The Effects of Crowding Distance and Mutation in Multimodal and Multi-objective Optimization Problems

Mahrokh Javadi*
Faculty of Computer Science, Otto von Guericke University
Universitätspl. 2, 39106 Magdeburg, Germany
Email:mahrokh1.javadi@ovgu.de

Heiner Zille
Faculty of Computer Science, Otto von Guericke University
Magdeburg, Germany
Email:heiner.zille@ovgu.de

Sanaz Mostaghim
Faculty of Computer Science, Otto von Guericke University
Universitätspl. 2, 39106 Magdeburg, Germany
Email:sanaz.mostaghim@ovgu.de

Summary

In this paper, we study the effects of a modified crowding distance method and a Polynomial mutation operator on multimodal multi-objective optimization algorithms. Our goal is to provide an in-depth analysis on these two modifications which we apply to NSGA-II: The weighted sum crowding distance and the neighborhood mutation operator. Furthermore, we examine the performance of the proposed weighted sum crowding distance method under different weight values, to find a trend for the behaviour of the proposed algorithm. We compare the different variations of the proposed method with the state-of-the-art algorithms and the baseline NSGA-II. The results show that we can improve the functionality of NSGA-II on multi-modal multi-objective problems.
Keywords: *Multi-modality, Multi-modal problems, Multi-objective Optimization, Evolutionary Algorithms, Non-dominated Sorting Genetic Algorithm.*

1 Introduction

In real-world applications, there are many problems with several conflicting objectives which need to be optimized at the same time. These problems are usually referred to as *Multi-Objective Problems* (MOP). In such problems, improving one of the objectives can affect satisfaction of other objectives.[1] Multi-objective optimization problems are mathematically formulated as follows (we consider minimization problems, without loss of generality):

$$\begin{aligned} \text{minimize} \quad & \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ..., f_M(\vec{x})) \\ \text{subject to} \quad & \vec{x} \in S \subset \mathbb{R}^D \\ & g_i(\vec{x}) \leq 0, i = 1, 2, ..., k \\ & h_j(\vec{x}) = 0, j = 1, 2, ..., p \end{aligned} \tag{1}$$

where $\vec{x} = (x_1, x_2, ..., x_D)$ is considered as a D–dimensional decision vector and $(f_1, f_2, ..., f_M)$ is a M–dimensional objective vector. $g_i(\vec{x})$ and $h_j(\vec{x})$ are inequality and equality constraints in decision space. In order to deal with these problems, the concept of domination can be used. Given two vectors $\vec{x}$, $\vec{y} \in S$, $\vec{x}$ is said to be dominated by $\vec{y}$ (denoted by $\vec{y} \prec \vec{x}$) if and only if $\forall j \in \{1, .., M\}$, $f_j(\vec{y}) \leq f_j(\vec{x})$, and $\exists k \in \{1, .., M\}$, $f_k(\vec{y}) < f_k(\vec{x})$.[2]

A solution which is not dominated by any other solution in the decision space is called a Pareto-optimal solution. The set of such optimal solutions in the decision space is called *Pareto-Set* (PS), and the corresponding solutions in the objective space are called *Pareto-Front* (PF).[3] The goal of multi-objective optimization algorithms is to find a set of non-dominated solutions with a good approximation of the PF both in terms of convergence and diversity.[1]

By using the definition of domination, there is no guarantee that finding all of the solutions in the PF leads to finding all the solutions which actually belong to the PS, since two solutions in decision space might map into one point in the objective space. This class of problems is referred to as multimodal multi-objective problems by Liang and Qu.[2] More precisely, in a multimodal problem, there are multiple subsets of the PS which map to the same objective function values. Therefore, the PF can be approximated by just finding one of these subsets of the PS. However, decision makers are very often interested in a high diversity in both decision space and objective space simultaneously. Therefore, it might be practical to develop algorithms that can find multiple Pareto-optimal solutions in both decision and objective space for such multimodal problems. In this paper, we modify the concept of crowding distance in both decision and objective spaces and investigate a neighborhood mutation operator. Both these approaches are based on our preliminary short study.[4] In this paper, we introduce them and evaluate their performances in both decision and objective spaces using various performance indicators and perform a detailed comparative study of the obtained PS and PF. In addition, we examine their effects separately and analyze the contribution of each of them on the approximation of PS and PF. We also evaluate the effects of different weight parameter values on the diversity of solutions in both the decision and objective spaces.

