2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Analysis of Partition Methods for Dominated Solution Removal from Large Solution Sets

Tianye Shu, Yang Nan, Ke Shang*, Hisao Ishibuchi*
Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China

***Abstract*—In evolutionary multi-objective optimization (EMO), one important issue is to efficiently remove dominated solutions from a large number of solutions examined by an EMO algorithm. An efficient approach to remove dominated solutions from a large solution set is to partition it into small subsets. Dominated solutions are removed from each subset independently. This partition method is fast but cannot guarantee to remove all dominated solutions. To further remove remaining dominated solutions, a simple idea is to iteratively perform this approach. In this paper, we first examine three partition methods (random, objective value-based and cosine similarity-based methods) and their iterative versions through computational experiments on artificial test problems (DTLZ and WFG) and real-world problems. Our results show that the choice of an appropriate partition method is problem dependent. This observation motivates us to use a hybrid approach where different partition methods are used in an iterative manner. The results show that all dominated solutions are removed by the hybrid approach in most cases. Then, we examine the effects of the following factors on the computation time and the removal performance: the number of objectives, the shape of the Pareto front, and the number of subsets in each partition method.**

I. Introduction

In evolutionary multi-objective optimization (EMO), two solutions are usually compared using the following Pareto dominance relation. For a minimization problem, a solution $\boldsymbol{a}$ is said to be dominated by another solution $\boldsymbol{b}$ (i.e., $\boldsymbol{a} \succ \boldsymbol{b}$) *iff* $\forall i \in \{1, ..., M\}$, $a_i \geq b_i$ and $\exists j \in \{1, ..., M\}$, $a_j \neq b_j$ where $\boldsymbol{a} = (a_1, a_2, ..., a_M)$, $\boldsymbol{b} = (b_1, b_2, ..., b_M)$, and M is the number of objectives. The target of EMO algorithms is to find a set of well distributed nondominated solutions which represents the best trade-off surface (i.e., Pareto front) among the conflicting objectives of a multi-objective optimization problem. To obtain these nondominated solutions, dominated solution removal is a necessary procedure for EMO algorithms.

There are two situations where we need to remove dominated solutions from a large solution set: performance evaluation of EMO algorithms and numerical approximation of unknown Pareto fronts. For EMO algorithms, performance evaluation is usually based on one of the following two scenarios [1], [2], [3]: the final population scenario and the reduced UEA (unbounded external archive) scenario. Although the final population is widely used to compare the performance of different EMO algorithms, it has some limitations: (1) some good solutions are discarded in the evolutionary process and not included in the final population [4], and (2) it is unfair to compare the performance of EMO algorithms with different population sizes [3]. To overcome these limitations, the reduced UEA scenario uses a pre-specified number of solutions selected from the nondominated solutions in the UEA for the performance assessment. Under the reduced UEA scenario, we often need to remove dominated solutions from a large number of examined solutions.

Dominated solution removal is also important for the approximation of an unknown Pareto front of a multi-objective problem. The Pareto fronts of most artificial test problems (e.g., DTLZ1-4 [5] and WFG4-9 [6]) are mathematically defined and have triangular shapes. In this paper, triangular shapes are consider as regular and other shapes (e.g., inverted triangular shapes and degenerated shapes) are considered as irregular [7]. However, for many real-world problems (e.g., DDMOP [8] and RE [9]) and some artificial problems (e.g., some partially degenerate test problems [10]), their true Pareto fronts are usually unknown. To obtain an approximated Pareto front, many researchers [8], [10], [9] use nondominated solutions among all solutions examined in multiple runs of various EMO algorithms. The approximated Pareto front can be used for problem analysis in real-world applications. It can be also used for estimating the nadir and ideal points and specifying the reference point set for the GD [11], IGD [12] and IGD+ [13] calculation for each multi-objective problem with an unknown Pareto front. To obtain a reliable approximated Pareto front, the number of examined solutions is usually very large (e.g., several millions). The identification of nondominated solutions (i.e., removal of dominated solutions) from such a large number of examined solutions is an important research topic.

The above discussions show the importance of designing an efficient algorithm to remove dominated solutions from a large solution set. A straightforward idea is to partition the large

This work was supported by National Natural Science Foundation of China (Grant No. 62002152, 62250710163, 62250710682), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).
*Corresponding authors: Ke Shang (kshang@foxmail.com) and Hisao Ishibuchi (hisao@sustech.edu.cn)

solution set into small subsets. From each subset, dominated solutions are eliminated independently. This method is fast but defective since it cannot guarantee to remove all dominated solutions. To further eliminate remaining dominated solutions, a simple idea is to iteratively use this method on the remaining solutions. In this paper, we examine the iterative versions of three partition methods: (1) random partition, (2) cosine similarity-based partition, and (3) objective value-based partition. In our computational experiments, we generate solution sets of size 500,000 for the DTLZ1-4 [5] and WFG1-9 [6] test problems with 3-10 objectives and the real-world RE [9] test problems with 3-9 objectives. Then, each partition method is applied to each solution set. Our experimental results show that the number of remaining dominated solutions is further decreased by iterating each partition method (i.e., by applying each partition method to the remaining solution set in an iterative manner). Interestingly, cosine similarity-based partition and objective value-based partition methods show their own superiority on the solution sets with regular Pareto fronts and irregular Pareto fronts, respectively. Thus, we further examine a hybrid approach where a different partition method is used in each iteration instead of iteratively applying the same partition method to the remaining solution set. Much better results are obtained by the hybrid approach than iterating a single partition method. We also examine the effects of the number of objectives, the shape of the Pareto front and the number of subsets on the performance of our partition methods with respect to their computation time and dominated solution removal performance.

