decision vector $\boldsymbol{x} \in \mathcal{S} = \{0,1\}^N$ was considered, together with a profit vector $\boldsymbol{p}_m = (p_{1,m}, p_{2,m}, \dots, p_{N,m})$ $(m = 1, 2, \dots, M)$ and a weight vector $\boldsymbol{w}_m = (w_{1,m}, w_{2,m}, \dots, w_{N,m})$ $(m = 1, 2, \dots, M)$. The objective functions were originally formulated as maximization problems, but were transformed into minimization form and defined as $f_m(\boldsymbol{x}) = \sum_{i=1}^{N} (p_{i,m} - p_{i,m} x_i)$ $(m = 1, 2, \dots, M)$. The constraint was given by $g_m(\boldsymbol{x}) = \sum_{i=1}^{N} w_{i,m} x_i \leq 0.5 \sum_{i=1}^{N} w_{i,m}$. Infeasible solutions were repaired using the method in [10], and the problem was

TABLE I: Experimental methods

	Archive (Sec. III-A)	**Re-selection** (Sec. IV-B)	**Archive-Directed Mating** (Sec. V-B)
Method 1 (**Alg. 2** or **3** with $r > T_{max}$)	✓	-	-
Method 2 (**Alg. 3** with $r < T_{max}$)	✓	✓	-
Method 3 (**Alg. 4** with $r > T_{max}$)	✓	-	✓
Method 4 (**Alg. 4** with $r < T_{max}$)	✓	✓	✓

therefore treated as effectively unconstrained. The number of variables was set to $N = 100$, and the number of objectives was $M \in \{2, 3, 4, 5, 6\}$.

B. Methods

Based on NSGA-II [9], four methods listed in **Table I** were examined. For Methods 2 and 4, the re-selection interval was varied as $r \in \{1, 2, 5, 10, 100\}$.

The algorithms were implemented using PlatEMO [15]. For each number of objectives $M \in \{2, 3, 4, 5, 6\}$, the sizes of the parent population $\mathcal{P}$ and the offspring population $\mathcal{Q}$ were set to $\mu, \lambda \in \{201, 210, 220, 210, 252\}$. The archive size was set to $\gamma = 600$, and the maximum number of generations was $T_{max} = 2000$. Offspring were generated using uniform crossover (crossover rate 1.0, bit exchange rate 0.5) and bit-flip mutation (mutation rate $1/N$). Each algorithm was executed 31 times with different random seeds.

C. Performance Metric

To evaluate both convergence and diversity of the obtained solution sets with respect to the Pareto front, Hypervolume (HV) [10] was employed as a comprehensive performance metric. For the reference point $\boldsymbol{R} = (R_1, R_2, \ldots, R_M)$, we used $R_m = 0.6 \sum_{i=1}^{n} p_{i,m}$ $(m = 1, 2, \ldots, M)$, which ensures that all non-dominated solutions obtained at the final generation contribute to the HV value.

In addition, Norm [16] was used as a convergence metric, and MS [17] was used as a diversity metric. Although larger values of Norm and MS indicate better performance, these metrics are sensitive to the shape of the Pareto front. Therefore, they were used only as auxiliary indicators to observe the behavior of the solution sets.

VII. Experimental Results and Discussion

A. Overall Performance (HV)

The HV values of the archive $\mathcal{A}$ at the final generation are shown in **Fig. 5**. The four methods are distinguished by different colors. Methods 2 and 4 include multiple results obtained with different re-selection intervals r. As noted above, the results of Method 2 ($r = 1$, red) and Method 4 ($r = 1$, blue) coincide.

First, we examine the effect of re-selection by comparing Method 1 (black) and Method 2 (red). Method 2 tends to achieve higher HV than Method 1 as the re-selection interval r decreases, i.e., as re-selection is performed more frequently. This tendency becomes more pronounced as the number of objectives increases, indicating that re-selection contributes to improved optimization performance.

Next, the effect of archive-directed mating is investigated by comparing Method 1 (black) and Method 3 (green). Method 3, which employs archive-directed mating, tends to achieve higher HV than Method 1 for problems with $M > 2$. This result suggests that archive-directed mating can be a contributing factor to performance improvement.