The rest of this paper is organized as follows: In Section 2 the related works are briefly reviewed. Section 3 introduces the proposed algorithms and the novelty of the work with more detail. In Section 4, the setting of the experiments will be explained. The results of the experiments and analysis are provided in Section 5. Finally the conclusion and the future works will be presented in Section 6.

2 Related Work

In recent years, there has been an increasing amount of literature focusing on finding multimodal solutions for multi-objective optimization problems.[5–7] Some of these works aim to get a better approximation of the PS by increasing the diversity of solutions in decision space. However, this might not provide a better convergence to the PS.[2]

The *Omni-optimizer Algorithm*[8] applies a crowding distance approach to the solutions in the decision space to preserve more solutions in decision space than the objective space. In this algorithm, the final crowding distance value for each solution is assigned based on the comparison of the average crowding distance in both the decision and objective space: If the crowding distance value of each solution either in decision or objective space becomes larger than or equal to the average value, the maximum crowding distance is selected, otherwise the minimum of these two values is taken as the final crowding distance value. The provided modification was applied to the well-known *Non-dominated Sorting Genetic Algorithm (NSGA-II)*.[1]

Zhou et al.[9] proposed a model to make better approximation of both the decision and objective space simultaneously. The population is classified into sub-populations in objective space, and the model increases the diversity of solutions in decision space by evaluation of diversity of the PS in each sup-population. The obtained solutions show better convergence to the PS and PF for the MOP in comparison with the Omni-optimizer algorithm.

Liang et al.[2] presented *DN-NSGA-II* which incorporates a niching algorithm in the decision space. It contains two modifications of the original NSGA-II: The crowding distance method is performed in the decision space instead of objective space, and it creates a mating pool of solutions with a niching technique. The resulting algorithm was able to cover more solutions in the PS than the original NSGA-II algorithm.

Yue et al.[6] proposed *Multi-objective Particle Swarm Optimization using Ring topology by applying Special Crowding Distance* (MO-Ring-PSO-SCD). In this work, a ring topology is used to capture more optimal solutions in the decision space by making robust niches, and a special crowding distance method assists to preserve solutions in the PS. The results of this algorithm show significant improvement compared to NSGA-II, DN-NSGA-II and Omni-optimizer in terms of approximation of the PS in decision space.

In a recent work, Liang et al.[5] adopt the concept of a mutation bound process, which gives a second opportunity to perform in-bound mutation if the mutated solutions lie outside the boundaries of the decision space. They also use both non-dominated sorting in objective space and the crowding distance technique in decision space. Their proposed method, called *Multimodal Multi-Objective Differential Evolution* (MMODE), was applied to a differential evolution algorithm. The results of this algorithm show improvement in terms of diversity of solutions in both decision and objective space.

In the work presented by Liang et al. in 2018,[7] they proposed an improved version of SMPSO algorithm[10] with the ability of creating neighborhoods in decision space. Furthermore, they designed a special version of crowding distance on both decision and objective space to keep the obtained optimal solutions. The experimental results show that the mentioned algorithm could obtain better approximations of the PS than other state-of-the-art algorithms like the MO-Ring-PSO-SCD.

3 Proposed NSGA-II-WSCD-NBM

In this section, we modify the existing NSGA-II[1] with a weighted sum crowding distance method and a new polynomial mutation operator in the so called NSGA-II-WSCD-NBM algorithm.

3.1 Weighted-Sum Crowding Distance method

The classical crowding distance (CD) approach is typically used in the objective space to improve the diversity of the solutions in the objective space.[1] This approach leads to a better approximation of the PF, but it does not promise to preserve all the solution in the PS. Therefore, similar to the Omni-optimizer algorithm,[8] we adopt the concept of crowding distance in both spaces to obtain the better

Algorithm 1: Weighted Sum Crowding Distance approach.