The remainder is organized as follows. In Section II, we briefly review some representative methods for dominated solution removal. In Section III, we introduce three partition methods and their iterative versions for dominated solution removal. Then, we examine the performance of these methods through computational experiments in Sections IV-VI. In Section IV, experimental settings are explained including the creation of various solution sets of size 500,000. In Section V, experimental results are reported and analyzed, and the hybrid approach is proposed. In Section VI, the effects of various factors on the performance of our partition methods are examined. Finally, we conclude this paper in Section VII.

II. Background

Dominated solution removal is an important topic in many fields (e.g., database [14] and computational geometry [15], [16]). In evolutionary multi-objective optimization (EMO), dominated solution removal is usually used to obtain all nondominated solutions from an archive. Figure 1 shows a two-objective solution set where solutions $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ are nondominated solutions and others are dominated solutions. To remove all dominated solutions, a naive method compares every pair of solutions and removes the dominated one. The time complexity of this naive method is $O(MN^2)$ where M is the number of objectives and N is number of solutions. The naive method is time-consuming when the number of solutions (i.e., N) is large. Kung et al. [15] proposed a divide-and-conquer

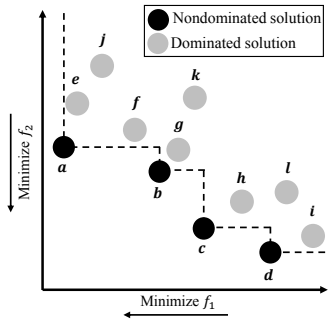


Fig. 1. A two-objective example for removing dominated solutions (i.e., grey points).

algorithm for the dominated solution removal, which has a time complexity of $O(N(\log N)^{\max\{M-2,1\}})$. An extension of Kung's method is proposed in [14] which recursively divides the solution set into m $(m > 2)$ subsets instead of two subsets. To avoid unnecessary comparisons in the naive method, another idea is to first check the solutions which are more likely to be nondominated [17], [18]. Based on this idea, many dominated solution removal methods [17], [19], [20] are designed which determine the solution check sequence based on the objective values in a different manner. These methods [17], [19] show better performance than Kung's method when the number of objectives is large. Many data structures (see [21]) are designed for efficiently removing dominated solutions. For example, the ND-Tree data structure [22] dynamically maintains a partition of a set of nondominated solutions and can be updated efficiently when a new solution is added to the solution set. In this paper, T-ENS [23] is used to remove dominated solutions which is an efficient and widely-used method (e.g., used in PlatEMO [24]) for dominated solution removal and nondominated sorting. T-ENS uses an $(M-1)$-ary tree to store the solutions in each nondominated front and has a time complexity of $O(MN \log N / \log M)$ when most solutions in the solution set are nondominated [23].

The idea of partition is frequently used to speed up the process of the dominated solution removal. For example, in Kung's method [15], the solution set is partitioned based on the median value of a pre-specified objective while α-quantiles are used in its extended method [14]. In ND-Tree method [22], the solutions are partitioned based on the distance in the objective space.

III. Partition Methods for Dominated Solution Removal

When the number of solutions is large, dominated solution removal becomes time-consuming. To handle a large solution set, a simple idea is to partition the large solution set into a number of small subsets. Dominated solutions are removed from each subset independently. Although this method is fast but cannot always remove all dominated solutions. In this section, we try to improve this method to remove almost all dominated solutions.

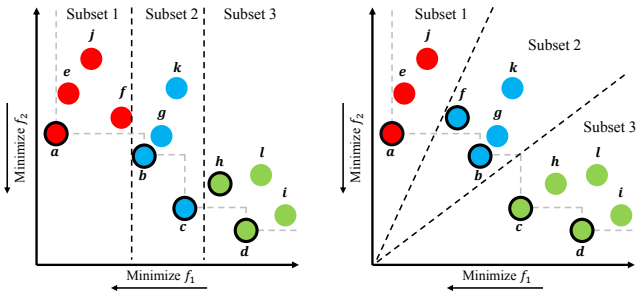
A. Partition Methods with Some Heuristics

The ideal case for a partition method is as follows: Each dominated solution is included in a subset together with at least one of its corresponding dominated solutions. In this case, all dominated solutions are removed by the partition method. The simplest partition method is the **random partition** which randomly partitions the solutions into a pre-specified number of subsets evenly. However, it is not very likely that two solutions with the dominance relation (i.e., one solution dominates the other solution) are included in the same subset. Thus, many dominated solutions cannot be removed by the random partition method. So, it is needed to use a special partition mechanism (instead of the random partition) so that two solutions with the dominance relation are included in the same subset. From Figure 1, we can see that each dominated solution and the corresponding dominated solution(s) have similar directions. For example, vector $\boldsymbol{j}$ (i.e., dominated solution $\boldsymbol{j}$) and vector $\boldsymbol{a}$ (i.e., dominating solution $\boldsymbol{a}$) have similar directions. That is, the angle between $\boldsymbol{j}$ and $\boldsymbol{a}$ is very small. One idea for partition is to use the angle between two solutions, which is measured by the cosine similarity S_c. Here the cosine similarity $S_c(\boldsymbol{s_1}, \boldsymbol{s_2})$ between two solutions $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ is defined as the cosine of the angle between these two solution vectors in the objective space (i.e., $S_c(\boldsymbol{s_1}, \boldsymbol{s_2}) = \frac{\boldsymbol{s_1} \cdot \boldsymbol{s_2}}{\|\boldsymbol{s_1}\| \|\boldsymbol{s_2}\|}$). From Figure 1, we can also see that solutions $\boldsymbol{j}$ and $\boldsymbol{a}$ have similar objective values for the first objective f_1. Thus another idea is to partition the solution set using objective values of a specific objective. Based on these observations, the following two partition methods can be used to remove most dominated solutions.

