



# Using Dominated Solutions at Edges to the Diversity and the Uniformity of Non-dominated Solution Distributions in NSGA-II

Yuji Sato<sup>1</sup> · Mikiko Sato<sup>2</sup>

Received: 22 October 2021 / Accepted: 4 July 2022 / Published online: 8 August 2022  
© The Author(s) 2022

## Abstract

This paper proposes a method for improving the diversity of the Pareto front and the uniformity of non-dominated solution distributions in a fast elitist non-dominated sorting genetic algorithm (NSGA-II), which is an evolutionary multi-objective optimization algorithm. Conventional NSGA-II has excellent convergence to the Pareto front, but it has been reported that for some test cases, it does not produce a more diverse solution distribution than the strength Pareto evolutionary algorithm 2 (SPEA2). In addition, selection using the crowding distance may cause a bias in the selected solution distribution. To avoid this problem, we propose a method that archives dominated solutions that may be effective in improving diversity in the conventional search process when used for genetic operations, and mates these archived solutions with non-dominated solutions at the edge of rank 1 for each objective function. We experimentally compare this approach with the conventional method on the typical ZDT suite of multi-objective test problems and a two-objective constrained knapsack problem. By evaluating the performance based on Pareto front diagrams, the number of non-dominated solutions, the maximum spread and hypervolume values, we show that the proposed method is effective at improving the diversity at both ends of the Pareto optimal front and the solution distribution.

**Keywords** Multi-objective optimization · NSGA-II · Non-dominated sorting · Pareto optimal front · Dominated solution · Crowding sorting

## Introduction

Many real-world optimization problems have multiple objectives. These objectives often have trade-off relationships, and there is no single solution that is optimal for all objective functions. It is therefore important to have some way of accurately locating the curved surface (Pareto optimal front

[1]) formed by the set of Pareto optimal solutions. Evolutionary multi-objective optimization algorithms, which are based on evolutionary computation, are being researched as a way of tackling this problem [2], due to their ability to find a set of solutions that approximate the Pareto optimal front by running a single algorithm, and due to the breadth of optimal solutions they are able to find. In this paper, we focus on a fast elitist non-dominated sorting genetic algorithm (NSGA-II) [3, 4], which is the most practical of these algorithms. The main characteristics of the NSGA-II algorithm are its fast non-dominated sort, which improves convergence to the Pareto optimal front, and the crowding sort for the uniform solution distribution.

On the other hand, compared with the strength Pareto evolutionary algorithm 2 (SPEA2) [5], which focuses on the dominance of solutions and the preservation of non-dominated solutions as in NSGA-II, it has been reported that although it achieves better convergence on the Pareto front, there are test cases where it did not achieve superiority in terms of the diversity of the solution distribution. Also,

---

This article is part of the topical collection “Computational Intelligence” guest edited by Kurosh Madani, Kevin Warwick, Juan Julian Merelo, Thomas Bäck and Anna Kononova.

---

✉ Yuji Sato  
yuji@k.hosei.ac.jp  
Mikiko Sato  
mikiko.sato@tokai.ac.jp

<sup>1</sup> Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan

<sup>2</sup> School of Information and Telecommunication Engineering, Tokai University, Tokyo, Japan

while the selection of solutions based on crowding distance often works well to maintain population diversity, the resulting solution distribution can be biased. When working on real problems, the execution may be halted before completion if there is insufficient time available, in which case the solution used for product design can be selected using the Pareto front at that time. Even in a knapsack problem, if there are many objectives and a large population size, it may be difficult to accurately obtain the final Pareto front within a practical execution time. Therefore, if the types of non-dominated solutions (range of solution distribution) that can be selected can be expanded and the uniformity of the solution distribution can be improved at the solution search stage, this algorithm will be more effective when considering practical applications. In this paper, we address this problem by proposing a method that improves the uniformity of the solution distribution by using an archive population to preserve some of the dominated solutions that are usually culled at the start of a new generation, but which may be effective at improving the diversity of the population, and by actively using these dominated solutions in genetic operations.

We have already proposed adding a Rank 2 edge solution to the parent population for mating [6]. However, in cases where the number of Rank 1 solutions is less than the population size, then at least one of the solutions at the edge of Rank 2 should already be in the parent population, so the benefits of the proposed method can be expected only where the number of Rank 1 solutions is larger than the population size. In addition, the efficiency of the solution search may deteriorate because the lowest individual of Rank 1 is deleted instead of the solution candidate of the edge of Rank 2. In this paper, we revise the algorithm to solve these problems and update on our progress with discussions of new benchmarking problems, evaluations and discussions of the use of maximum spread (MS), and traditional solution selection problems.

## Conventional Methods

### Overview of NSGA-II

As shown in Eq. (1), a constrained multi-objective optimization problem involves minimizing (or maximizing)  $k$  different objective functions  $f$  based on  $m$  different inequality constraints  $g$ .

$$\begin{cases} f_i(x_1, x_2, \dots, x_n) \quad (i = 1, 2, \dots, k) \\ g_j(x_1, x_2, \dots, x_n) \leq 0 \quad (j = 1, 2, \dots, m) \end{cases} \quad (1)$$

Since there are trade-off relationships between the objective functions, studies are being made to find the Pareto optimal front by means of evolutionary computation. A typical

evolutionary multi-objective optimization algorithm is NSGA-II, which was proposed by Deb et al. in 2000 as an improved version of the non-dominated sorting genetic algorithm (NSGA) [7]. It searches for solutions using a combination of fast non-dominated sort, crowding sort, and crowded tournament selection. NSGA ranks individuals using non-dominant sorting and sharing these individuals according to their rank. While this method has the advantage of obtaining a wide variety of solutions within the same rank, it has problems such as the necessity of determining the sharing parameter and the high calculation cost. NSGA-II solved these problems by accelerating the calculation of ranking in non-dominated sort and using a new index called crowding distance as an alternative to sharing parameters.

Figure 1 shows a conceptual illustration of a fast dominated sort. A fast non-dominated sort is an operation that classifies all individuals by rank, focusing on the dominated/non-dominated relationships between individuals. For example, in a minimization problem, a candidate solution (individual)  $x$  is defined as dominating  $y$  when the following Eq. (2) is satisfied:

$$\forall f_i(x) \leq f_i(y) \wedge \exists f_i(x) < f_i(y) \quad (2)$$

Using this definition, we can rank each individual by ascertaining the dominated/non-dominated relationships between each individual. First, we determine the individuals that belong to the best Rank 1 group. For each individual, count the number of other individuals that it dominates, and the number of other individuals by which it is dominated. If it is not dominated by any other individual, then it is deemed to be a non-dominated solution and is placed in Rank 1. The other individuals are dominated solutions. Next, the Rank 1 individuals are ignored, and the dominated/non-dominated definitions are used to determine Rank 2 individuals under the same conditions as when determining the Rank 1

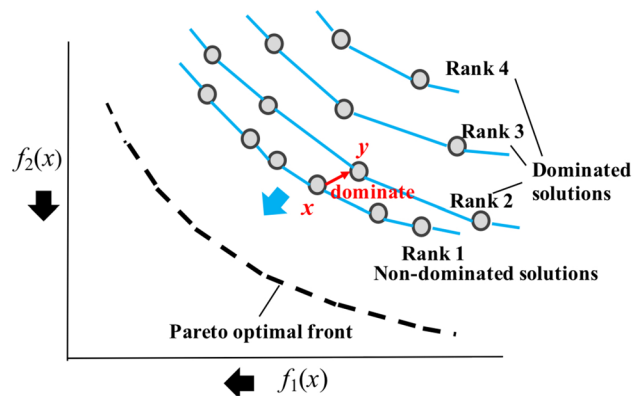


Fig. 1 Conceptual illustration of a fast dominated sort

individuals. A fast non-dominated sort is achieved by repeating this operation until all the individuals have been ranked.

Crowding sort is a method that determines dominated/non-dominated relationships between individuals at the same rank based on their crowding distance, which is the average distance between two solutions on either side of the  $i$ th solution along each of the objectives. A larger crowding distance (i.e., a solution that is less crowded) is ranked with higher precedence. Therefore, in many cases, it can be expected to work effectively to maintain the diversity of the solution population. In a crowded tournament selection, the solution candidates are first ranked by fast non-dominated sort, and then the candidates of equal rank are sorted by crowding distance.

Figure 2 shows a conceptual illustration of how the population is updated in a crowded tournament selection. NSGA-II advances the solution search using an archive population  $P_t$  that stores non-dominated solutions as the parent population, and the initial search population  $Q_t$  for performing the search as the child population. First, generate a group  $R_t = P_t \cup Q_t$  that combines the parent population  $P_t$  and the child population  $Q_t$ . This group  $R_t$  is first subjected to a non-dominated sort to rank each solution candidate. In the figure,  $F_n$  represents a solution candidate group of rank  $n$ . Next, perform a crowding sort and select the top  $N$  individuals with the highest number from the  $2N$  individuals  $R_t$  as  $P_{t+1}$ . The parent individuals selected from  $P_{t+1}$  are then crossed over and genetically manipulated by mutation to generate a new child population  $Q_{t+1}$ . These operations constitute one generation step, and these generation steps are repeated the specified number of times.

**Problems with Conventional Methods**

Figure 3 shows an example of the state diagram of a population  $R_t$  that is considered to lead to a decrease in the diversity of the solution distribution. The non-dominant sort used in

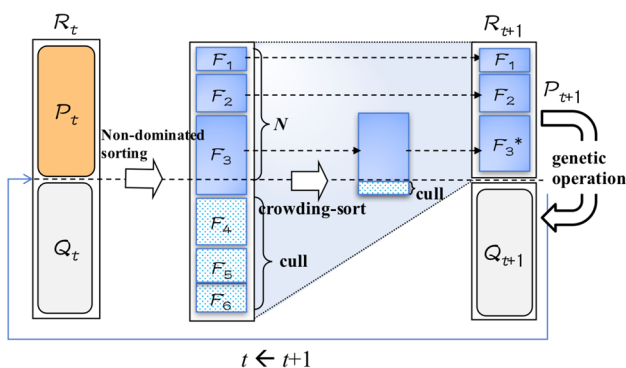


Fig. 2 Creating a new population generation by crowded tournament selection [1]

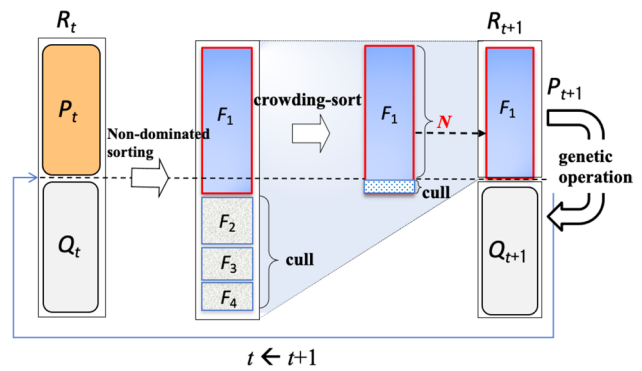


Fig. 3 Example of the state diagram of population  $R_t$  where all the archive groups are of Rank 1

NSGA-II is an operation that classifies all individuals into several ranks by focusing on the dominated/non-dominated relationships between individuals. Here, the best solution group is defined as Rank 1, followed in turn by Rank 2 and Rank 3. For some problems, these dominated/non-dominated solution relationships may cease to hold as the number of generations increases. In such cases, all  $N$  individuals selected for the archive population  $P_{t+1}$  would become Rank 1 solutions, while the other solutions of Rank 2, Rank 3, etc., obtained in the previous solution search would be completely eliminated. In this study, we consider that one of the reasons why NSGA-II sometimes has inferior solution diversity compared with SPEA2 is the lack of solutions of other ranks besides Rank 1 at the initial stages of the search.