Input: List S of non-dominated solutions with added Crowding Distance (CD_{obj}) values for each solution in objective space according to NSGA-II[1] algorithm with $s := |S|$,
Number of Objectives: M,
Number of Decision Variables
1 : D
Output: List S with added Weighted Sum Crowding Distance (CD_{WS}) values for each solution
2 **for** $i \in \{1,..,D\}$ **do**
3 $x_{i,max}$ = maximum of values for i-th decision variable in S
4 $x_{i,min}$ = minimum of values for i-th decision variable in S
5 **end**
6 **for** $j \in \{1,..,s\}$ **do**
7 $S[j].CD_{dec} = 0$ //initialize CD_{dec} of j-th solution in S
8 $S[j].CD_{WS} = 0$ //initialize CD_{WS} of j-th solution in S
9 **end**
10 **for** $i \in \{1,..,D\}$ **do**
11 S' = sort S ascending based on i-th decision variable
12 $S'[1].CD_{dec} += 2 \cdot \frac{|S'[j+1].x_i - S'[j].x_i|}{|x_{i,max} - x_{i,min}|}$
13 $S'[s].CD_{dec} += 2 \cdot \frac{|S'[j].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
14 **for** $j \in \{2,..,s-1\}$ **do**
15 $S'[j].CD_{dec} += \frac{|S'[j+1].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
16 **end**
17 **end**
18 **for** $j \in \{1,..,s\}$ **do**
19 $S[j].CD.obj = norm(S[j].CD_{obj})$ //normalize CD_{obj} of j-th solution using max and min values of CD_{obj} in S
20 $S[j].CD.dec = norm(S[j].CD_{dec})$ //normalize CD_{dec} of j-th solution using max and min values of CD_{dec} in S
21 $S[j].CD_{WS} = w_1 \cdot S[j].CD_{dec} + w_2 \cdot S[j].CD_{obj}$;
22 **end**
23 **return** S

approximation of the PS and PF. Our approach is called Weighted Sum Crowding Distance *WSCD* as it is calculated as the weighted sum of the crowding distances in objective and decision space. The WSCD method is shown in Algorithm 1.

In WSCD, the calculation of crowding distance in the objective space is similar to the proposed CD calculations in NSGA-II. The extreme solutions in the objective space are assigned a large CD values (infinity).The CD values for the rest of the solutions are calculated by the sum of the normalized distances between the left-side and the right-side neighbors in the objective space.[1]

In the proposed WSCD approach, first the calculation of the crowding distance in decision space is adopted from the Omni-optimizer from the literature (Lines 1 to 16). The maximum and minimum values for all solutions are calculated (Lines 2 and 3). The crowding distance values in decision space and the WSCD values for all solutions are first set to zero (Lines 6 and 7). Then the solutions are sorted based on the decision variable values for each variable (Line 10). The crowding distance value for the boundary solutions are calculated from the normalized distance values between the solution and its adjacent neighbors (Lines 11 and 12). The crowding distance values for the rest of the solutions are calculated by normalizing the distances between the left-side and right side neighbors for the solutions in decision space (Line 14). The novelty of our work is as follows: The crowding distance values in decision and objective spaces are normalized in order to make the scores of crowding distance values comparable for different dimensions in decision and objective space (Lines 18 and 19). Given the importance of having a good diversity of solutions in both decision and objective space, we allocate a final weighted sum crowding distance value based on the assigned weights w_1 and w_2 for the crowding distance in the decision and the objective space (Line 21).

3.2 Neighborhood Polynomial Mutation

In multi-objective evolutionary algorithms, the Polynomial mutation is shown to be one of the effective operators.[11] It was originally proposed by Deb and Goyal.[12] In this section, we propose a modification to this operator inspired by the concept of neighborhood mutation by Qu et al.[13] to make it more applicable on multimodal optimization problems. The neighborhood polynomial mutation is presented in Algorithm 2. In this algorithm, a set of neighbors is computed for each solution, and the mutation operator is applied to each of them.

In Algorithm 2, at first the Euclidean distances between all solutions in the decision space are computed (Line 3). The neighborhood of each solution is composed out of the individual itself and its K nearest neighbors in terms of computed distances (Line 7). Afterwards, for each individual in the population, a Polynomial mutation is used to mutate the individual and its neighbors (Lines 9 to 26). The mutated offsprings are returned (Line 27). Using this mutation operator implies that a neighboring solution which appears in the neighborhood of many solutions, has the chance to be mutated more often than other solutions. In that way, the solutions which are located in crowded areas in the search space have a higher chance of being mutated. As a result, this might lead to a better exploration in the decision space.