Objective Value-based Partition: One of the M objectives is selected, and all the solutions are sorted based on their objective values of the selected objective. Then, the sorted solutions are partitioned into a pre-specified number of subsets with the same size.

Cosine Similarity-based Partition: Objective values of all solutions are normalized for each objective so that the minimum and maximum values for each objective are 0 and 1, respectively. After this normalization, all solutions are in the M-dimensional unit hypercube $[0, 1]^M$. Then, each solution is mapped to a point on the hypersphere defined by $f_1^2 + f_2^2 + ... + f_M^2 = 1$ using the following formulation: $f_i' = f_i / (f_1^2 + f_2^2 + ... + f_M^2)^{\frac{1}{2}}$ for $1 \leq i \leq M$. The Euclidean distance $Dist(\boldsymbol{s_1'}, \boldsymbol{s_2'})$ between two mapped solutions $\boldsymbol{s_1'}$ and $\boldsymbol{s_2'}$ equals to $\sqrt{2 - 2S_c(\boldsymbol{s_1}, \boldsymbol{s_1})}$. A clustering algorithm is used to partition the mapped solutions into a pre-specified number of subsets (i.e., clusters) based on the Euclidean distance. In this manner, two solutions with higher cosine similarity S_c are more likely to be included in the same subset (i.e., cluster). In our computational experiments, we use an efficient clustering algorithm called Mini-batch k-Means [25]. For Mini-batch k-Means, we set the number of mini-batch size as 1000 and the number of iterations as 100.

The above two partition methods are illustrated in Figure 2 where 12 solutions are divided into three subsets with different colors. As an example, the first objective is used for the objective value-based partition method. As we can see from Figure 2 (b), the subset size is not always the same in the cosine similarity-based partition method since the clustering results are used for partition. In Figure 2 (a), each subset has four solutions, and solution $\boldsymbol{h}$ cannot be removed (since $\boldsymbol{h}$ is not dominated by any other solutions in the same subset). In Figure 2 (b), solution $\boldsymbol{f}$ cannot be removed.



(a) Objective value-based partition (f_1 is used) (b) Cosine similarity-based partition

Fig. 2. Illustration of the two partition methods to partition the solution set in Figure 1 into three subsets. Dominated solution $\boldsymbol{h}$ cannot be removed in (a), and dominated solution $\boldsymbol{f}$ cannot be removed in (b).

B. Iterative Version

To further decrease the number of remaining dominated solutions, one simple idea is to iteratively apply the partition method to the remaining solution set. Figure 3 shows the iterative version of the partition method for dominated solution removal. In each iteration, the current solution set S is divided into K subsets, and dominated solutions are removed from each subset independently. The remaining solutions are merged into a new solution set T which is used as the current solution set S in the next iteration. When a termination condition (e.g., the maximum number of iterations) is satisfied, the remaining solutions are output as the result.

When the random partition method is iteratively used, different subsets are created in each iteration even when no dominated solution is removed in the previous iteration. When the objective value-based partition method is used, a different objective is selected in each iteration in a sequential order. More specifically, the first objective is used in the first iteration, and the M-th objective is used in the M-th iteration. Then the first objective is used again in the $(M + 1)$-th iteration. In this case, if no dominated solution is removed in the previous M iterations, any other dominated solution cannot be removed in the future iterations. This is because the same series of M partitions iteratively appear with the cycle M. When the cosine similarity-based partition method is used, different subsets are created in each iteration due to the randomness of the clustering algorithm even when no dominated solution is removed in the previous iteration. However, it is likely that similar clustering results are obtained when the number of removed dominated solutions is small in the previous iteration.

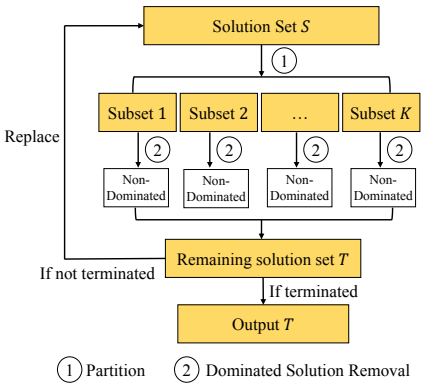


Fig. 3. Illustration of the iterative version of the partition method for dominated solution removal.

Suppose that the time complexity of a dominated solution removal method is $O(N^2)$ where N is the number of solutions. When the solution set of size N is divided into K subsets of size N/K, the time complexity for each subset is $O(N^2/K^2)$. Thus, the time complexity for the K subsets is $O(N^2/K)$. When the partition method is iterated t times, the time complexity for dominated solution removal is $O(tN^2/K)$. Thus, the total computation time of the partition method with t iterations will be about t/K of the straightforward case where the dominated solution removal method is directly applied to all solutions. Actually, the computation time in each iteration gradually decreases since the solution set size gradually decreases (i.e., since some dominated solutions are removed in each iteration). Thus, the computation time of the iterated partition method can be much shorter than t/K of the straightforward removal as shown in the computational experiments in this paper.