Next, we discuss one of the features of NSGA-II, which is its ability to select solutions based on crowding distance [1, 3, 4]. NSGA-II seeks to maintain the diversity of solutions and the extent of distributions by preferentially selecting solutions with larger crowding distances (i.e., solutions that have no other solutions nearby). Figure 4 shows an example

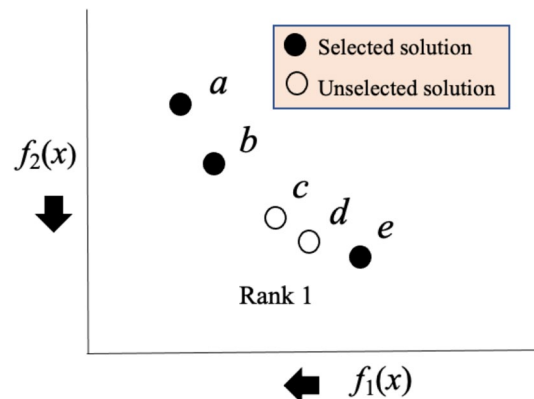


Fig. 4 An example that the crowding distance selection caused a bias in the selected solution distribution

of solution selection based on crowding distance when the number of Rank 1 individuals is larger than the population size  $N$ . In this figure, selected solutions are indicated by black circles and non-selected solutions are indicated by white circles. Since the crowding distance of the boundary solution is assigned to infinity, when selecting a solution based on the crowding distance, the boundary solutions  $a$  and  $e$  are selected first, and then  $b$  is selected because it has the largest crowding distance of the remaining solutions. The selection of solutions based on crowding distance works effectively in many cases. But in some cases, such as the distribution shown in Fig. 4, solutions located in dense areas may not be selected, causing these regions to be left blank (without any solutions). In other words, it can result in the formation of a Pareto front with a solution distribution that lacks uniformity.

### Proposal of Genetic Operation Using Dominated Solutions

#### Genetic Manipulation Using Dominated Solutions

In this paper, we propose a method applied to genetic operations that involves preserving some of the inferior solutions that are culled in conventional search processes but may be capable of leading to improved diversity. For example, the solutions at both ends of the Rank 2 solution distribution could be preserved. By preserving the solutions at both ends of the Rank 2 solution distribution in the archive population and using them for genetic manipulation, it may be possible to improve the diversity of the next generation of solutions. Figure 5 shows an example of a distribution diagram of current solutions generated by performing genetic operations only on Rank 1 solutions using the two-objective minimization problem as an example. On the other hand, Fig. 6 shows an example of a distribution diagram of current solutions

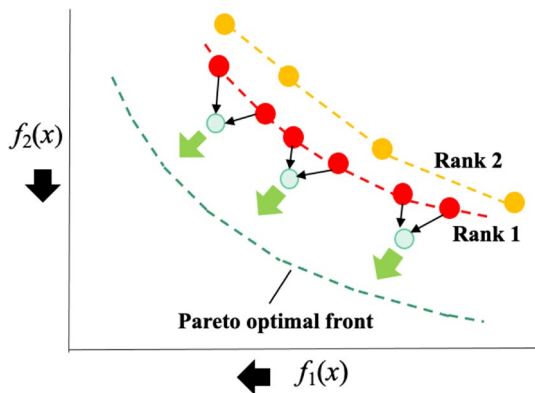


Fig. 5 Searching for a solution using only Rank 1 solutions

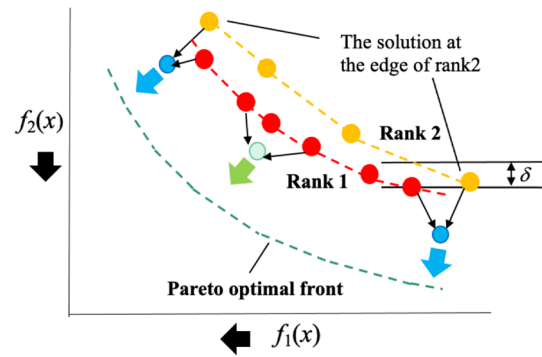


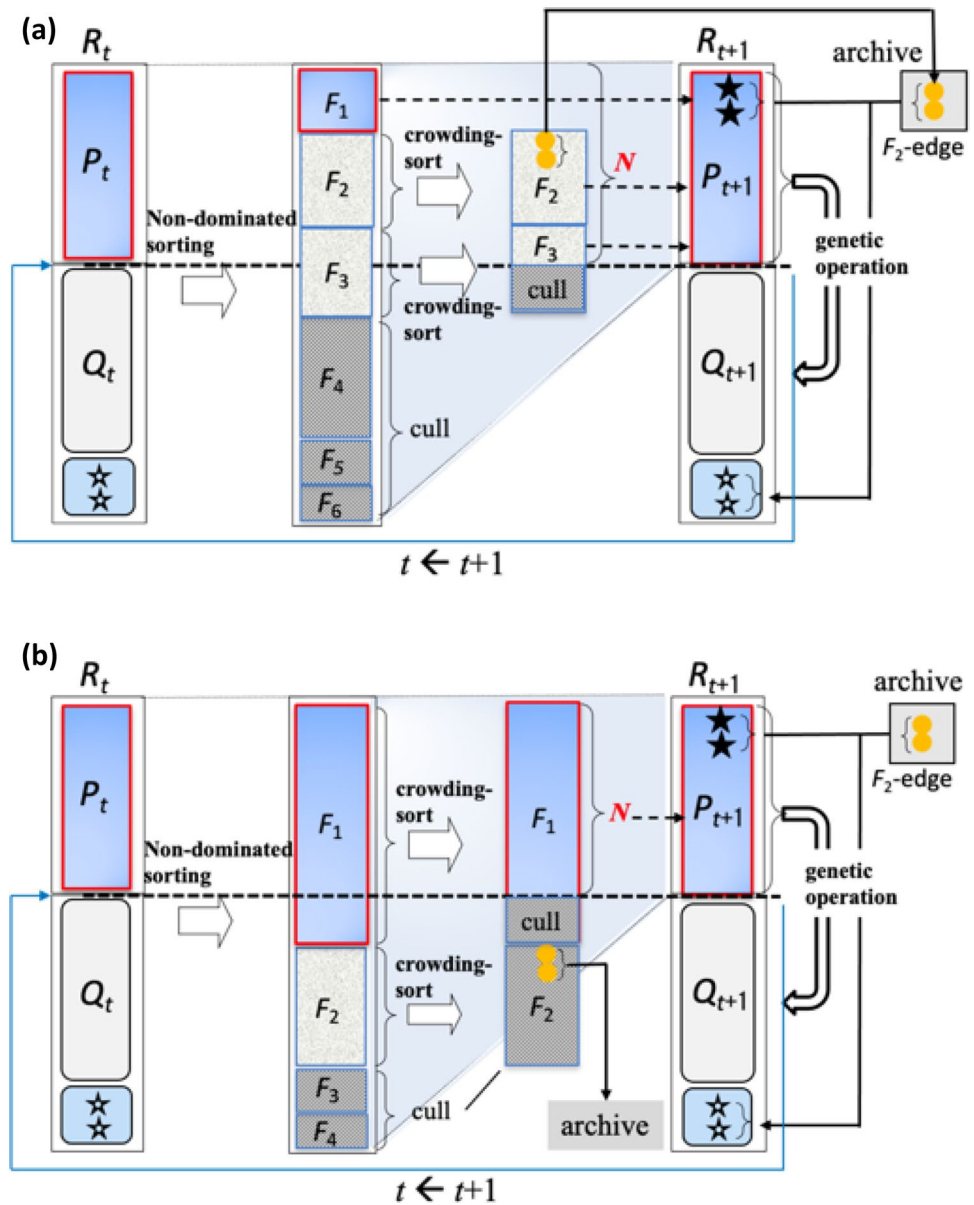
Fig. 6 Including Rank 2 solutions for greater solution diversity

generated by mating the solution candidates at the edges of Rank 1 and Rank 2 according to the proposed method. In the conventional genetic operations shown in Fig. 5, the individuals represented by green circles are generated from Rank 1 parent individuals. Although this improves the convergence to the true Pareto front, it also creates new solution candidates that dominate the original Rank 1 edge solution candidates and reduces the breadth (i.e., diversity) of the solution distribution. In the proposed method shown in Fig. 6, the genetic operations also include dominated solutions at both ends of the Rank 2 solution distribution. Suppose the solution candidate at the end of Rank1 is  $(x_{R1E1}, y_{R1E1})$ , the solution candidate at the end of Rank2 is  $(x_{R2E1}, y_{R2E1})$  ( $x_{R1E1} < x_{R2E1}, y_{R1E1} < y_{R2E1}$ ) and the solution candidate after mating these two solution candidates is  $(x'_{R1E1}, y'_{R1E1})$ . If the difference between the values of  $x_{R1E1}$  and  $x_{R2E1}$  is small, it may be possible to generate solution candidates that satisfy the conditions of  $x_{R1E1} < x'_{R1E1}$  and  $y'_{R1E1} < y_{R1E1}$ . The solution candidate obtained from this mating operation belongs to Rank 1, and is possible that it might become a new solution candidate at the edge of Rank 1, thereby expanding the distribution of solution candidates. This idea can be developed even further. For example, when considering the spread of the solution distribution for the objective function 1, for some small value  $\delta$ , mate all the dominated solutions in the range from  $y_{R1E1}$  to  $y_{R1E1} + \delta$  with the non-dominated solution  $(x_{R1E1}, y_{R1E1})$  at the edge of Rank 1. Here, since the appropriate value of  $\delta$  may depend on the problem, on this occasion we performed a feasibility study by conducting an evaluation experiment for the case where only the edge of Rank 2 is targeted.

#### Leverage with Archived Dominated Solutions

Figure 7 summarizes the method used to generate child individuals by mating the solutions at the edge of Rank 2 stored in the archive population with the solutions at the edge of Rank 1. Figure 7a shows an example when the number of Rank 1 individuals is less than  $N$ , and Fig. 7b shows an

**Fig. 7** **a** NSGA-II making use of an archived dominated solution (the number of Rank 1 is less than  $N$ ). **b** NSGA-II making use of an archived dominated solution (the number of Rank 1 is larger than  $N$ )



example when the number of Rank 1 individuals is larger than  $N$ . In either case, there are no changes to the original NSGA-II regarding the generation of  $P_{t+1}$  and  $Q_{t+1}$  from  $R_t$ , except that the solutions at the edge of Rank 2 are archived. In cases where the number of Rank 1 is less than  $N$ , if the child individuals created by mating the solutions at the edge of Rank 2 stored in the archive population with the solutions at the edge of Rank 1 belong to Rank 1 solutions, there will be more non-dominated solutions, and as shown in Fig. 6, this is highly likely to lead to an increase in the diversity of rank 1 solution sets. On the other hand, when the number of Rank 1 is larger than  $N$ , if the child individuals created by mating the solutions at the edge of Rank 2 stored in the archive population with the solutions at the edge of Rank 1 belong to Rank 1 solutions, since the solutions belonging

to Rank 1 are sorted based on their crowding distance, the solution with the smallest crowding distance (i.e., the most densely distributed solutions) in Rank 1 are replaced.

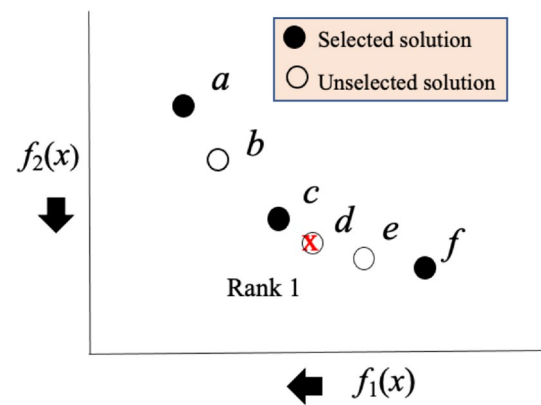
This replacement maintains a constant number of individuals  $N$  and leads to a uniformity of the solution distribution, it is also highly likely to lead to an increase in the diversity of rank 1 solution sets. In both Fig. 7a, b, if a child individual created by mating the solutions at the edge of Rank 2 stored in the archive population with the solutions at the edge of Rank 1 is a poor individual that interferes with search, it will be eliminated by non-dominated sorting and will not adversely affect the generation of  $P_{t+1}$ . Even in the original NSGA-II, if the number of non-dominated solutions for Rank 1 is less than  $N$ , the solution for the edge of Rank 2 is already included in  $P_{t+1}$ . However, the probability of



mating a solution for a Rank 1 edge with a Rank 2 edge is small. On the other hand, in the proposed method, archived Rank 2 edge solution candidates are mated with Rank 1 edge solution candidates with a probability of 1.0.