Algorithm 2: Neighborhood Polynomial Mutation.

Input: List O of offspring of solutions of current generation with $o := |O|$,
Neighborhood Size=K
Probability of Mutation=p_m,
Distribution Index=η_m
Upper and lower bounds x_k^u and x_k^l for each variable k
Output: Mutated Individuals O

for $i \in \{1,..,o\}$ **do**
  for $j \in \{1,..,o\}$ **do**
    $Euc(i,j) = \|O[i].\vec{x} - O[j].\vec{x}\|_2$ //calculate Euclidean distances between solutions
  end
end
for $i \in \{1,..,o\}$ **do**
  $N(i)$ = list of indices of $K+1$ smallest values in $Euc(i)$ //Set the neighborhood of each solution i as itself and its K nearest neighbors
end
for $i \in \{1,..,o\}$ **do**
  for $j \in N(i)$ **do**
    for $k \in \{1,..,D\}$ **do**
      $b = U(0,1)$
      if $b \leq p_m$ **then**
        $\delta_1 = \frac{O[j].x_k - x_k^l}{x_k^u - x_k^l}$
        $\delta_2 = \frac{x_k^u - O[j].x_k}{x_k^u - x_k^l}$
        $b = U(0,1)$
        if $b \leq 1/2$ **then**
          $\delta_q = [(2b) + (1-2b)(1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1$
        else
          $\delta_q = [1 - (2(1-b)) + 2(b-0.5)(1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}$
        end
        $O[j].x_k += \delta_q.(x_k^u - x_k^l)$
      end
    end
  end
end
return O

4 Experimental Setting

In order to evaluate the effectiveness of the modifications, we considered various versions of the proposed algorithm. The NSGA-II with the Neighbourhood Mutation operator (NSGA-II-NBM), the NSGA-II with the Weighted Sum Crowding Distance (NSGA-II-WSCD), and NSGA-II with both of the modifications (NSGA-II-WSCD-NBM). The results are compared with the results of the state-of-the-art multimodal optimization algorithm Mo-Ring-PSO-SCD.[6] We additionally compare the results with NSGA-II[1] as the baseline.

The median and the interquartile range (IQR) of all the experimental results are calculated over 31 independent runs for a maximum of 10,000 function evaluations. The population size is set to 100 for all the experiments.

The parameters of NSGA-II are set to be similar as in the literature.[1] We set the distribution index of both crossover η_c and mutation η_m to be 20. The probability of crossover is set to $p_c = 1.0$, and the probability of mutation is set to $p_m = 1/D$, where D is the number of decision variables.

The neighborhood size for the neighborhood mutation in both the NSGA-II-WSCD-NBM and NSGA-II-NBM is set to 20. In both WSCD variations, NSGA-II-WSCD-NBM and NSGA-II-WSCD, the weights are equally divided for crowding distances in decision and objective spaces as $w_1 = 0.5$ and $w_2 = 0.5$. In the Mo-Ring-PSO-SCD, we use the same parameter values as in the literature.[6] Therefore, we set $C_1 = C_2 = 2.05$ and $W = 0.7298$. We used codes provided in Matlab-based PlatEmo[14] framework for the NSGA-II and the codes by the original authors for Mo-Ring-PSO-SCD.[6]

4.1 Test Problems

We take the state-of-the-art test problems for multimodal multi-objective optimization[2,6] to test our proposed algorithms. We use the SSUF1 and SSUF3 test problems[2] and MMF3, MMF4, MMF5,and MMF6 problems.[6] The problems contain different levels of complexity and different numbers of equivalent subsets of the PS to challenge the functionality of the proposed algorithms.

The dimensions of decision and objective spaces are 2 in all of the problems. Since the problems are multimodal, one of the most important features of these test problems is that there are always multiple distinct subsets of the PS in each problem, where each of them covers the PF completely on its own. The related features of the test problems are listed in Table 1.