IV. Experimental Settings

A. Solution Set Generation

As explained in Section I, the dominated solution removal procedure is usually needed for the performance evaluation of EMO algorithms and the approximation of unknown Pareto fronts. Corresponding to these two situations, we generate two types of solution sets. For the first type of solution sets, we gather all solutions examined by an EMO algorithm in a single run for a test problem. Specifically, we apply a widely-used many-objective algorithm NSGA-III [26] to scalable test problems DTLZ1-4 [5] and WFG1-9 [6]. Among these test problems, DTLZ1-4 and WFG4-9 have regular (i.e., triangular) Pareto fronts while WFG1-3 have irregular Pareto fronts. The population size is set as 120, 126 and 275 for 3-objective, 5-objective and 10-objective problems, respectively. The maximum number of solution evaluations is set as 500,000. For each test problem, all solutions examined by NSGA-III are stored as a solution set. Thus, we have 39 solution sets with 500,000 solutions. The second type of solution sets contains all examined solutions of a real-world problem with an unknown Pareto front in multiple runs of various EMO algorithms. We use a real-world problem test suit RE [9] (with 16 problems) whose true Pareto fronts are unknown. Among them, two-objective problems (4 out of 16 problems) are not used since dominated solutions can be easily removed in the case of two objectives. The remaining 11 problems are used in our experiments. Four EMO algorithms MOEA/D-PBI [27], SMS-EMOA [28], [29], NSGA-II [30] and NSGA-III [26] are applied to each RE problem five times. The maximum number of solution evaluations is set as 25,000 in each run. We gather all examined solutions for each problem to generate a solution set of size 500,000 (25,000 solution evaluations $\times$ 4 algorithms $\times$ 5 runs). In this manner, we obtain 11 solution sets of size 500,000. It should be noted that some duplicated solutions are included in some solution sets (e.g., RE3-4-6). This is because the same solution in the objective space is generated and evaluated multiple times especially when discrete decision variables are included in the problem.

In total, 50 solution sets of size 500,000 are generated (39 from artificial test problems and 11 from real-world problems). Each solution set is used to evaluate the performance of the proposed partition methods for dominated solution removal in our computational experiments.

B. Compared Methods

We consider the iterative versions of the three partition methods in Section III: (a) random partition, (b) objective value-based partition, and (c) cosine similarity-based partition. For simplicity, they are denoted as IR, IO and IC, respectively. For all of these methods, the number of iterations is set as 20 and the number of subsets is set as 50.

As we have already explained, T-ENS [23] is used as the baseline algorithm which is applied to the entire solution set for comparison. T-ENS is also used in our three partition methods. The random partition and cosine similarity-based partition methods are performed ten times with different random seeds. The other methods (i.e., the baseline T-ENS algorithm and the objective value-based partition method) are performed once since they have no randomness. To evaluate the performance of each method, we calculate the number of remaining dominated solutions and record the computation time. The experiments are conducted on a virtual machine equipped with two ADM EPYC 7702 64-Core CUP@2.4GHz, 256GB RAM and Ubuntu Operating System. All codes are implemented in MATLAB R2021b and are available at https://github.com/HisaoLabSUSTC/Iterative-Partition-Method-for-Dominated-Solution-Removal.

V. Experimental Results

As explained in the previous section, the average computation time and the average number of remaining dominated solutions obtained by each removal methods are calculated over 10 runs (one run for objective value-based partition method and the baseline T-ENS algorithm since they have no randomness) for 50 solution sets (12 for DTLZ problems, 27 for WFG problems and 11 for RE problems). Experimental results on all 50 solution sets are included in the supplementary

file[1]. Among 50 solution sets, we choose eight representative ones and their results are shown in Figure 4. In Figure 4, "Exact" means the results of the baseline T-ENS algorithm where T-ENS is applied to the entire solution set. Thus, all dominated solutions are always removed. The vertical axis is the computation time, and the horizontal axis is the number of remaining dominated solutions. The smaller values show better results for both axes. It should be noted that both axes are logarithmic.