We then compare Method 2 (red) and Method 3 (green) to examine the relative effects of re-selection and archive-directed mating. For $M = 2$ and $M = 3$, Method 2 with the r that maximizes HV outperforms Method 3. In contrast, for $M = 4, 5, 6$, Method 3 achieves higher HV than Method 2 with the best r. These results indicate that both mechanisms are effective, with re-selection being more influential for fewer objectives and archive-directed mating becoming relatively more effective as the number of objectives increases.

Finally, we evaluate the combined effect using Method 4 (blue), which integrates re-selection and archive-directed mating. For $M = 4, 5$, Method 4 with the r that maximizes HV achieves the highest HV among all compared methods. The optimal value was $r = 100$, which is likely because a larger r increases opportunities for archive-directed mating. Moreover, Method 4 with $r = 100$ tends to outperform Method 3, which does not employ re-selection. This suggests that while archive-directed mating is effective on its own, combining it with re-selection can further improve optimization performance.

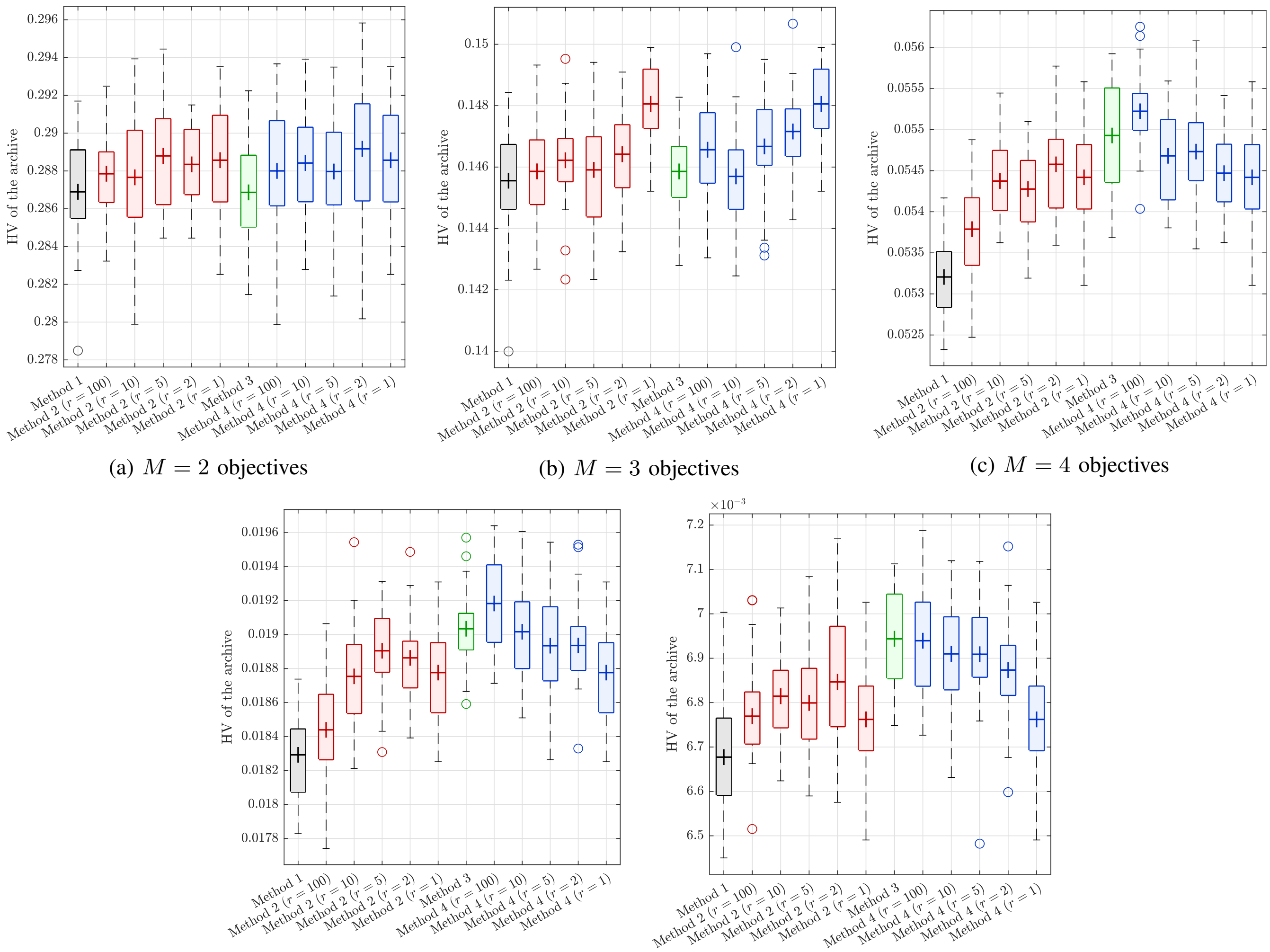
The median HV over generations for the archive $\mathcal{A}$ is shown in **Fig. 6**. Comparing Methods 2 and 4 with the same r, Method 4 generally maintains higher HV throughout the search. This confirms that archive-directed mating contributes to performance improvement over a wide range of generations. Furthermore, for problems with $M > 3$, Method 4 with $r = 100$ shows higher median HV than Method 2 with $r = 1$. While Method 2 with $r = 1$ avoids parent degradation by re-selecting parents every generation, Method 4 with $r = 100$ allows parent degradation to occur and exploits degraded parents through archive-directed mating. These results indicate that intentionally using degraded parents, rather than always selecting only highly optimal parents, can lead to improved optimization performance.