Table 1: Properties of Test Problems

Test Problems	No. of subsets in the PS	PF Shape
SSUF1	2	concave
SSUF3	2	concave
MMF3	2	concave
MMF4	4	convex
MMF5	4	concave
MMF6	4	concave

4.2 Performance Measures

Since our primary focus lies in decision space, the Inverted Generational Distance in decision space (IGDX)[9] is adopted as a metric to measure the effectiveness of the algorithms. The IGDX performance metric is calculated as the average Euclidean distance between the set of obtained solutions and the PS in decision space. This metric demonstrates the diversity and convergence of obtained solutions in relation to the Pareto-optimal solutions set. A lower IGDX value indicates a better performance. Let P^* be a sample of the PS of the problem, and R a set of obtained solutions in decision space by an algorithm, the IGDX indicator is formulated as:

$$IGDX(P^*,R) = \frac{\sum_{v \in P^*} \|R - v\|_2}{|P^*|} \tag{2}$$

Where $\|R - v\|_2$ is the minimum Euclidean distance between the sampled point v and any point in R.

Additionally, in order to compare the performance of the algorithms with each other in the objective space, we use the Inverted Generational distance (IGD):[15,16]

$$IGD(P^*,R) = \frac{\sum_{v \in P^*} \left\| \vec{f}(R) - \vec{f}(v) \right\|_2}{|P^*|} \tag{3}$$

This indicator is formulated in the same way as the IGDX. The IGD value is calculated, with the difference that the distances are calculated in the objective space using a sample of the PF (which can be obtained by evaluating the PS as $\vec{f}(P^*)$) and $\vec{f}(R)$ accordingly.

5 Analysis of Results

The experimental results (median and IQR) for the comparison of the used algorithms concerning IGDX and IGD indicators are shown in Tables 2 and 3 respectively. In order to test the statistical significance, we take the Mann-Whitney U statistical test with respect to the best algorithm on each test problem. That is, we test for each algorithm the hypothesis that the performance of the algorithm and the performance of the best algorithm on this problem have equal medians. A difference between the two results is regarded as significant for values of $p < 0.01$. The best values are highlighted in bold and significance compared to the best algorithm is shown by an asterisk (*) in the respective columns.

From the analysis of Table 2 regarding the comparison of IGDX values, it can be concluded that the NSGA-II-WSCD-NBM algorithm outperforms the NSGA-II-NBM in four out of six problems. It also shows its significant superiority for all the test problems compared with the results of the other algorithms. This means the proposed algorithm provides better approximations of PS in terms of the both diversity and convergence of the obtained solutions.

To analyze the performance in the objective space, Table 3 shows the IGD values for the different algorithms. As can be observed from the results, NSGA-II-NBM obtains a better IGD value than the others, while both the NSGA-II and NSGA-II-WSCD-NBM algorithms gained IGD values similar to each other.

We can further observe that the proposed methods significantly outperform the original NSGA-II and the state-of-the-art Mo-Ring-PSO-SCD. In terms of IGDX, the proposed NSGA-II-WSCD-NBM performs significantly better than both algorithms from the literature on all of the six test problems. In the objective space, measured by the IGD indicator, NSGA-II-NBM outperforms the state-of-the-art in all of the used benchmarks, and the original NSGA-II on all but one test problem.

According to the analysis of the results, the WSCD variants lead to preserving distinct solutions with the same objective function values. Therefore the NSGA-II-WSCD shows improvement compared to NSGA-II in terms of the decision-space related metric. In addition, introducing neighborhood mutation helps to discover more Pareto-optimal solutions during the search by increasing the diversity of solutions.

In order to better understand the similarity between the obtained solutions in both decision and objective spaces, we present the obtained solutions for the NSGA-II-WSCD-NBM, NSGA-II-WSCD, NSGA-II-NBM and Mo-Ring-PSO-SCD in Figures 1 and 2. The figures show the runs which achieved the median IGDX indicator for each of the algorithms.