When the number of Rank 1 solutions is larger than the population size  $N$ , it can help reduce congestion in the solution distribution by replacing the lowest ranked solution in Rank 1, that is, the densely crowded non-dominated solutions. The removal of these solutions thus improves the likelihood of selecting solutions from regions that were densely populated with candidate solutions. Figure 8 shows an example of solution selection based on crowding distance after removing two solution candidates in a region where the solution candidates were densely crowded, resulting in better uniformity. It appears that the creation of children by mating solutions at the ends of Ranks 1 and 2 increases the probability of generating solution candidates for the next generation at both ends of the Pareto front, and may also be effective at improving the distribution uniformity of current solutions by thinning out solutions in regions where solution candidates are densely distributed.

In the original NSGA-II, the diversity of solutions and the spread of the distribution of solutions are maintained by selecting solutions with a large crowding distance, that is, solutions with sparse surroundings. Therefore, when using the crowding distance as shown in Fig. 4, since solution  $b$  having a large crowding distance is selected, solutions



**Fig. 8** An example of crowding distance selection where a high-density cluster of solution candidates has been deleted

from places where the solutions are concentrated are hardly selected at all, which may result in a blank area. On the other hand, Fig. 8 shows an example in which a new Rank 1 solution  $f$  is generated by mating a Rank 1 edge solution and a Rank 2 edge solution, and instead the solution  $d$  having the smallest crowding distance is eliminated. As a result, instead of solution  $b$ , solution  $c$  is selected, and the spread of the solution distribution and the uniformity of the solution distribution are also improved compared with Fig. 4.

### Pseudo Code of the Proposed Method

---

#### Algorithm 1: Pseudo code of the proposed method

---

```

// t: number of generations
// P(t): archive population of size N
// Q(t): search population of size N
// Fi: solution set of rank i
1  t = 0;
2  Generate P(t); // generate initial population
3  Generate Q(t); // by applying genetic operations to P(t);
4  R(t) ← P(t) ∪ Q(t);
5  Until STOPPING CONDITION do
6      Apply non-dominated sort to R(t);
7      Apply crowding sort to R(t);
8      Store the solutions at the edge of Rank 2 in the archive;
9      selection; // archive the top N individuals to P(t) and cull the remaining individuals
10     Generate Q(t); // by applying genetic operations to P(t);
11     Generate offspring by mating the solutions at the edge of Rank 2 stored
        in the archive with the solutions at the edge of Rank 1 with a probability of 1.0
12     t ← t + 1;
13  end

```

---

The pseudo code of the proposed method is summarized in Algorithm 1. In the execution step in the pseudo code, lines 8 and 11 are correspond to the modifications made in the proposed method. In the proposed method, after the crowding sort, the solution at the edge of Rank 2 is stored in the archive, and the child generated by mating the solution stored in this archive with the solution at the edge of Rank 1 is added to  $R_{t+1}$ .

The main points of the proposed method are shown below.

1. Archive the solution candidates for Rank 2 edges.
2. For each objective function, the solution of the edge of Rank 1 and the solution of the edge of Rank 2 are mated with a probability of 1.0, and the resulting offspring are added to  $Q_t$ .
3. If the above added individuals are dominated solutions, they will be weeded out and this algorithm becomes the same as the original NSGA-II. That is, the proposed mating does not adversely affect the original NSGA-II even if no valid offspring can be produced.
4. When the above added individual becomes Rank 1, the number of Rank 1 individuals (non-dominated solutions) increases when the number of Rank 1 individuals is less than  $N$ . If the number of Rank 1 individuals is larger than  $N$ , it will be replaced with the solution candidate at the bottom of Rank 1 (where the solutions are densely packed). In addition, if the added individual becomes Rank 1, this is expected to have the effect of widening both ends of the distribution of Rank 1 solution candidates (regardless of the number of Rank 1 individuals and the size of  $N$ ).

## Evaluation

### Experimental Method

We performed an experimental comparison of the proposed algorithm with the conventional NSGA-II algorithm using the ZDT test suite [1, 8, 9] and a two-objective constrained knapsack problem [10], which are typical test functions for multi-objective optimization problems. ZDT1 has a convex Pareto optimal front and is suitable for evaluating convergence. It is defined by the following equations:

$$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(\mathbf{x}) \\ g(\mathbf{x}) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{n-1} \\ h(\mathbf{x}) = 1 - \sqrt{f_i/g(\mathbf{x})} \end{cases} \quad (3)$$

where  $x_i \in [0, 1], i = 2, \dots, n, n = 30$ .

ZDT2 has a concave Pareto-optimal front and is suitable for evaluating diversity. It is defined by the following equations:

$$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(\mathbf{x}) \\ g(\mathbf{x}) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{n-1} \\ h(\mathbf{x}) = 1 - (f_i/g(\mathbf{x}))^2 \end{cases} \quad (4)$$

where  $x_i \in [0, 1], i = 2, \dots, n, n = 30$ .

ZDT3 has a convex Pareto optimal front and is a complicated problem characterized by a discontinuous Pareto optimal front, and is defined by the following equations:

$$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(\mathbf{x}) \\ g(\mathbf{x}) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{n-1} \\ h(\mathbf{x}) = 1 - \sqrt{f_i/g(\mathbf{x})} \\ \quad - \left( \frac{f_i}{g(\mathbf{x})} \right) \cdot \sin(10\pi f_1) \end{cases} \quad (5)$$

where  $x_i \in [0, 1], i = 2, \dots, n, n = 30$ .

ZDT4 is a multimodal problem with two objectives and ten design variables, which is characterized by the difficulty of finding the Pareto optimal solution  $g(\mathbf{x}) = 1$  due to the wide range of values of design variables other than  $x_1$ , and there are many local convergence areas. It is defined by the following equations.

$$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(x) \left[ 1 - \sqrt{\frac{x_1}{g(x)}} \right] \\ g(x) = 1 + 9 \cdot \sum_{i=2}^n \frac{x_i}{n-1} \\ h(x) = 1 + 10(n-1) \\ \quad + \left( \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \right) \end{cases} \quad (6)$$

where  $x_1 \in [0, 1]$ ,  $x_i \in [-5, 5]$ ,  $i = 2, \dots, n$ ,  $n = 30$ . ZDT6 has a concave Pareto-optimal front and is characterized in that the value in a certain range of  $f_1(x)$  is determined by the value in a very small range of  $x_1$ . It is defined by the following equations.

$$\begin{cases} f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ f_2(\mathbf{x}) = g(x) \left[ 1 - \left( \frac{f_1}{g(x)} \right)^2 \right] \\ g(x) = 1 + 9 \cdot \left( \sum_{i=2}^n \frac{x_i}{n-1} \right)^{0.25} \end{cases} \quad (7)$$

where  $x_i \in [0, 1]$ ,  $n = 30$ .

As a test problem to evaluate the ability to find the boundary solutions, we focus on *mk* knapsack problems (*mk*-KPs) [10]. A *mk*-KP is defined as follows.

$$\begin{cases} \text{Maximize } f_j(x) = \sum_{i=1}^n p_{ij} \times x_j (j = 1, 2, \dots, m) \\ \text{Subject to } \sum_{i=1}^n w_{il} \times x_i \leq c_l (l = 1, 2, \dots, k) \end{cases} \quad (8)$$

This problem has  $n$  items and  $k$  knapsacks, and each item  $i$  has  $m$  profits  $p_{ij}$  ( $j = 1, 2, \dots, m$ ) and  $k$  weights  $w_{il}$  ( $l = 1, 2, \dots, k$ ). The task is to find a set of items  $x = x_1, x_2, \dots, x_n \in \{0, 1\}^n$  that maximize  $m$  objectives while not exceeding  $k$  knapsack capacities  $c_l$ . The knapsack capacity  $c_l$  is defined as follows.

$$c_l = \varphi \sum_{i=1}^n w_{il} (l = 1, 2, \dots, k) \quad (9)$$

where  $\varphi$  is the feasibility ratio for each knapsack (constraint), and we can control the strictness of constraints by varying  $\varphi$ . The *mk*-KP is different from the multi-objective knapsack problem (MOKP) [11] in that the numbers of objectives  $m$  and knapsacks  $k$  can be independently determined. Here, we set the number of items to 300 and the feasibility rate  $\varphi$  to 0.5, and we evaluated the cases of  $(m, k) = (2, 1)$  and  $(3, 2)$ .

Using these six types of test problem, we performed experiments to compare the following four items:

1. Overall solution search performance using hypervolume (HV) values [1, 12, 13].
2. Extent of the distribution of current solutions using maximum spread (MS) [14].
3. Number of non-dominated solutions generated at both ends of the Pareto front.
4. Uniformity of the solution distribution in Pareto front diagrams.

Here, HV is an index representing the overall effectiveness of the solutions, and is defined as the volume (or, in the case of two objectives, the surface area) of the hyperplane formed by the origin and the finally obtained Pareto front. The definition of HV can be expressed as shown in Eq. (10). In Eq. (10),  $n_{PF}$  represents the number of solutions in the Pareto set, and a hypercube  $v_i$  is constructed with a reference point and the solution  $i$  as its diagonal corners.

$$HV = \text{volume}(\cup_{i=1}^{n_{PF}} v_i) \quad (10)$$

By comparing the HV values, we can check that the proposed method does not adversely affect the convergence to the Pareto front or the overall solution search performance.

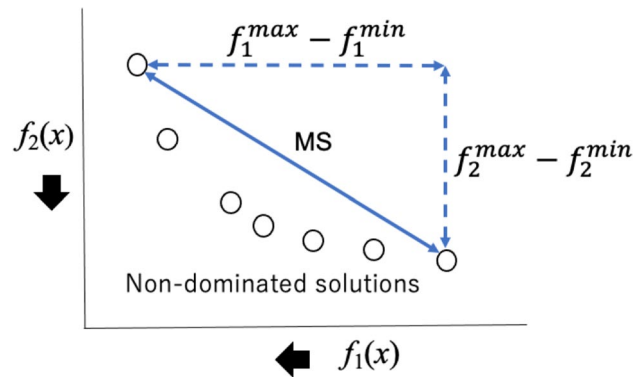


Fig. 9 Overview of the maximum spread (MS)

Table 1 GA parameters and origin point

Population size	20, 100
Max. number of generations	1000
Crossover method	2-point crossover
Crossover rate	0.9
Mutation method	Polynomial mutation
Mutation rate	0.033
Reference point	(1.2, 1.5)



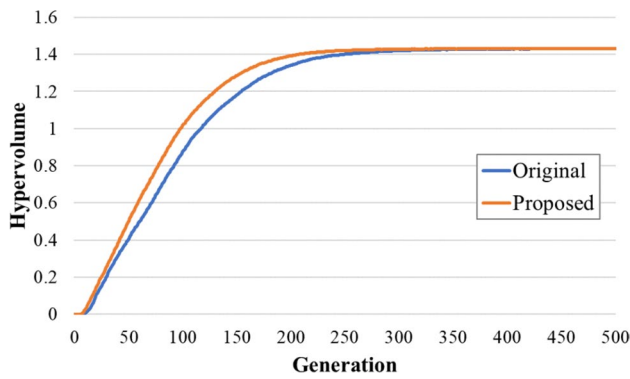


Fig. 10 The relationship between the number of generations and the HV value for ZDT1 (20 individuals)