B. Search Dynamics Analysis

We focus on the $M = 5$ objective problem to analyze how archive-directed mating influences the search process.

1) Effect on Convergence and Diversity: The generational transitions of the median Norm (convergence) and MS (diversity) values of the archive $\mathcal{A}$ are shown in **Fig. 7 (a)** and **Fig. 7 (b)**, respectively. Here, we focus on Method 1, Method 2 with $r = 100$, and Method 4 with $r = 100$.

As shown in **Fig. 7 (a)**, Method 4 with archive-directed mating consistently achieves higher Norm values than Method 1 and Method 2, which do not employ archive-directed mating. In contrast, **Fig. 7 (b)** shows that all three methods maintain similar MS values. These results indicate that archive-directed mating improves convergence without sacrificing diversity.



(a) $M = 2$ objectives

(b) $M = 3$ objectives

(c) $M = 4$ objectives

(d) $M = 5$ objectives

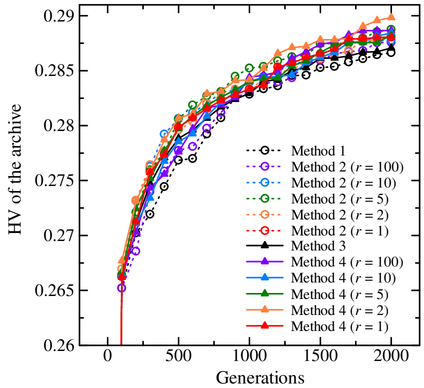
(e) $M = 6$ objectives

Fig. 5: HV values of the archive at the final generation over 31 independent runs.

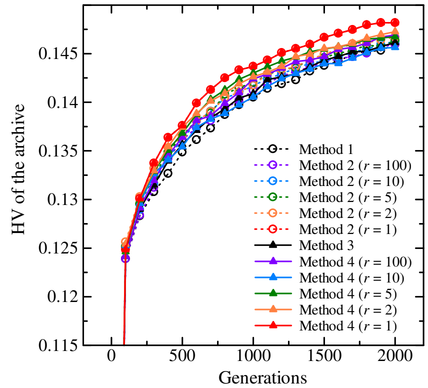
2) Parent–Archive Interaction Analysis: Next, we analyze differences in search behavior by focusing on the interaction between the parent population and the archive. **Fig. 7 (c)** shows the generational transition of the median execution ratio of conventional mating, where the second parent $\boldsymbol{p}_2$ is selected from the parent population $\mathcal{P}$, and archive-directed mating, where $\boldsymbol{p}_2$ is selected from the archive $\mathcal{A}$. Note that the horizontal axis is shown on a logarithmic scale. For Method 1 and Method 2, only conventional mating is executed in all generations, and the execution ratio remains at 1.0. For Method 4, the results with $r = 100$ are shown.