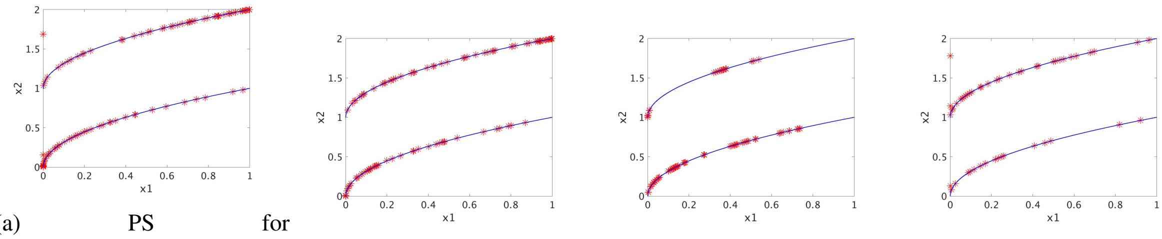
As an example, in Figure 1, we illustrate the obtained solutions in the decision space for the SSUF3 problem of the algorithms. The same is shown for the objective space. We can observe that all algorithms obtain an evenly spread solution set along the PF in the objective space. However, when we look at the decision space we see differences. As can be seen from the Figures 1 and2, the obtained solutions in the decision space for NSGA-II-WSCD-NBM are evenly distributed along the PS while covering more points in each of the subsets of the PS. This is because both the NBM and WSCD methods could help the algorithm to locate and maintain the captured optimal solutions in decision space in each generation.

In NSGA-II-NBM, the obtained solutions are mostly located in one of the subsets. This means that this algorithm could not preserve the solutions in different subsets, since the crowding distance is only used in objective space. While the solutions in decision space are distributed in all the equivalent subsets of the PS in the NSGA-II-WSCD algorithm, we still lack an even spread along these subsets (Figure 2c).

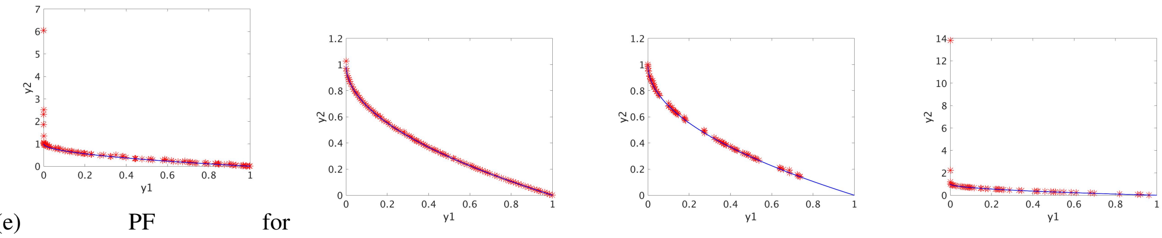
Altogether, we conclude that NSGA-II-WSCD which uses the crowding distance in the decision space helps to maintain most of the so far found solutions. However, due to a lack of neighborhood mutation process, it could not find all the solutions of the PS.

The results of the Mo-Ring-NSGA-II shown in the Figures 1 and 2, also reveal that the PS could not be fully covered by the algorithm and the solutions are not evenly distributed along the PS.

In further experiments, we will investigate the impact of the weights in the WSCD variants. Our preliminary studies show that increasing the weight value in each of the spaces improves the distribution of solutions in the corresponding space, while deteriorating the distribution of solution in the other space.

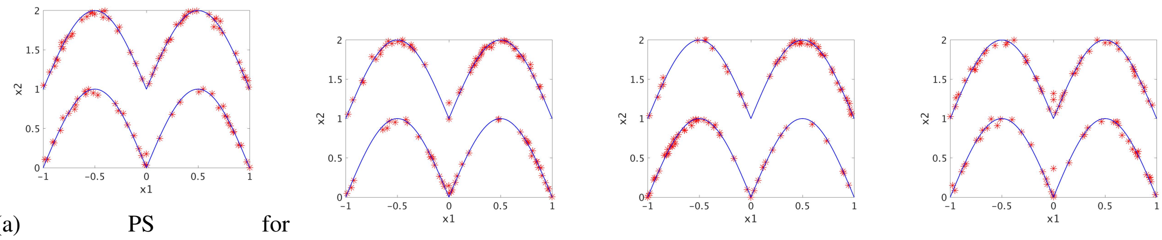


(a) PS for NSGA-II-WSCD-NBM (b) PS for NSGA-II-NBM (c) PS for NSGA-II-WSCD (d) PS for Mo-Ring-PSO-SCD