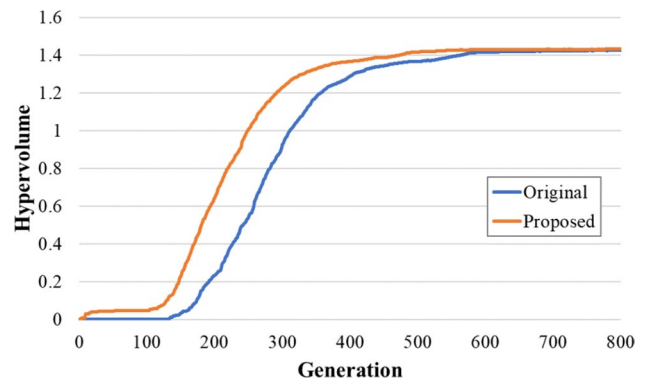


Fig. 13 The relationship between the number of generations and the HV value for ZDT4 (20 individuals)

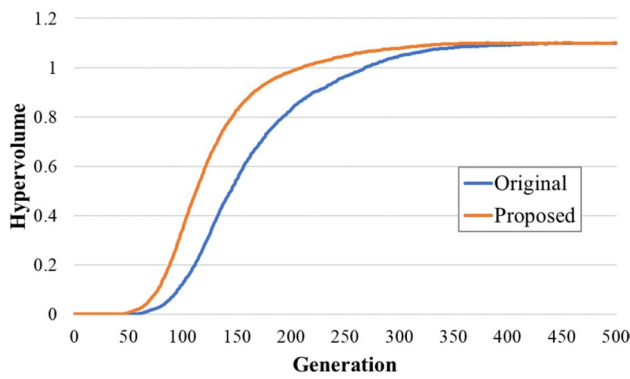


Fig. 11 The relationship between the number of generations and the HV value for ZDT2 (20 individuals)

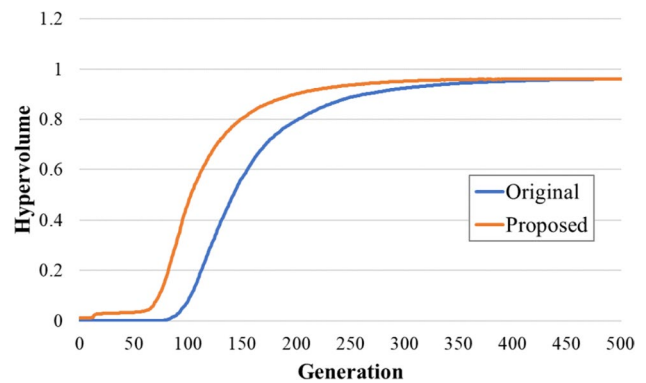


Fig. 14 The relationship between the number of generations and the HV value for ZDT6 (20 individuals)

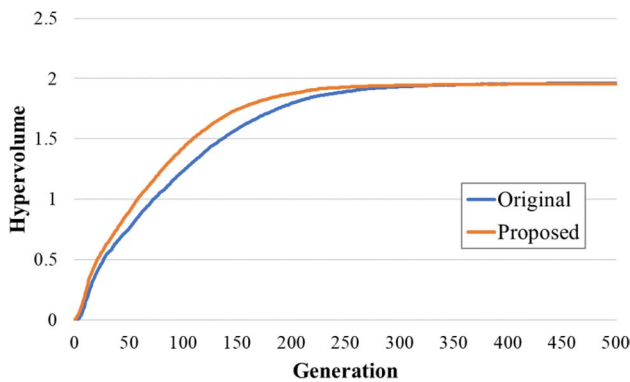


Fig. 12 The relationship between the number of generations and the HV value for ZDT3 (20 individuals)

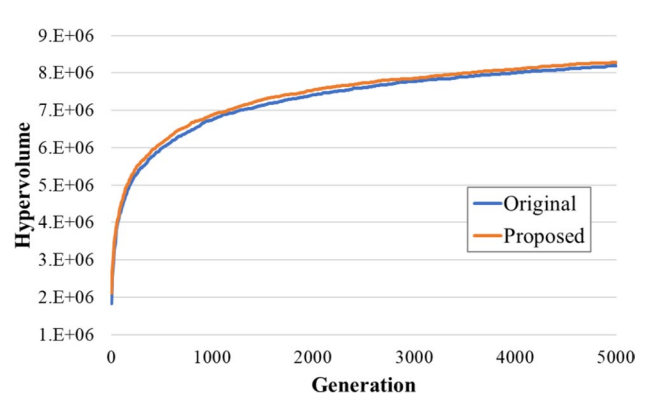


Fig. 15 The relationship between the number of generations and the HV value for knapsack problem (100 individuals)

The MS value is an indicator used to evaluate the spread of a solution set based on the extreme points of the set obtained by searching. A larger MS value indicates a solution set that is more widely distributed. If  $f_k^{\min}$  and  $f_k^{\max}$  are the minimum and maximum values of the objective

function  $f_k$  amongst the set of approximate Pareto solutions obtained by searching, then the MS value is calculated using the formula shown in Eq. (11). A schematic illustration of MS is shown in Fig. 9.

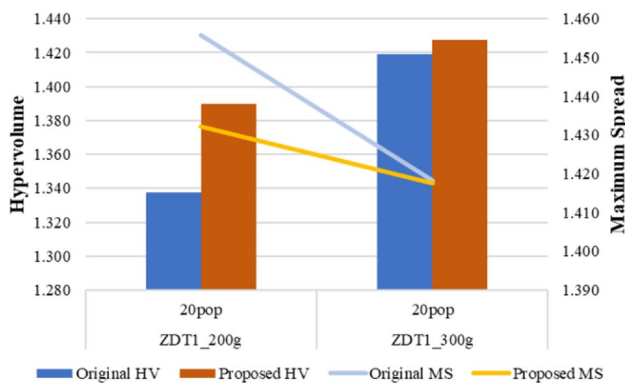


Fig. 16 Comparison of HV and MS values for ZDT1

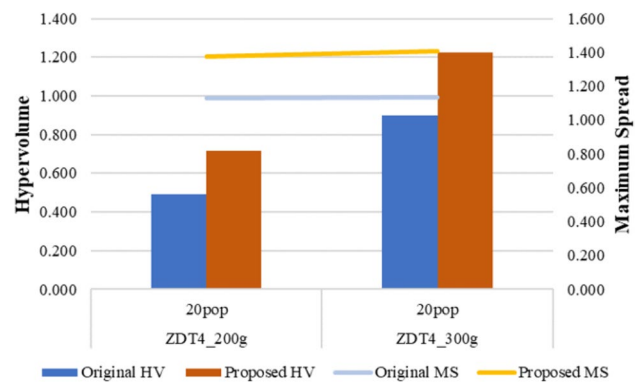


Fig. 19 Comparison of HV and MS values for ZDT4

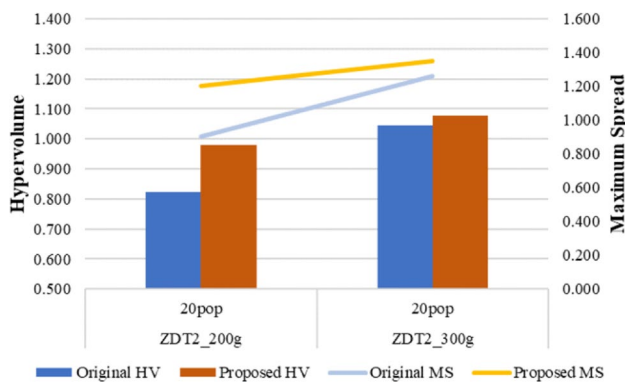


Fig. 17 Comparison of HV and MS values for ZDT2

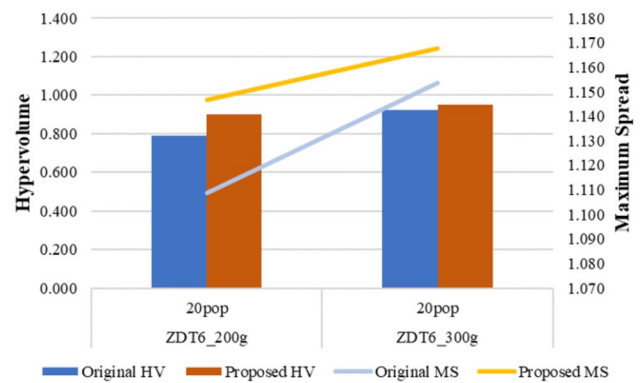


Fig. 20 Comparison of HV and MS values for ZDT6

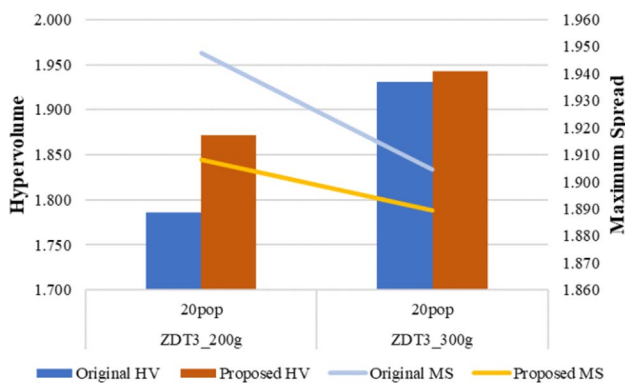


Fig. 18 Comparison of HV and MS values for ZDT3

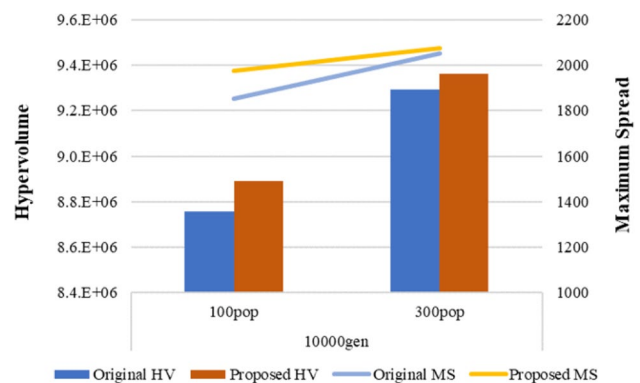
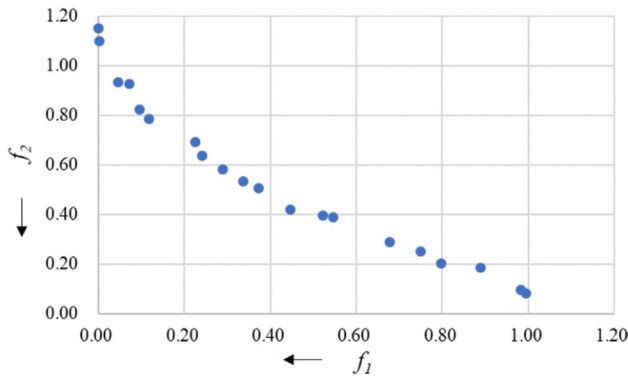


Fig. 21 Comparison of HV and MS values for Knapsacks (10,000th generations)

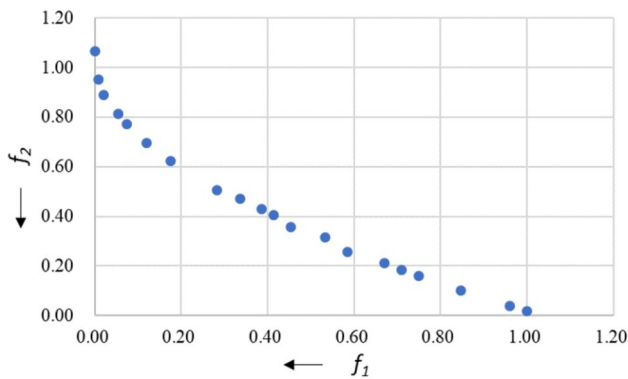
$$MS = \sqrt{\sum_{k=1}^r (f_k^{\max} - f_k^{\min})} \quad (11)$$

By comparing MS values, we can confirm whether the proposed method is effective at improving the diversity of the Pareto front. At the same time, we can compare the

number of non-inferior solutions generated around both ends of the Pareto front to check whether the search performance is improved around the Pareto front extremities. Furthermore, to confirm the effectiveness of this approach at remedying the problem that can arise when selecting solutions based on crowding distance, whereby no solutions are selected from places where solutions are bunched



(a) Conventional method



(b) Proposed method

Fig. 22 Comparison of the distribution of current solutions for ZDT1 (20 pop, 200 gen.)