As shown in **Fig. 7 (c)**, archive-directed mating begins to be executed after approximately 50 generations in Method 3 and Method 4. This occurs because the non-dominated archive $\mathcal{A}$ reaches its size limit γ, and situations arise in which the first parent $\boldsymbol{p}_1$ is dominated by archive solutions. In Method 3, the execution ratio of archive-directed mating increases with generations but saturates at around 20%, indicating that the parent population $\mathcal{P}$ stabilizes without being fully dominated by the archive. In Method 4 ($r = 100$), the execution ratio of archive-directed mating temporarily drops to zero every 100 generations. This is because re-selection reconstructs the parent population $\mathcal{P}$ from the archive, eliminating parents dominated by archive solutions. However, this drop is temporary, and the archive-directed-mating execution ratio rapidly increases after re-selection. The maximum archive-directed-mating execution ratios of Method 3 and Method 4 ($r = 100$) are nearly identical.

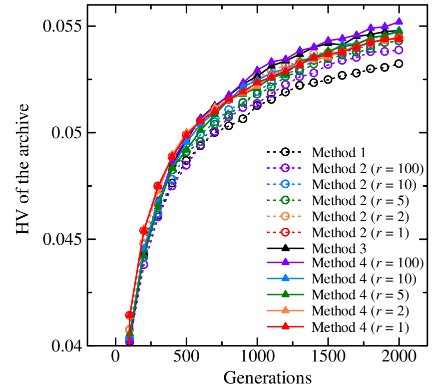
We further compare Method 1 and Method 4 in terms of the relationship between the archive $\mathcal{A}$ and the offspring population $\mathcal{Q}$, as shown in **Fig. 7 (d)–(f)**. **Fig. 7 (d)** shows the ratio of offspring that dominate archive solutions. For offspring generated by conventional mating, no clear difference is observed between Method 1 and Method 4. In contrast, offspring generated by archive-directed mating in Method 4 appear after approximately 50 generations and show a higher probability of dominating archive solutions than offspring generated by conventional mating. This result suggests that archive-directed mating produces offspring with better convergence properties than conventional mating.



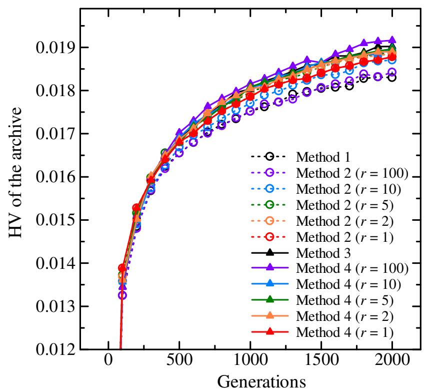
(a) $M = 2$ objectives



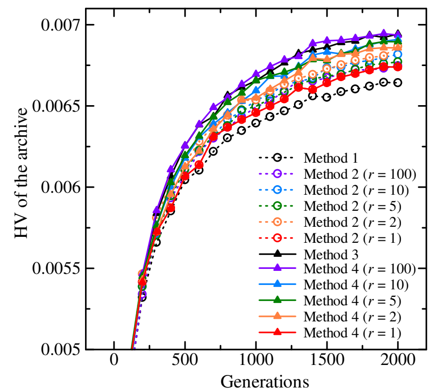
(b) $M = 3$ objectives



(c) $M = 4$ objectives



(d) $M = 5$ objectives



(e) $M = 6$ objectives

Fig. 6: Median HV values of the archive over generations over 31 runs.

Fig. 7 (e) shows the ratio of offspring that are non-dominated with respect to archive solutions. For offspring generated by conventional mating, Method 4 yields a higher ratio than Method 1. Moreover, offspring generated by archive-directed mating show a higher ratio of non-dominated solutions than offspring generated by conventional mating. **Fig. 7 (f)** shows the ratio of offspring dominated by archive solutions. For offspring generated by conventional mating, Method 4 suppresses the generation of dominated offspring compared with Method 1. For offspring generated by archive-directed mating, this ratio is further reduced compared with conventional mating.

These results demonstrate that archive-directed mating increases the number of promising offspring that dominate or are non-dominated with respect to archive solutions, while suppressing the generation of inferior offspring dominated by the archive. In addition, it improves convergence while maintaining diversity. Overall, these findings indicate that archive-directed mating provides an effective and stable search strategy for archive-based EMO algorithms.