A. Performance Comparison

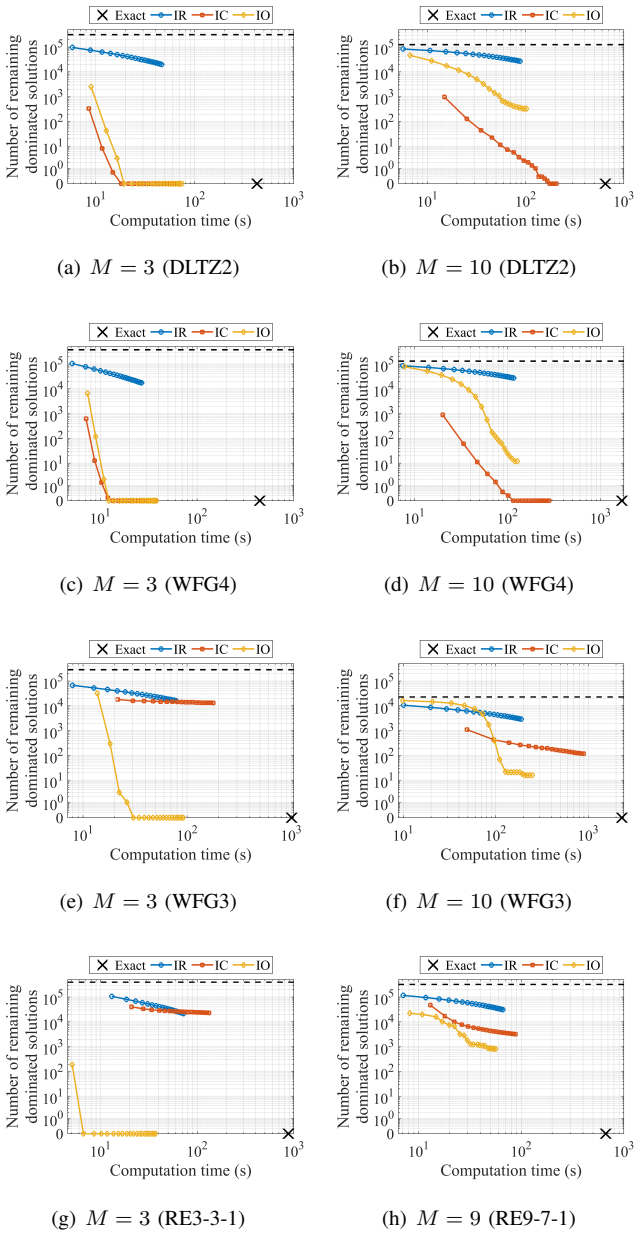
In Figure 4, the dashed horizontal line close to the top of each figure shows the total number of dominated solutions included in each solution set. Generally, when the number of objectives increases, the number of nondominated solutions increases and the number of dominated solutions decreases. Unsurprisingly, the exact method (i.e., T-ENS), which removes all dominated solutions, needs the largest computation time. In general, when the number of iterations is one (i.e., the leftmost point of each curve), each partition method removes part of dominated solutions in a very short computation time. Among the three partition methods, the random partition method is the fastest one while the cosine similarity-based partition method is the most time-consuming one. This is because the cosine similarity-based partition uses a clustering algorithm which takes a considerable computation time. As the number of iterations increases, the number of remaining dominated solutions decreases in each partition method.

As expected, the random partition method has the worst performance among the three partition methods in Figure 4. Many dominated solutions are not removed by the random partition method even after 20 iterations. This is because two solutions with the dominance relation are often included in different subsets. In Figure 4 whereas the cosine similarity-based partition method shows the best performance for the DTLZ2 and WFG4 problems, it does not work well for the WFG3 and two RE problems. This observation implies that the performance of the cosine similarity-based partition method depends on the distribution of solutions (i.e., the Pareto front shape). In contrast, the objective value-based partition method shows the best performance for the WFG3 and two RE problems. However, it is slightly worse than the cosine similarity-based partition method for the DTLZ2 and WFG4 problems. We will further analyze these results in detail in Section VI.

B. Further Improvement by Hybrid Partition

In our computational experiments, the cosine similarity-based partition and the objective value-based partition methods have demonstrated the best performance on different solution sets. A straightforward idea is to hybridize these two partition methods into a single method. We examine two versions of hybrid partition methods. The first version (denoted as IC+IO) uses cosine similarity-based partition in the first half of iterations and objective value-based partition in the second half of iterations. The second version (denoted as IO+IC) uses objective value-based partition in the first half of iterations and cosine similarity-based partition in the second half of iterations. Experimental results are shown in Figure 5. In Figure 5, more dominated solutions are removed by the hybrid partition methods (i.e., IC+IO and IO+IC) than the original methods (i.e., IC and IO) on the 10-objective DTLZ2 and WFG3 problems.

For each partition method, we divide the obtained solution sets into three categories based on the number of remaining dominated solutions: (a) All dominated solutions are removed. (b) The number of remaining dominated solutions is less than ten. (c) The number of remaining dominated solutions



(a) $M = 3$ (DLTZ2) (b) $M = 10$ (DLTZ2)

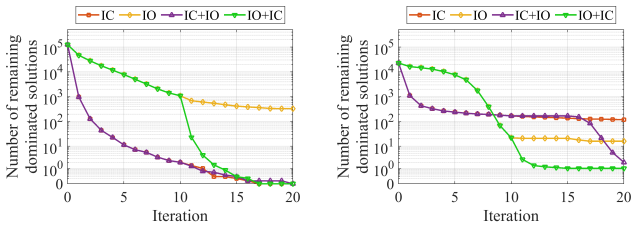
(c) $M = 3$ (WFG4) (d) $M = 10$ (WFG4)

(e) $M = 3$ (WFG3) (f) $M = 10$ (WFG3)

(g) $M = 3$ (RE3-3-1) (h) $M = 9$ (RE9-7-1)

Fig. 4. Performance comparison of different methods for removing dominated solutions from different solution sets. Average values over ten runs are shown.

[1]https://github.com/HisaoLabSUSTC/Iterative-Partition-Method-for-Dominated-Solution-Removal/blob/main/Supplementary%20file.pdf



(a) $M = 10$ (DLTZ2) (b) $M = 10$ (WFG3)

Fig. 5. Comparison between the hybrid partition methods and the original partition methods for the 10-objective DTLZ2 and WFG3 problems.