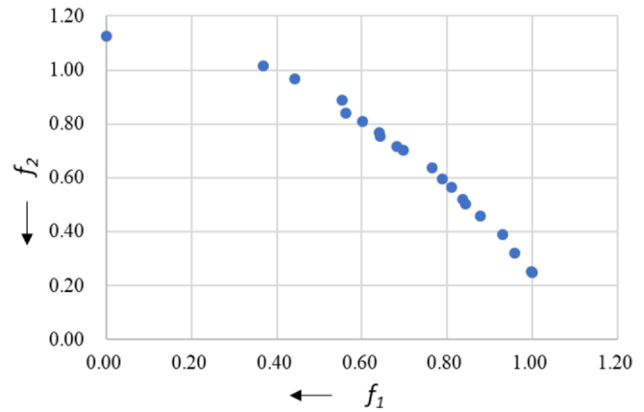
together (leaving empty spaces with no solutions), we compared the uniformity of the solution distributions in the Pareto optimal front. For the HV and MS values, we plotted the average of 31 trials, and for the Pareto front diagram, we plotted the median data.

Table 1 shows the GA parameters used in all the test questions and the origin point used for the calculation of HV values.

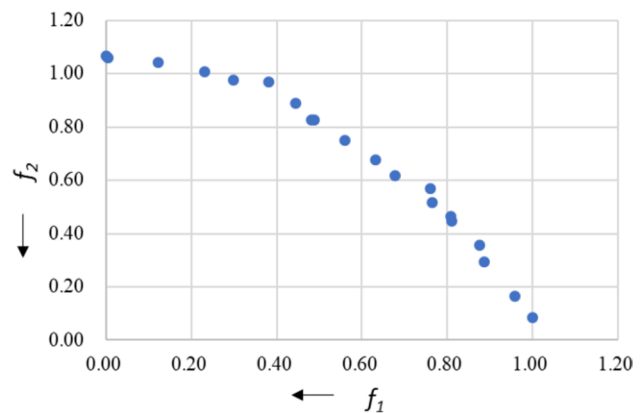
### Experimental Results and Discussion

#### Comparison of HV and MS Values

At first, to check the ability of this method to converge on the Pareto front, we investigated how the HV value varies with the number of generations. Figures 10, 11, 12, 13, 14 and 15 show the relationship between the number of generations and the HV value for each test problem. In the ZDT test suite, the number of individuals is set to 20 because it quickly converges to the Pareto optimal front when the number of individuals is large. For all the test problem, the convergence curve of the proposed method rose sharply over the original NSGA-II method, but no



(a) Conventional method

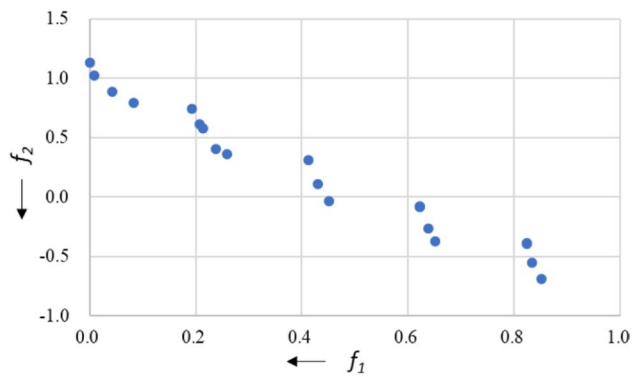


(b) Proposed method

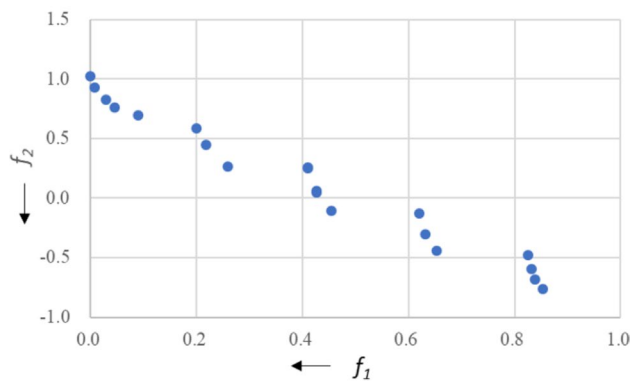
Fig. 23 Comparison of the distribution of current solutions for ZDT2 (20 pop, 200 gen.)

significant differences were observed in the final HV value (probably because both methods eventually have reached the accurate Pareto optimal front). Therefore, the details of the HV and MS values for the 300 generations just before converging to Pareto optimal front and the 200 generations during the search are shown below. However, in the knapsack problem, it is difficult to accurately obtain the Pareto front unless a certain number of individuals are set, so we set the number of generations to 10,000 and compared the cases where the number of individuals is 100 and 300. For the HV and MS values, we plotted the average of 31 trials.

Figures 16, 17, 18, 19 and 20 show the 200th and 300th generation HV and MS values for ZDT1 to ZDT6, respectively. Figure 21 shows the HV and MS values for the constrained knapsack problem for populations 100 and 300 in the 10,000 generation. As shown in these figures, for all test questions, in the 200th generation of the search process, the proposed method tends to show higher HV values than the original NSGA-II, and found that the difference tends to narrow according to converge. And in the problem of reaching



(a) Conventional method



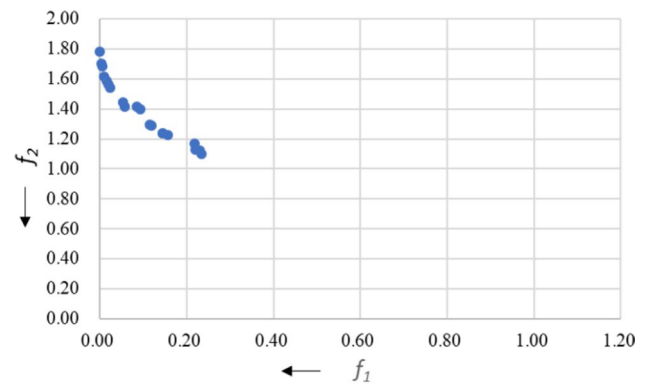
(b) Proposed method

**Fig. 24** Comparison of the distribution of current solutions for ZDT3 (20 pop, 200 gen.)

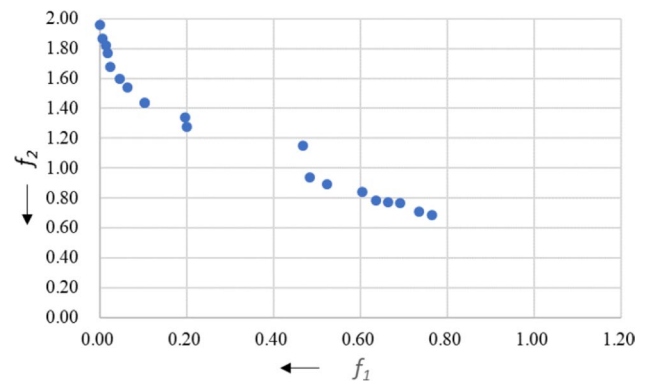
the correct Pareto optimal front, it will eventually reach about the same HV value as the original NSGA-II. On the other hand, in the search process of the 200th generation, the MS value of the proposed method may be higher or lower than that of the original NSGA-II, and found that the difference tends to become narrower due to convergence. Upon closer observation, it was found that the proposed method had a high MS value for problems in which the spread of the solution distribution increased as the search progressed, and a low MS value for problems in which the distribution of current solutions was convergent. That is, the proposed method effectively works to improve diversity (improvement of MS value) and to improve the comprehensive solution search ability (improvement of HV value) in problems that require expansion of solution distribution in the search process.

### Comparison of the Distribution of Current Solutions

Figures 22, 23, 24, 25 and 26 are comparisons when the Pareto front diagram obtained in the 200th generation is applied to the conventional method and the proposed method when the population is 20. However, the knapsack problem



(a) Conventional method

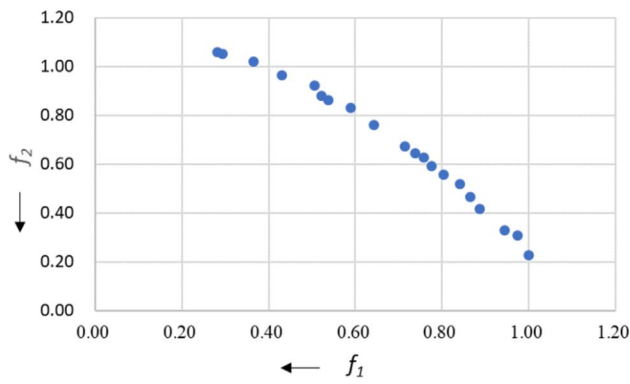


(b) Proposed method

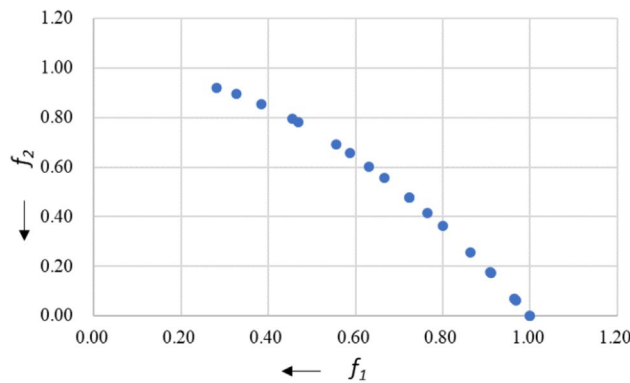
**Fig. 25** Comparison of the distribution of current solutions for ZDT4 (20 pop, 200 gen.)

shown in Fig. 27 is a comparison of Pareto fronts obtained in the 10000th generation when the population size is 100. In addition, the Pareto front diagram is created by converting it so that it becomes a minimization problem like the other test problems. The original NSGA-II displays the median Pareto front of 31 trials sorted by HV value, as plotting the average of 31 trials would obscure the individual trends. On the other hand, the proposed method displays the Pareto front when the same initial value as the original NSGA-II is used. Table 2 summarize the number of non-dominated solutions contained in the Pareto front shown in Figs. 22, 23, 24, 25, 26 and 27 for the original NSGA-II and the proposed method.