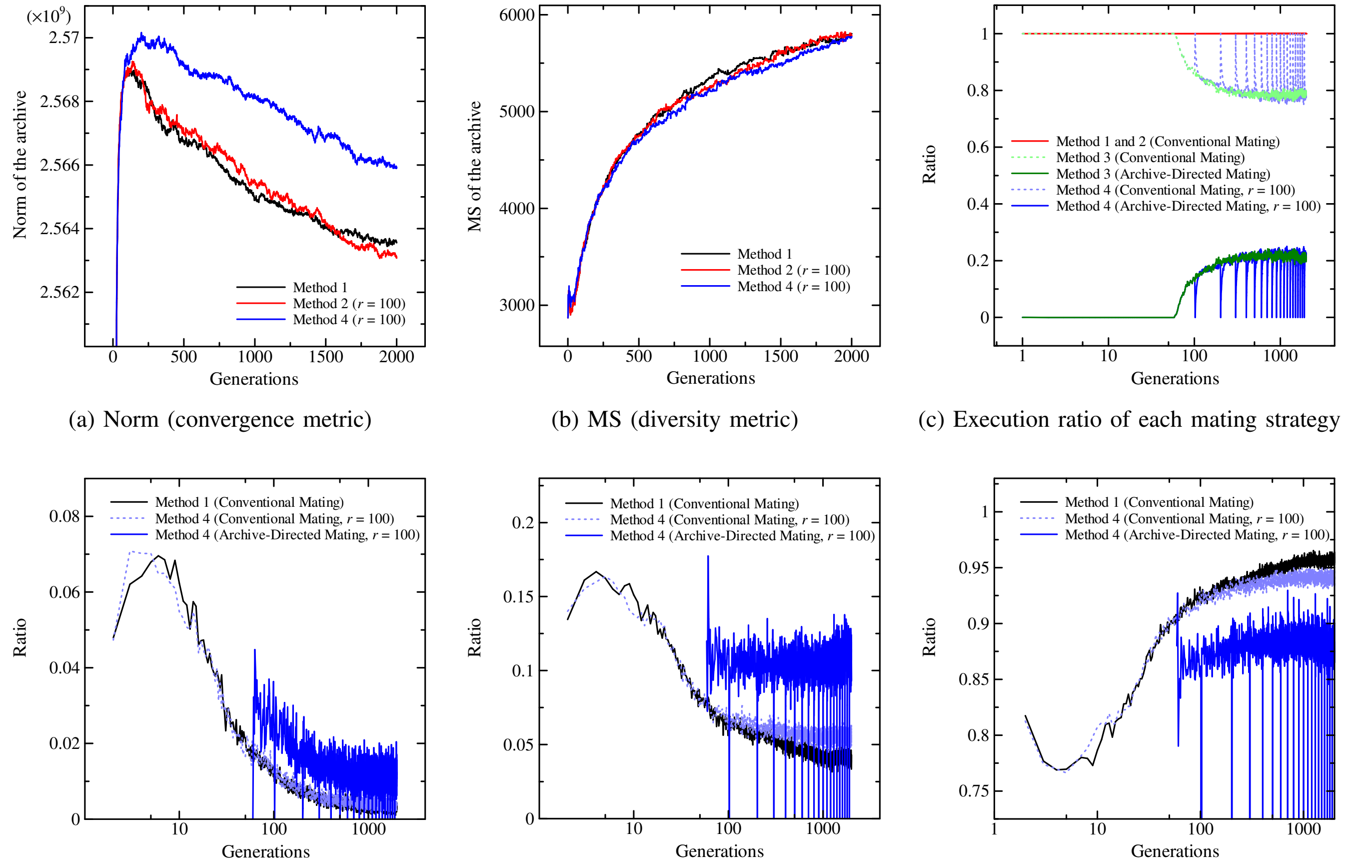
VIII. Conclusion

This paper proposed archive-directed mating, which exploits parent degradation—traditionally regarded as a cause of performance deterioration—as a useful source of information for search improvement in EMO algorithms. When the first parent is dominated by archive solutions, the proposed method selects the second parent from the archive, directly reflecting information from high-quality solutions in crossover.

Numerical experiments on problems with $M = 2$ to $M = 6$ objectives showed that archive-directed mating consistently achieves better performance in terms of HV, especially for many-objective problems, compared with conventional mating and re-selection-based methods. These results indicate that archive-directed mating is not a substitute for re-selection, but a complementary strategy that further enhances optimization performance.