TABLE I
THE NUMBERS OF SOLUTION SETS WITH DIFFERENT DOMINATED SOLUTION REMOVAL PERFORMANCE. IN EACH CELL, (n_1, n_2, n_3) MEANS THE FOLLOWING. n_1: THE NUMBER OF SOLUTION SETS FOR WHICH ALL DOMINATED SOLUTIONS ARE REMOVED (I.E., WITH 0 REMAINING DOMINATED SOLUTION), n_2: THE NUMBER OF SOLUTION SETS WITH 1-9 REMAINING DOMINATED SOLUTIONS, AND n_3: THE NUMBER OF SOLUTION SETS WITH MORE THAN 9 REMAINING DOMINATED SOLUTIONS. LARGER VALUES OF n_1 AND SMALLER VALUES OF n_3 IN (n_1, n_2, n_3) MEAN BETTER METHODS.

Method	WFG	DTLZ	RE	Total
IR	(0, 0, 27)	(0, 0, 12)	(1, 0, 10)	(1, 0, 49)
IC	(14, 3, 10)	(7, 3, 2)	(0, 0, 11)	(21, 6, 23)
IO	(11, 5, 11)	(4, 3, 5)	(7, 2, 2)	(22, 10, 18)
IC+IO	(20, 6, 1)	(10, 2, 0)	(9, 1, 1)	(39, 9, 2)
IO+IC	(21, 4, 2)	(10, 2, 0)	(10, 0, 1)	(41, 6, 3)

is more than or equal to ten. The results are summarized in Table I. When the hybrid methods are used in Table 2, only two or three solution sets (among the examined 50 solution sets of size 500,000) have ten or more remaining dominated solutions. These observations show the effectiveness of the hybrid partition methods for dominated solution removal from large solution sets.

VI. RESULTS ANALYSIS

A. Effects of the Number of Objectives

One clear observation from Figure 4 is that the dominated solution removal becomes difficult for the partition method in high-dimensional spaces. For example, to remove all dominated solutions, the cosine similarity-based partition method needs four iterations for the 3-objective DTLZ2 problem, but more than 10 iterations for the 10-objective DTLZ2 problem as shown in Figure 4 (a) and (b). This is because each dominated solution has less dominating solutions in higher-dimensional spaces. For each dominated solution in each solution set, we examine the number of its dominating solutions. Then, we categorize each dominated solution into one of four types depending on the number of its dominating solutions. Type 1: dominated solutions with only one dominating solution, Type 2: dominated solutions with 2-10 dominating solutions, Type 3: dominated solutions with 11-100 dominating solutions, and Type 4: dominated solutions with more than 100 dominating solutions. Since Type 4 dominated solutions have many dominating solutions, it is very likely that they are easily

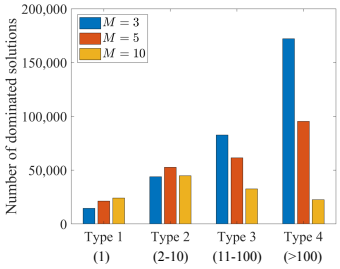


Fig. 6. Number of dominated solutions in each of the four types for the 3-objective, 5-objective and 10-objective DTLZ2 problems. The four types of dominated solutions have 1, 2-10, 11-100 and more than 100 dominating solutions, respectively.

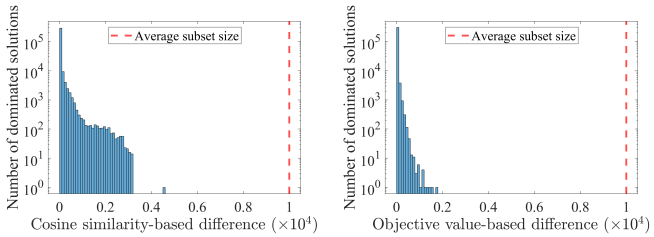
removed by any partition method. On the contrary, Type 1 dominated solutions have only one dominating solutions. Thus, their removal by partition methods is very difficult. We need a careful partition method to include each Type 1 solution and its dominating solutions in the same subset. Figure 6 shows the number of each type of dominated solutions in the 3-objective, 5-objective and 10-objective DTLZ2 problems. As we can see, most dominated solutions have more than 100 corresponding dominating solutions in the 3-objective DTLZ2 problem. However, the number of corresponding dominating solutions becomes less than 11 for most dominated solutions in the 10-objective DTLZ2 problem. That is, it becomes difficult to include each dominated solution and at least one corresponding dominating solution in the same subset in high-dimensional objective spaces. Therefore, the performance of all partition methods deteriorates as the number of objectives increases in Figure 4 with respect to both the computation time and the number of remaining dominated solutions.

B. Effects of the Solution Distribution

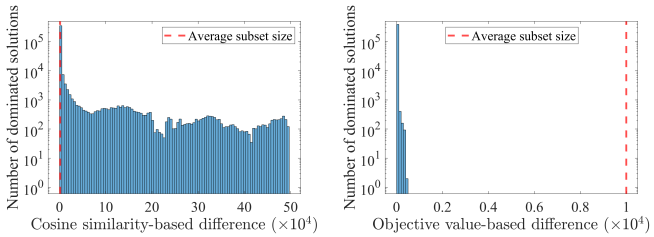
As shown in Figure 4, the cosine similarity-based partition method shows excellent results for the solution sets with regular Pareto fronts (e.g., the DTLZ2 and WFG4 problems). However, for the solution sets with irregular Pareto fronts (e.g., the WFG3 and two RE problems), the cosine similarity-based partition method does not work. For those solution sets, the objective value-based partition method shows good performance. To further analyze these results, we quantify the cosine similarity-based difference and the objective value-based difference between the dominated solutions and their corresponding dominating solutions in the following manner. For a dominated solution $\boldsymbol{s}$ in the solution set S, we rank the other solutions in S based on their cosine similarity with $\boldsymbol{s}$. The most similar solution to $\boldsymbol{s}$ has Rank 1 and the least similar solution has Rank $(N-1)$ where N is the solution set size ($N = 500,000$ in our computational experiments). The cosine similarity-based difference $Diff_c(\boldsymbol{s}, S)$ is defined as the minimum rank over all dominating solutions of $\boldsymbol{s}$. As we can see, larger $Diff_c(\boldsymbol{s}, S)$ means that it is more difficult for the cosine similarity-based partition method to include the dominated solution $\boldsymbol{s}$ and its corresponding dominating solutions(s) in the same subset. In a similar manner, we define