From Fig. 22, in the case of ZDT1, it can be seen that the proposed method does not show a large difference in the distribution spread of current solutions but does improve their distribution uniformity. In the case of ZDT2 from Fig. 23, the solution is generated by the conventional method in a sparse region where no solution exists, and the distribution spread of the current solutions is improved for objective function 2. In ZDT3 shown in Fig. 24, there is no difference in the distribution spread of the current solutions, but



(a) Conventional method

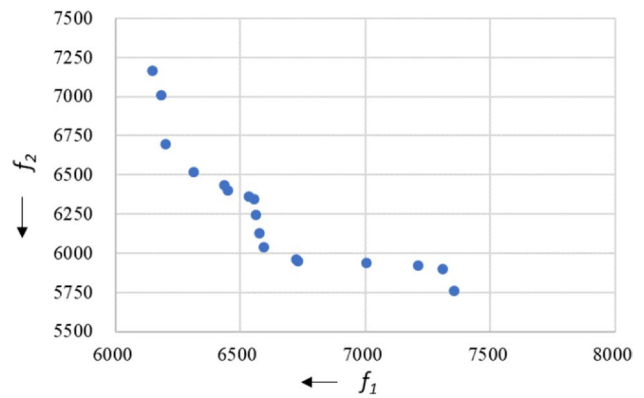


(b) Proposed method

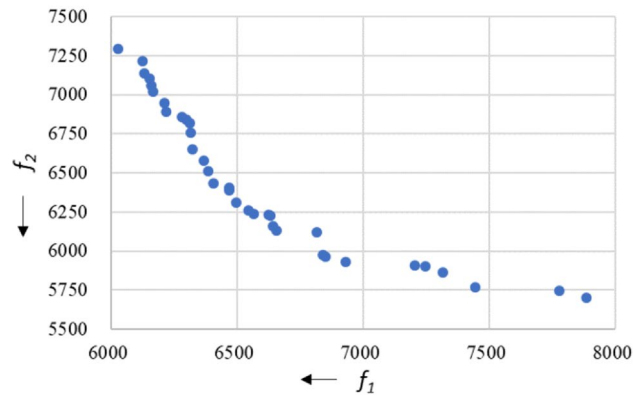
Fig. 26 Comparison of the distribution of current solutions for ZDT6 (20 pop, 200 gen.)

looking at each group of solution groups, it can be seen that the solutions are more uniformly dispersed than in the conventional method, and that there are more non-dominated solutions forming islands on the left and right ends. From Fig. 25, in the case of ZDT4, the distribution spread of the current solutions is greatly improved for both objective functions 1 and 2 by the proposed method. From Fig. 26, in ZDT6, the distribution of current solutions for the objective function 2 is improved, and so is the distribution uniformity of current solutions. In the case of the knapsack problem shown in Fig. 27, the proposed method greatly improves the distribution spread of the current solutions for both objective functions 1 and 2, and also improves the distribution uniformity of current solutions.

From Table 2, the number of non-dominated solutions for ZDT1 to ZDT6 is 20, which reached the value of the population size during the solution search. Therefore, these are the cases in Fig. 7b, and the number of non-dominated solutions does not increase even if the search is continued. This can alleviate the problem of solution distribution non-uniformity that occurs when using crowding distances. These experimental results are in line with the idea that if



(a) Conventional method



(b) Proposed method

Fig. 27 Comparison of the distribution of current solutions for knapsack (100 pop, 10,000 gen.)

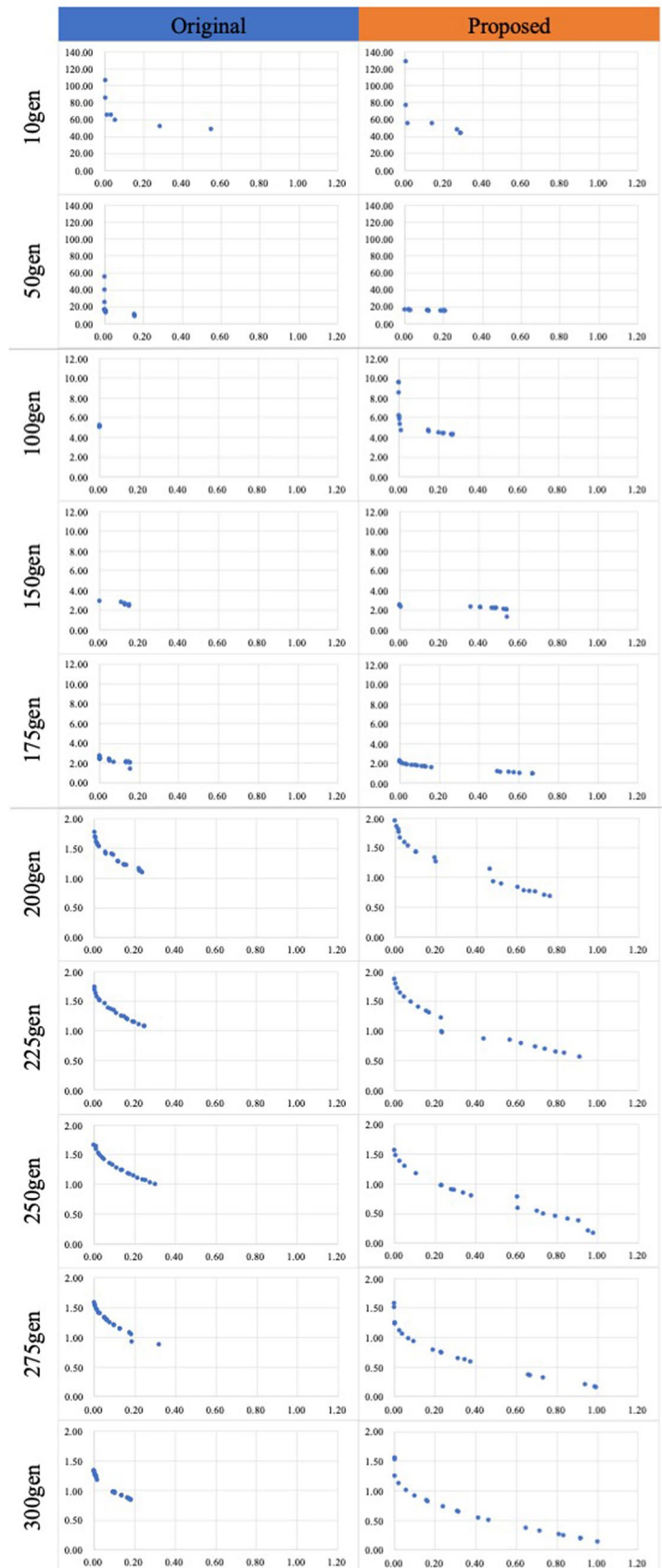
Table 2 Number of non-dominated solutions in Pareto Front (ZDT test suite: 20 pop, Knapsack: 100 pop)

Test problem	Conventional method	Proposed method
ZDT1	20	20
ZDT2	20	20
ZDT3	20	20
ZDT4	20	20
ZDT6	20	20
Knapsack	17	35

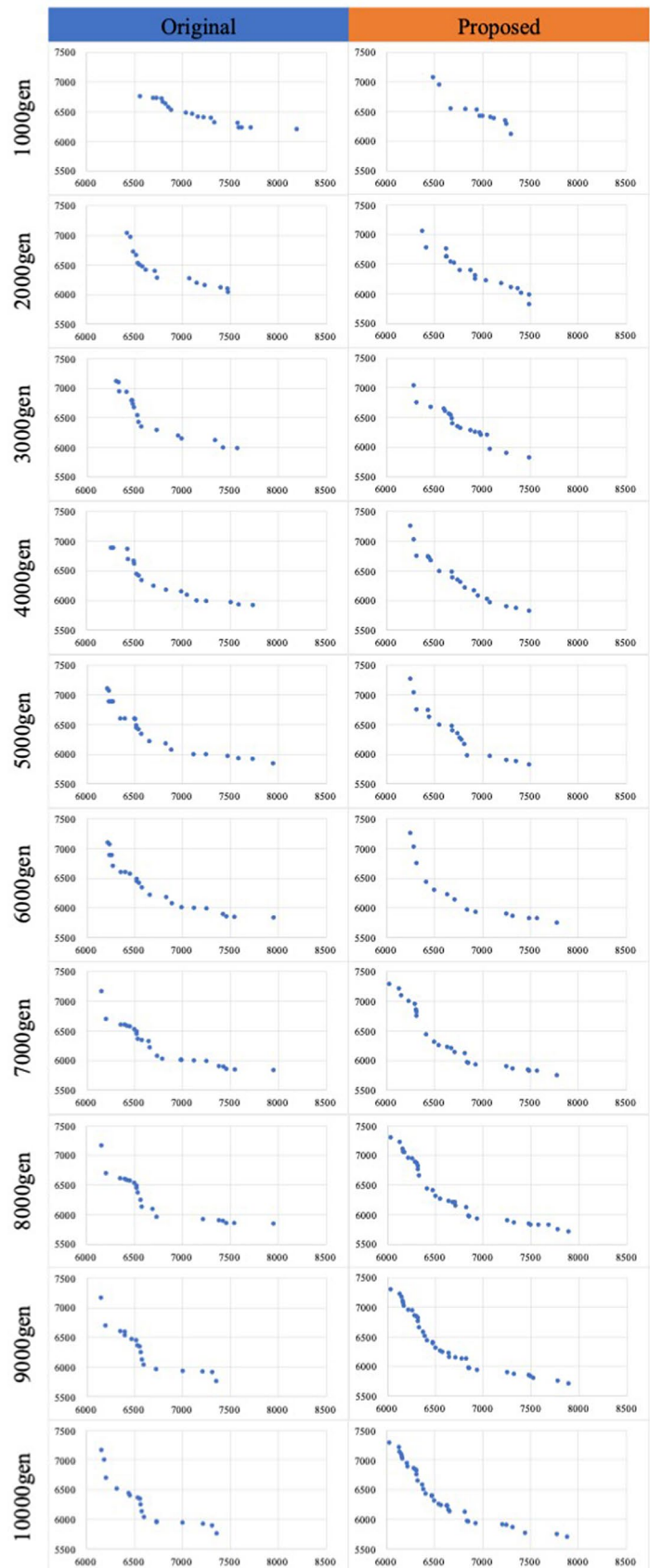
the offspring produced by mating the solutions at the ends of Rank 1 and Rank 2 contribute to the solution search, it may lead to improved spread and the distribution uniformity of current solutions. On the other hand, in the knapsack problem, the number of non-dominated solutions is smaller than the number of individuals, so it corresponds to the case of Fig. 7a. Since the number of non-dominated solutions of the proposed method is larger than that of the conventional

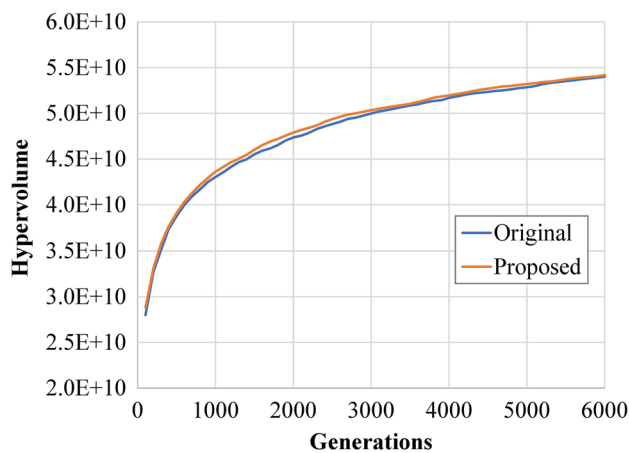


**Fig. 28** Comparison of the transition diagram of the shape of the Pareto front of ZDT4



**Fig. 29** Comparison of the transition diagram of the shape of the Pareto front of KP





**Fig. 30** The relationship between the number of generations and the HV value for knapsack problem (300 individuals)

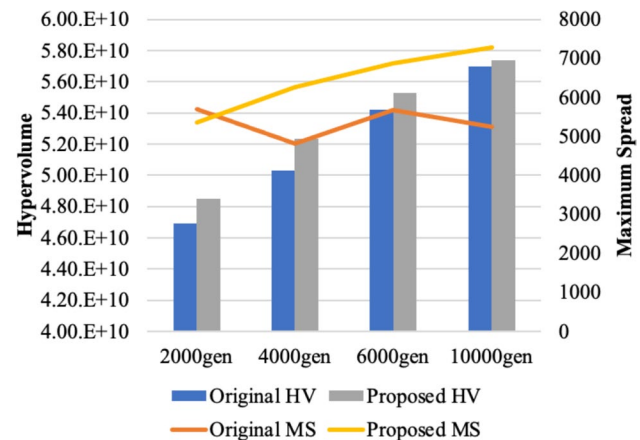
method, it is believed that effective offspring for improved spread and uniformity of the solution distribution were generated by mating the solutions at the ends of Rank1 and Rank2.