Furthermore, analysis of search dynamics revealed that archive-directed mating increases the number of promising offspring that dominate or are non-dominated with respect to archive solutions, while suppressing the generation of inferior offspring. This demonstrates that parent degradation can be effectively exploited as an opportunity for search improvement.

Overall, archive-directed mating provides a generally applicable strategy for archive-based EMO algorithms and offers a new perspective on mating design and re-selection mechanisms. Future work includes applying the proposed method to other archive-based algorithms and examining its effectiveness on continuous multi-objective optimization problems.



(a) Norm (convergence metric)

(b) MS (diversity metric)

(c) Execution ratio of each mating strategy

(d) Ratio of offspring dominating archive solutions

(e) Ratio of offspring non-dominated with respect to archive solutions

(f) Ratio of offspring dominated by archive solutions

Fig. 7: Median search dynamics over 31 runs on the $M = 5$-objective problem.

References

[1] O. Schütze and C. Hernández, *Archiving Strategies for Evolutionary Multi-objective Optimization Algorithms*, Springer Cham, 2022.

[2] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," Evolutionary Computation Journal, Vol. 10, Issue 3, pp. 263–282, 2002.

[3] M. Li and X. Yao, "An Empirical Investigation of the Optimality and Monotonicity Properties of Multiobjective Archiving Methods," in *Proc. of Evolutionary Multi-Criterion Optimization (EMO2019)*, Springer, Vol. 11411, pp. 15–26, 2019.

[4] H. Ishibuchi, L. M. Pang and K. Shang, "Difficulties in Fair Performance Comparison of Multi-Objective Evolutionary Algorithms," *IEEE Computational Intelligence Magazine*, Vol. 17, No. 1, pp. 86–101, 2022.

[5] H. Mori and A. Oyama, "Multi-Objective Evolutionary Algorithm Fully Based on Unbounded Archive for Problems Requiring Very Expensive Solution Evaluations", *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 111–118, 2022.

[6] N. Okumura, Y. Ohta and H. Sato, "Solution Archive and Its Re-use in Evolutionary Many-objective Facility Control Optimization," *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS2022)*, pp. 1–7, 2022.

[7] K. Sato, N. Okumura, K. Takadama, and H. Sato, "Should Multi-objective Evolutionary Algorithms Use Always Best Non-dominated Solutions as Parents?," *2024 IEEE World Congress on Computational Intelligence (IEEE WCCI 2024)*, 2024.

[8] M. Miyakawa, K. Takadama, and H. Sato, "Two-Stage Non-Dominated Sorting and Directed Mating for Solving Problems with Multi-Objectives and Constraints," *Proc. of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*, pp. 647–654, 2013.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, Issue 2, pp. 182–197, 2002.

[10] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, Issue 4, pp. 257–271, 1999.

[11] R. Tanabe, H. Ishibuchi and A. Oyama, "Benchmarking Multi-and Many-Objective Evolutionary Algorithms Under Two Optimization Scenarios," *IEEE Access*, Vol. 5, pp. 19597–19619, 2017.

[12] Q. Zhang, W. Liu and H. Li, "The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," *in 2009 IEEE Congress on Evolutionary Computation*, pp. 203–208, 2009.

[13] M. Lopez-Ibanez, J. D. Knowles and M. Laumanns, "On Sequential Online Archiving of Objective Vectors," in *Proc. of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011)*, Vol. 6576, pp. 46–60, 2011.

[14] H. Ishibuchi, Y. Setoguchi, H. Masuda and Y. Nojima, "How to Compare Many-Objective Algorithms Under Different Settings of Population and Archive Sizes," in Proc. of *the 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, pp. 1149–1156, 2016.

[15] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization," *IEEE Computational Intelligence Magazine*, Vol. 12, Issue 4, pp. 73–87, 2017.

[16] M. Sato, H.E. Aguirre and K. Tanaka, "Effects of δ-Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm," in *Proc. of the 2006 IEEE International Conference on Evolutionary Computation (CEC 2006)*, pp. 1164–1171, 2006.

[17] E. Zitzler, L. Thiele, and K. Deb, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 2, pp. 173–195, 2000.