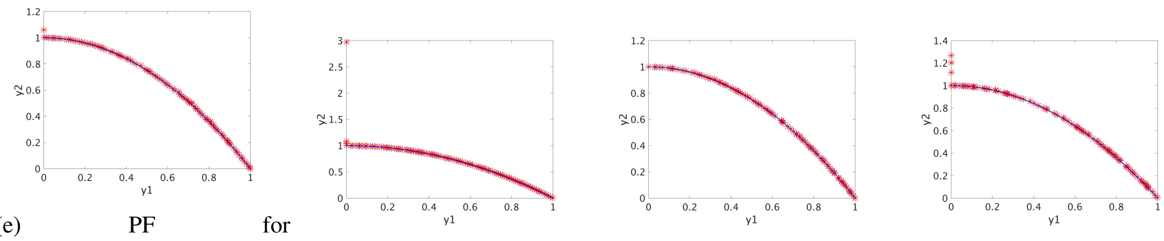


(e) PF for NSGA-II-WSCD-NBM (f) PF for NSGA-II-NBM (g) PF for NSGA-II-WSCD (h) PF for Mo-Ring-PSO-SCD

Figure 1: Obtained solutions in decision and objective space for SSUF3 problem



(a) PS for NSGA-II-WSCD-NBM (b) PS for NSGA-II-NBM (c) PS for NSGA-II-WSCD (d) PS for Mo-Ring-PSO-SCD



(e) PF for NSGA-II-WSCD-NBM (f) PF for NSGA-II-NBM (g) PF for NSGA-II-WSCD (h) PF for Mo-Ring-PSO-SCD

Figure 2: Obtained solutions in decision and objective space for MMF4 problem

6 Conclusion

The purpose of the current study is to propose two mechanisms for acquiring better approximations of the PS in multimodal multi-objective problems. These two mechanisms are (1) the WSCD method which combines crowding distance in both objective and decision space, and (2) a neighborhood Polynomial mutation. The two proposed operators were included in different combinations into the existing NSGA-II algorithm. In order to examine the performance of the presented combination of operators, we compare the algorithms with the original NSGA-II algorithm as well as the state-of-the-art multimodal algorithm from the literature (Mo-Ring-PSO-SCD) on six different test problems. The IGDX and IGD performance indicators are used to compare the performance of the algorithms in decision and objective spaces, respectively. The results show significant differences between the proposed variants of NSGA-II-WSCD-NBM, NSGA-II-WSCD, NSGA-II-NBM algorithms and the existing algorithms in terms of approximations of the PS and PF. The proposed algorithm NSGA-II-WSCD-NBM is able to outperform the state-of-the-art Mo-Ring-PSO-SCD and the standard NSGA-II on all of the test problems in terms of approximating the PS, while at the same time obtaining comparable IGD values.

For future work, we want to compare the proposed NSGA-II-WSCD-NBM with other state-of-the-art multimodal algorithms like those recently proposed in.[5,7,17] In addition, we will study the influence of the weights in the WSCD variants.

Table 2: IGDX values of different algorithms. An asterisk (*) indicates statistical significance compared to the respective best algorithm

	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
SSUF1	**0.063212 (1.817E-3)**	0.075516 (9.316E-3) *	0.07923 (7.412E-3) *	0.07242 (7.267E-3) *	0.109946 (0.017274) *
SSUF3	**0.01688 (2.885E-3)**	0.01771 (3.956E-3)	0.089485 (7.2725E-2) *	0.038545 (1.18391E-2) *	0.083872 (4.0917E-2) *
MMF3	**0.014856 (1.309E-3)**	0.015017 (2.318E-3)	0.0.05839 (3.49894E-2) *	0.03063 (1.0334E-3) *	0.072747 (2.8699E-2) *
MMF4	**0.041635 (4.949E-3)**	0.058955 (1.0412E-2) *	0.057928 (1.0978E-2) *	0.045201 (5.714E-3) *	0.111808 (31111E-1) *
MMF5	**0.113945 (4.388E-3)**	0.129521 (1.7379E-2) *	0.144731 (1.1124E-2) *	0.125208 (9.125E-3) *	0.184868 (6.653E-2) *
MMF6	**0.099212 (6.021E-3)**	0.108116 (1.054E-2) *	0.124064 (1.2611E-2) *	0.106895 (9.113E-3) *	0.212675 (7.075113E-2) *