### Comparison of Pareto Front Shape Transitions

Figures 28 and 29 show the changes in the shape of the Pareto front as the number of generations increases with respect to the ZDT4 and knapsack problems, for which the proposed method achieved remarkable spreading in the distribution of current solutions. These graphs show the changes that occur before arriving at the Pareto fronts shown in Figs. 25 and 27 in “Comparison of the distribution of current solutions” section, respectively.

In the case of ZDT4, the conventional method seems to have a strong tendency to converge on the Pareto front in the lower left direction for all generations. On the other hand, in the proposed method, in addition to the force acting in the lower left direction, it can be seen that there is a force acting so as to expand the distribution of current solutions in the lower right direction works after about 100 generations. This is thought to be because the offspring produced by mating the solutions at the ends of rank 1 and rank 2 worked effectively in the solution search. In addition, since the number of non-dominant solutions reached 20 in the 175th generation, the offspring replace to the individuals in the region where the distribution of current solutions is dense, which is effective in improving the distribution uniformity of current solutions.

On the other hand, in the case of the knapsack problem, there is no significant difference during the initial stages of the search, but from around 7000 generations, it seems that the proposed method has the effect of widening both ends of the Pareto front and the effect of making the solution



**Fig. 31** Comparison of HV and MS values for Knapsacks (2000, 4000, 6000 and 10,000th generations)

distribution uniform. It can also be observed that the number of non-dominated solutions increases compared to the conventional method. In the conventional method, there tends to be a strong force driving convergence towards the Pareto front in the lower left direction, which is the same as in the case of ZDT4. In addition, in the conventional method, the solution candidates (at the 8000th generation) at edges found in the search process are eliminated when other solution candidates that dominates this individual are found (at the 9000th generation), so the distribution spread of the current solutions cannot be maintained. On the other hand, in the proposed method, when the offspring produced by mating the solutions at the edges of Rank 1 and Rank 2 work effectively in the solution search, they converge toward the true Pareto front while maintaining the distribution spread of the current solutions.

Here, if the original NSGA-II is searching for a solution with high accuracy, applying the proposed method will not have any further improvement. It was also observed that the HV value decreased slightly depending on the initial value. However, the superiority of the proposed method is clear from the average of 31 trials, and the comparison of the median Pareto front also shows that it tends to effectively improve the spread and uniformity of the solution distribution.

### The Experiments on Tri-objective Optimization Problems

Here, we confirm the effectiveness of increasing the number of objectives from 2 to 3 for knapsack problems where it is difficult to maintain the distribution of solutions during the search process.

#### 1. Comparison of HV and MS values

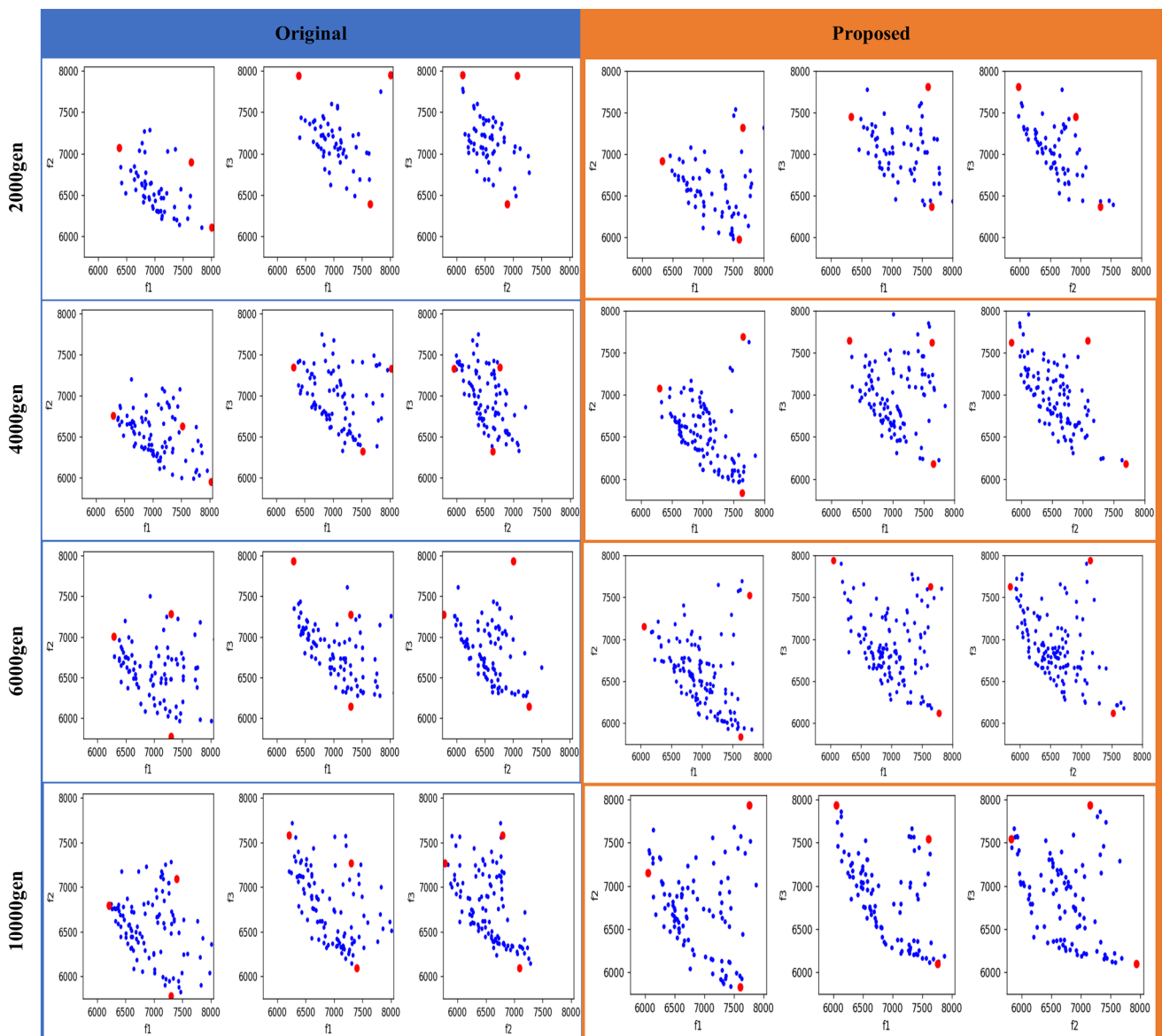


Fig. 32 Comparison of the transition diagram of the shape of the 2D-Pareto front of KP

Figure 30 shows the relationship between the number of generations and the HV value for the knapsack problem when  $(m, k) = (3, 2)$ . We set the number of generations up to 10,000 and compared the cases where the number of individuals is 300. For the HV and MS values, we plotted the average of 31 trials. Figure 31 show the HV and MS values of the knapsack problem at the 2000th, 4000th, 6000th and 10000th generations. As shown in these figures, for all generations, the proposed method tends to show higher HV values than the original NSGA-II, although the difference tends to decrease as the algorithms converge. However, in the search process of the 2000th generation, the MS value of the proposed method was about the same as the original

NSGA-II. In other words, the proposed method shows the effect of effectively improving diversity (improvement of MS value) and comprehensive solution search ability (improvement of HV value) even if the number of objectives is increased from 2 to 3.

2. Comparison of Pareto front shape transitions

Figure 32 shows the transition in the shape of the Pareto front for knapsack problems as the number of generations increases. Since it is difficult to confirm visually the diversity in the minimization problem in 3D space, the results are shown mapped to the  $x-y$ ,  $y-z$  and  $z-x$  planes to confirm the convergence, spread, and uniformity of the solution distribution. The original NSGA-II displays the median Pareto front of 31 trials

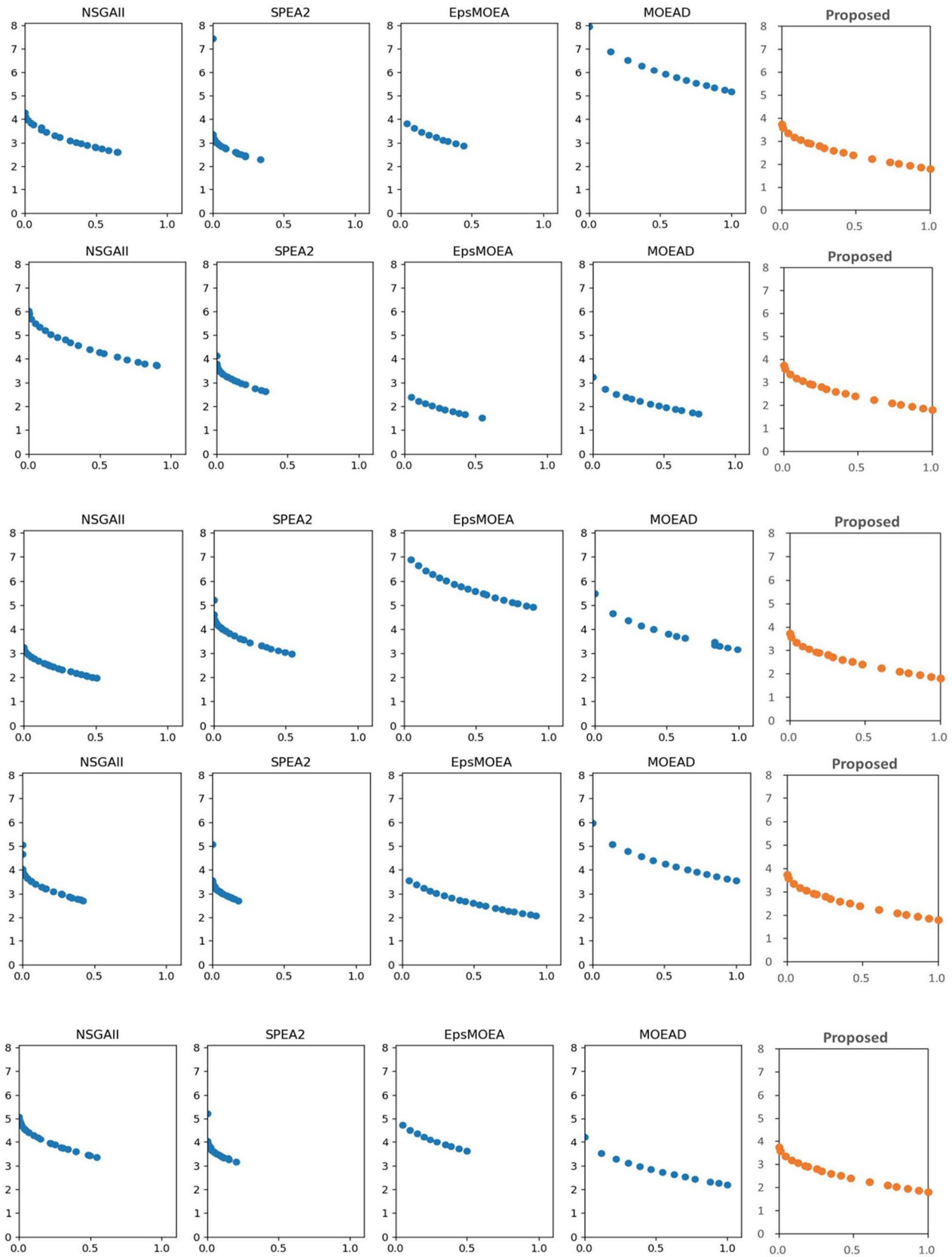


Fig. 33 Non-dominated solutions generated for problem ZDT4 at the 3000 generations



sorted by HV value, as plotting the average of 31 trials would obscure the individual trends. On the other hand, the proposed method displays the Pareto front when using the same initial value as the original NSGA-II.