the objective value-based difference $Diff_o(\boldsymbol{s}, S)$ using the objective value-based ranking of the solutions in S for each dominated solution $\boldsymbol{s}$. First, all solutions in S are sorted using the i-th objective value. The solution with the smallest objective value has Rank 1, and the solution with the largest objective value has Rank N. Let us denote the rank of solution $\boldsymbol{s}$ based on the i-th objective by $\sigma_i(\boldsymbol{s})$ for $i = 1, 2, ..., M$. The objective value-based difference between a dominated solution $\boldsymbol{s}$ and its dominating solution $\boldsymbol{s'}$ is defined by the difference between their ranks as $|\sigma_i(\boldsymbol{s}) - \sigma_i(\boldsymbol{s'})|$. When this objective value-based difference is small, it is likely that $\boldsymbol{s}$ and $\boldsymbol{s'}$ are included in the same subset by the objective value-based partition method for the i-th objective. Since all objectives are used in the iterated version, we define the objective value-based difference $Diff_o(\boldsymbol{s}, S)$ between $\boldsymbol{s}$ and its dominating solution(s) in S as the minimum objective value-based difference over all dominating solutions for all objectives. That is, $Diff_o(\boldsymbol{s}, S) = min\{|\sigma_i(\boldsymbol{s}) - \sigma_i(\boldsymbol{s'})||1 \leq i \leq M, \boldsymbol{s'} \in S'\}$ where M is the number of objectives and S' is the set of solutions in S which dominate $\boldsymbol{s}$.

Figure 7 shows the cosine similarity-based difference and objective value-based difference distributions for all dominated solutions for the 3-objective DTLZ2 and RE3-3-1 problems. As a reference, the average size of subsets (i.e., 10,000) is shown by a red dotted line. Roughly speaking, a dominated solution $\boldsymbol{s}$ is likely to be group together with at least one of its corresponding dominating solution(s) by the objective value-based (cosine similarity-based) partition if its objective value-based (cosine similarity-based) difference is smaller than the average size of subsets. The dominated solutions and their corresponding dominating solutions have both small cosine similarity-based difference and small objective value-based difference in the 3-objective DTLZ2 problem as shown in Figure 7 (a). For the RE3-3-1 problem, each dominated solution has large cosine similarity-based difference but small objective similarity-based difference as shown in Figure 7 (b). Further discussions are included in page 21 of the supplementary file.

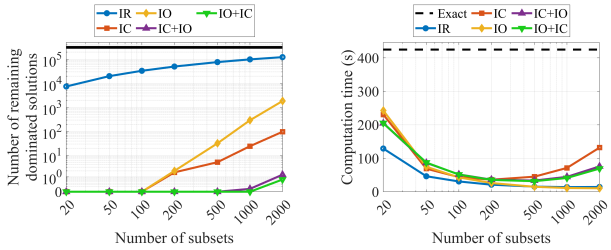


(a) $M = 3$ (DTLZ2)



(b) $M = 3$ (RE3-3-1)

Fig. 7. Distributions of the cosine similarity-based difference and the objective value-based difference for all dominated solutions for the 3-objective DTLZ2 problem in (a) and the 3-objective RE3-3-1 problem in (b). A dominated solution is likely to be removed by the cosine similarity-based (objective value-based) partition method if its cosine similarity-based (objective value-based) difference is smaller than the average size of subsets.



(a) (b)

Fig. 8. (a) Number of remaining dominated solutions and (b) computation time of each partition method with different specifications of the number of subsets for the 3-objective DTLZ2. The bold horizontal line in (a) shows the number of dominated solutions included in each solution set with 500,000 solutions. The dashed horizontal line in (b) shows the computation time of the exact method (where the number of subsets is one).

C. Effects of the Number of Subsets

Figure 8 (a) shows the number of remaining dominated solutions by each partition method with various specifications of the number of subsets. As we can see, when the solution set is partitioned into more subsets (i.e., smaller sizes), the number of remaining dominated solutions increases in each partition method. This is because each dominated solution is less likely to be grouped together with its dominating solution(s) when the subset size decreases.

Figure 8 (b) shows the computation time of each partition method with various specifications of the number of subsets. For the random and objective value-based partition methods, the computation time decreases as the number of subsets increases. This is consistent with the analysis in Section III-B. However, the computation time of the cosine similarity-based partition method first decreases and then increases with the increase in the number of subsets (i.e., with the decrease in the subset size). The reason for the increase of the computation time by too many subsets is as follows: To partition the solution set, the cosine similarity-based partition method uses a clustering algorithm whose computation time is linear to the number of subsets (i.e., clusters). When the number of subsets is large, the clustering algorithm takes a major part of the computation time in the cosine similarity-based partition method.