Table 3: IGD values of different algorithms. An asterisk (*) indicates statistical significance compared to the respective best algorithm

	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
SSUF1	5.528E-3 (6.2E-4) *	**4.6E-3 (1.16E-4)**	5.441E-3 (3.22E-4) *	6.459 E-3 (8.19E-4) *	5.35E-3 (2.72E-4) *
SSUF3	1.4495E-2 (2.516E-3)	**1.4452E-2 (2.497E-3)**	1.6955E-2 (1.4602E-2)*	1.883E-2 (5.794E-3) *	1.6727E-2 (9.216E-3)
MMF3	1.2298 E-2 (2.123E-3) *	**1.098E-2 (2.007E-3)**	1.527E-2 (1.3291E-2) *	1.6865E-2 (4.86E-3) *	1.4027E-2 (9.412E-3) *
MMF4	5.347E-3 (7.62E-4) *	**4.762E-2 (2.37E-4)**	5.425E-3 (2.52E-4) *	7.047E-3 (9.98E-4) *	5.165E-3 (2.65E-4) *
MMF5	5.369E-3 (3.94E-4)*	**4.595E-3 (1.73E-4)**	5.588E-3 (3.24E-4) *	6.544E-3 (5.4E-4) *	5.403E-3 (3.2E-4) *
MMF6	5.433E-3 (4.57E-4)	**4.589 E-3 (2.01E-4)**	5.489E-3 (2.83E-4) *	6.44E-3 (7.58E-4) *	5.185E-3 (2.13E-4) *

References

[1] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation* **6**(2), 182–197 (2002).

[2] Liang, J., Yue, C., and Qu, B. Multimodal multi-objective optimization: a preliminary study. In *Evolutionary Computation (CEC)*, 2454–2461 (IEEE, Vancouver, BC, Canada, 2016).

[3] Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation* **8**(3), 256–279 (2004).

[4] Javadi, M., Zille, H., and Mostaghim, S. Modified crowding distance and mutation for multimodalmulti-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO)* (ACM, Prague,czech republic, 2 pages. To appear, 2019).

[5] Liang, J., Xu, W., Yue, C., Yu, K., Song, H., Crisalle, O. D., and Qu, B. Multimodal multiobjective optimization with differential evolution. *Swarm and Evolutionary Computation* **44**, 1028–1059 (2019).

[6] Yue, C., Qu, B., and Liang, J. A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. *IEEE Transactions on Evolutionary Computation* **22**(5), 805–817 (2018).

[7] Liang, J., Guo, Q., Yue, C., Qu, B., and Yu, K. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In *International Conference on Swarm Intelligence*, 550–560 (Springer, Shanghai, China, 2018).

[8] Deb, K. and Tiwari, S. Omni-optimizer: A procedure for single and multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, 47–61 (Springer, Guanajuato, Mexico, 2005).

[9] Zhou, A., Zhang, Q., and Jin, Y. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE transactions on evolutionary computation* **13**(5), 1167–1189 (2009).

[10] Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello, C. C., Luna, F., and Alba, E. Smpso: A new pso-based metaheuristic for multi-objective optimization. In *Computational intelligence in multi-criteria decision-making*, 66–73 (IEEE, Nashville, TN, USA, 2009).

[11] Hamdan, M. The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study. In *International Conference on Electronics Computer Technology* (IEEE, Kanyakumari,India, 2012).

[12] Deb, K. and Goyal, M. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics* **26**, 30–45 (1996).

[13] Qu, B.-Y., Suganthan, P. N., and Liang, J.-J. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE transactions on evolutionary computation* **16**(5), 601–614 (2012).

[14] Tian, Y., Cheng, R., Zhang, X., and Jin, Y. Platemo: A matlab platform for evolutionary

multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine* **12**(4), 73–87 (2017).

[15] Reyes-Sierra, M. and Coello, C. A. C. A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In *Evolutionary Computation*, volume 1, 65–72 (IEEE, Edinburgh, Scotland, UK, 2005).

[16] Zhang, Q., Zhou, A., and Jin, Y. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* **12**(1), 41–63 (2008).

[17] Tanabe, R. and Ishibuchi, H. A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, 249–261 (Springer, Coimbra, Portugal, 2018).