There was no significant difference during the first 2000 generations, but from around 4000 generations, it seems that the proposed method has the effect of widening both ends of the Pareto front at the  $z$ - $x$  plane, which has the effect of making the solution distribution uniform. At the 10,000th generation, it can be seen that the proposed method improves the spread and uniformity of the solution distribution in all of the  $x$ - $y$ ,  $y$ - $z$ , and  $z$ - $x$  planes compared to the original NSGA-II. In the conventional method, there tends to be a strong force driving convergence towards the Pareto front in the lower left direction, which is the same tendency as in the case of two-objective problems.

### Effect on the Spread of Solution Distribution

Here, we will consider the effect when the proposed method works effectively. Figure 33 compares the shape of the Pareto front with NSGA-II, SPEA2,  $\epsilon$ -dominant MOEA [15], MOEA/D [16], and the proposed method in the 3000 generations when the search was started from the same initial population. In the comparative experiment in this section, our proposed method is improved based on the program (in C language) downloaded from Deb [17] and modified for the experiments. For the other algorithms, evaluation experiments are conducted using the Platypus Library [18]. When Platypus was used, even if the search was started from the same initial population, the results were different each time it was executed. Therefore, Fig. 33 lists and compares the results of five trials. From Fig. 33, it can be observed that  $\epsilon$ -dominance MOEA and MOEA/D tend to be superior to NSGA-II and SPEA2 in terms of the spread of the solution distribution. On the other hand, NSGA-II has better convergence to Pareto optimal pronto than  $\epsilon$ -dominance MOEA and MOEA/D. The reason for the small spread of the SPEA2 solution distribution may be due to the fact that SPEA2 requires a large number of generations for solution search, so 3000 generations may not have been sufficient. The proposed improvement shows superior results to  $\epsilon$ -dominance MOEA and MOEA/D in terms of the spread of solution distribution, and maintains the convergence of the original NSGA-II. The proposed method does not always work effectively, but if it works, it can be expected to have a great effect on the spread of the solution distribution.

### Comparison with Related Research

An approach similar to the proposed method was studied by Sato, who focused on controlling the balance between convergence and diversity in a search for solutions where the objective function is transformed by introducing control parameters to modify the dominated and non-dominated relation regions of candidate solutions [19]. This approach works effectively when suitable control parameters have been set. However, it is not easy to set suitable control parameter values unless the rough shape of the Pareto front is known in advance. The constant changes in the shape of the Pareto front at the beginning of the search also make it difficult to determine appropriate control parameter values in advance. On the other hand, our proposed method has the advantage that it does not require any prior knowledge of the Pareto front geometry and can be implemented with only a small modification to the original NSGA-II program.

Another paper [20] describes the use of Pareto partial dominance in a study that focuses on countermeasures to be used when search using dominance/non-dominance relations does not work as effectively in a similar manner to the proposed method. This is a two-step NSGA-II method that cuts out partial multi-objective problems from many objectives to address the problem whereby, in multi-objective optimization problems with four or more objectives, almost all the solution candidates are of Rank 1, and the search for solutions using dominance/non-dominance relations does not work effectively. This method is thought to be effective when there are four or more objectives, but when applied to a multi-objective problem with fewer objectives, a single objective problem will be generated. It also requires much more computation time than the original NSGA-II. Furthermore, algorithms with better search capability, such as MOEA/D and NSGA-III [21], have been proposed for many-objective optimization problems with more than four objectives. On the other hand, our proposed method is also effective for multi-objective optimization problems where there are three or fewer objectives, and can easily combined with algorithms such as [22] that focus on improving convergence to the Pareto front.

Another method called Cone  $\epsilon$ -dominance MOEA has also been reported [23]. This is an improved version of  $\epsilon$ -dominance that is effective at improving the bias of the solution distribution and ensuring uniformity in the finally obtained rank 1 solution group. In [23], the ZDT and the DTLZ families are used as benchmark problems, and the results show that the uniformity of the solution distribution is significantly improved compared to the original NSGA-II.

On the other hand, since additional calculation is required for all non-dominated solutions, its worst-case computational load is  $O((mN)^2)$ , where  $m$  is the objective number and  $N$  is the number of individuals. As a result, when the number of individuals increases, the increase in calculation time cannot be ignored. On the other hand, the proposed method can be expected to increase the number of new rank 1 solutions in the process of solution search to widen the solution distribution and to improve the uniformity of the solution distribution. In addition, it can be implemented with a slight program modification to the original NSGA-II and has the advantage of low calculation cost. The proposed method does not preclude the use of Cone  $\epsilon$ -dominance, and the two techniques can be used in combination.

In this study, we evaluated the proposed method using the ZDT test suite and a two-objective constrained knapsack problem. However, in the future it will also be, necessary to evaluate using three or more objective test problems such as DTLZ test suite [24] and WFG test suite [25]. A more detailed comparison with related studies such as SPEA2,  $\epsilon$ -dominance MOEA, Cone  $\epsilon$ -dominance MOEA is also important. Furthermore, this time, the experiment was conducted without changing the reference point values of the previous experiment, but there is a paper [26] that the solution accuracy greatly depends on the reference point value. Therefore, an evaluation experiment with different reference points is also necessary.

## Conclusions

In this paper, we have proposed a method whereby, in the NSGA-II evolutionary multi-objective optimization algorithm, some of the dominated solutions outside Rank 1 that would normally be culled during the search process are instead preserved and actively used for genetic operations, which may be an effective way of actively improving diversity. More exactly, we have proposed a method that uses the offspring produced by mating the solutions at the edges of Rank 1 and Rank 2, in the solution search. Using the typical ZDT test suite and a two-objective constrained knapsack problem, we have shown that the proposed method is effective in increasing the hyper-volume value in the early stages of the search. We also may show that the proposed method can help to increasing the number of non-dominated solutions when the number of non-dominated solutions is less than the population size, and to improve the uniformity of the solution distribution when the number of non-dominated solutions is larger than the population size.

In the future, it will be necessary to evaluate using three or more objective test problems such as DTLZ test suite

and WFG test suite, and A more detailed comparison with related studies is also important.

**Acknowledgements** This work was supported by JSPS KAKENHI Grant Number JP19K12162.

**Author contributions** Not applicable.

**Funding** This work was supported by JSPS KAKENHI Grant Numbers JP19K12162.

**Availability of data and materials** Not applicable.

**Code availability** Not applicable.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Additional information** This article incorporates revisions to the proposed method published at ECTA2020, and is updated with new benchmark problems, evaluation and consideration of the use of maximum spread (MS), and a discussion on the problem of conventional solution selection.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Deb K. Multi-objective optimization using evolutionary algorithms. Chichester: John Wiley and Sons; 2001.
2. Carlos A, Coello C. Evolutionary multi-objective optimization: a historical view of the field. *Comput Intell Mag IEEE*. 2006;1(1):28–36. <https://doi.org/10.1109/MCI.2006.1597059>.
3. Deb K, Agrawal S, Pratap A, Meyarivanet, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: International conference on parallel problem solving from nature. Springer; 2000. p. 849–58. [https://doi.org/10.1007/3-540-45356-3\\_83](https://doi.org/10.1007/3-540-45356-3_83).
4. Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput*. 2002;6(2):182–97. <https://doi.org/10.1109/4235.996017>.
5. Zitzler E, Laumanns M, Thiele, L. SPEA2: improving the performance of the strength pareto evolutionary algorithm. In: Technical report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology. 2001. <https://doi.org/10.3929/ethz-a-004284029>.

6. Yokoyama K, Sato Y. Using dominated solutions to the uniformity of non-dominated solution distributions in NSGA-II. In: The 12th international joint conference on computational intelligence (IJCCI 2020). 2020. p. 212–19. <https://doi.org/10.5220/0010143502120219>.
7. Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evolut Comput.* 1995;2(3):221–48. <https://doi.org/10.1162/evco.1994.2.3.221>.
8. Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolut Comput.* 2000;8(2):173–95.
9. Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Proceedings of the 2002 congress on evolutionary computation; 2002. p. 825–30. <https://doi.org/10.1109/CEC.2002.1007032>.
10. S. D. at KanGAL, multi-objective NSGA-II code in C, revision 1.1.6, July 2011. <http://www.iitk.ac.in/kangal/codes.shtml>. Accessed 12 May 2021.
11. Kellerer DPH, Pferschy U. Knapsack problems. London: Springer; 2004. <https://doi.org/10.1007/978-3-540-24777-7>.
12. Zitzler E, Thiele L. An evolutionary algorithm for multiobjective optimization: the strength pareto approach. Tech. Rep. 43, Swiss Federal Institute of Technology; 1998. <https://sop.tik.ee.ethz.ch/publicationListFiles/zt1998a.pdf>. Accessed 12 May 2021.
13. Beume N, Fonseca CM, Lopez-Ibanez M, Paquete L, Vahrenhold J. On the complexity of computing the hypervolume indicator. *IEEE Trans Evolut Comput.* 2009;13(5):1075–82. <https://doi.org/10.1109/TEVC.2009.2015575>.
14. Zitzler E. Evolutionary algorithms for multiobjective optimization: methods and applications. Ph. D. thesis, Swiss Federal Institute of Technology, Zurich. 1999. <https://sop.tik.ee.ethz.ch/publicationListFiles/zitz1999a.pdf>. Accessed 12 May 2021.
15. Deb K, Mohan M, Mishra S. Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolut Comput.* 2005;13(4):501–25. <https://doi.org/10.1162/106365605774666895>.
16. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput.* 2007;12(6):712–31. <https://doi.org/10.1109/TEVC.2007.892759>.
17. Deb. <https://www.egr.msu.edu/~kdeb/codes.shtml>. Accessed 31 May 2022.
18. Platypus—multiobjective optimization in Python. <https://platypus.readthedocs.io/en/latest/>. Accessed 31 May 2022.
19. Sato H, Aguirre H.E, Tanaka K. Local dominance including control of dominance area of solutions in MOEAs. In: Proceedings the 2007 IEEE Symp. on computational intelligence in multicriteria decision-making (MCDM 2007). IEEE; 2007. p. 310–7. [http://vigir.missouri.edu/~gdesouza/Research/Conference\\_CDs/IEEE\\_SSCI\\_2007/Multicriteria%20Decision%20Making%20-%20MCDM%202007/data/papers/MCDM/S001P048.pdf](http://vigir.missouri.edu/~gdesouza/Research/Conference_CDs/IEEE_SSCI_2007/Multicriteria%20Decision%20Making%20-%20MCDM%202007/data/papers/MCDM/S001P048.pdf). Accessed 12 May 2021.
20. Ohki M. Linear subset size scheduling for many-objective optimization using NSGA-II based on Pareto partial dominance. In: Proceedings of the 15th international conference on informatics in control, automation and robotics (ICINCO 2018). 2018. p. 277–83. <https://doi.org/10.5220/0006905402770283>.
21. Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. *IEEE Trans Evolut Comput.* 2014;18(4):577–601. <https://doi.org/10.1109/TEVC.2013.2281535>.
22. Miyakawa M, Takadama K, Sato H. Controlling selection area of useful infeasible solutions and their archive for directed mating in evolutionary constrained multiobjective optimization. In: Proceedings of the 2014 genetic and evolutionary computation conference (GECCO 2014). 2014. p. 629–36. <https://doi.org/10.1145/2576768.2598313>.
23. Batista LS, Campelo F, Guimarães FG, Ramírez JA. The cone epsilon-dominance: an approach for evolutionary multiobjective optimization. *CoRR abs/2008.04224*. 2020. <https://doi.org/10.48550/arXiv.2008.04224>.
24. Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R, editors. Evolutionary multiobjective optimization. Theoretical advances and applications. Berlin: Springer; 2005. p. 105–45. [https://doi.org/10.1007/1-84628-137-7\\_6](https://doi.org/10.1007/1-84628-137-7_6).
25. Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evolut Comput.* 2006;10(5):477–506.
26. Li M, Yao X. Quality evaluation of solution sets in multiobjective optimisation: a survey. *ACM Comput Surv.* 2019;52(2):1–43. <https://doi.org/10.1145/3300148>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.