VII. Conclusion

In this paper, we examined three partition methods, their iterative versions, and their hybrid versions for dominated solution removal from various large solution sets of size 500,000, which are created from artificial test problems (DTLZ and WFG) and real-world problems in the RE problem suite. We obtained the following observations from our experimental results.

1) Dominated solution removal becomes more difficult for the partition method as the number of objectives increases.
2) Although a single iteration of any partition method cannot remove all dominated solutions, the number of remaining dominated solutions is further decreased by iteratively applying the partition method. When the number of iterations is not large (e.g., 20 iterations), the partition method is still much faster than the baseline exact method with no partition.
3) The cosine similarity-based partition method shows the best performance on the solution sets of multi-objective problems with regular Pareto fronts while the objective value-based partition method shows the best performance on the solution sets of multi-objective problems with irregular Pareto fronts.

Inspired by the third observation, we further examined two hybrid versions where both the cosine similarity-based and objective value-based partition methods are used (i.e., each method is used in a different iteration). Our experimental results showed that all dominated solutions are removed by the two hybrid versions from most solution sets in much shorter computation time than the baseline exact algorithm with no partition. To analyze the first and the third observations, we calculated some statistics for all dominated solutions in the solution set. Then, we explained the reason for these observations using the relations between each dominated solution and its corresponding dominating solutions.

As future work, it is interesting to further examine the remaining dominated solutions in some solution sets for the design of a new efficient mechanism to remove them. In this work, the solution sets are generated by applying EMO algorithms to multi-objective problems. It is also interesting to test the partition method on large solution sets in other research fields. Another research direction is the parallel implementation of the partition methods, which may be needed when we handle much larger solution sets.

REFERENCES

[1] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking multi- and many-objective evolutionary algorithms under two optimization scenarios," *IEEE Access*, vol. 5, pp. 19 597–19 619, 2017.

[2] H. Ishibuchi, L. M. Pang, and K. Shang, "A new framework of evolutionary multi-objective algorithms with an unbounded external archive," in *Proc. Eur. Conf. Artif. Intell.*, Santiago de Compostela, Spain, Aug./Sep. 2020, pp. 283–290.

[3] 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. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, July 2016, pp. 1149–1156.

[4] M. Li and X. Yao, "An empirical investigation of the optimality and monotonicity properties of multiobjective archiving methods," in *Proc. Evol. Mult. Criter. Optim.*, East Lansing, MI, USA, Mar. 2019, pp. 15–26.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 105–145.

[6] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[7] H. Ishibuchi, L. He, and K. Shang, "Regular Pareto front shape is not realistic," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 2034–2041.

[8] C. He, Y. Tian, H. Wang, and Y. Jin, "A repository of real-world datasets for data-driven evolutionary multiobjective optimization," *Complex Intell. Syst.*, vol. 6, no. 1, pp. 189–197, 2020.

[9] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Appl. Soft Comput.*, vol. 89, p. 106078, 2020.

[10] L. M. Pang, Y. Nan, and H. Ishibuchi, "Partially degenerate multi-objective test problems," in *Proc. Evol. Mult. Criter. Optim.* Cham: Springer Nature Switzerland, 2023, pp. 277–290.

[11] D. A. Van Veldhuizen, *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. Ph.D. Dissertation, Air Force Institute of Technology, 1999.

[12] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Mexican International Conference on Artificial Intelligence*. Springer, 2004, pp. 688–697.

[13] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Proc. Evol. Mult. Criter. Optim.* Springer, 2015, pp. 110–125.

[14] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proceedings 17th International Conference on Data Engineering*, 2001, pp. 421–430.

[15] H.-T. Kung, F. Luccio, and F. P. Preparata, "On finding the maxima of a set of vectors," *Journal of the ACM (JACM)*, vol. 22, no. 4, pp. 469–476, 1975.

[16] F. P. Preparata and M. I. Shamos, *Computational geometry: An introduction*. Springer Science & Business Media, 2012.

[17] L. Ding, S. Zeng, and L. Kang, "A fast algorithm on finding the non-dominated set in multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 4. IEEE, 2003, pp. 2565–2571.

[18] Y. Nan, K. Shang, H. Ishibuchi, and L. He, "Reverse strategy for non-dominated archiving," *IEEE Access*, vol. 8, pp. 119 458–119 469, 2020.

[19] J. Du, Z. Cai, and Y. Chen, "A sorting based algorithm for finding a non-dominated set in multi-objective optimization," in *Proceedings of the Third International Conference on Natural Computation*, vol. 4. IEEE, 2007, pp. 436–440.

[20] K. K. Mishra and S. Harit, "A fast algorithm for finding the non dominated set in multi objective optimization," *Int. J. Comput. Appl.*, vol. 1, no. 25, pp. 35–39, 2010.

[21] J. E. Fieldsend, "Data structures for non-dominated sets: Implementations and empirical assessment of two decades of advances," in *Proc. Conf. Genet. Evol. Comput.*, New York, NY, USA, 2020, pp. 489–497.

[22] A. Jaszkiewicz and T. Lust, "ND-Tree-based update: A fast algorithm for the dynamic nondominance problem," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 778–791, 2018.

[23] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, 2018.

[24] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, 2017.

[25] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 1177–1178.

[26] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[27] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[28] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[29] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Proc. Evol. Mult. Criter. Optim.* Springer, 2005, pp. 62–76.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.