MM-NAEMO : Multimodal Neighborhood-sensitive Archived Evolutionary Many-objective Optimization Algorithm

Kalyanbrata Maity
Department of
Computer Science and Engineering
Indian Institute of Technology Patna
Patna, India
maitykalyanbrata94@gmail.com

Raunak Sengupta
Department of
Electrical Engineering
Indian Institute of Technology Patna
Patna, India
raunaksengupta@gmail.com

Sriparna Saha
Department of
Computer Science and Engineering
Indian Institute of Technology Patna
Patna, India
sriparna.saha@gmail.com

***Abstract*—In certain multi-objective optimization problems, it may happen that there are two or more different Pareto optimal sets corresponding to the exact same Pareto Front. These are known as multimodal multi-objective optimization problems (MMOPs). Algorithms which are able to provide all the Pareto sets can give the flexibility to choose solutions and therefore a possible improvement in the performance. In this work, we build upon the original framework of NAEMO (Neighborhood-sensitive Archived Evolutionary Many-objective Optimization Algorithm) and present Multimodal NAEMO (MM-NAEMO). In MM-NAEMO, the mutation occurs between the points which are local neighbors and each reference line always keeps at least two or more candidate solutions associated with it. Clustering is used to maintain the diversity amongst the population associated with each reference line, thus giving multiple Pareto sets as output. The algorithm is tested on the Test problems of MMO test suite of CEC2019. The experimental results suggest that the proposed algorithm effectively find majority of the Pareto sets without hampering the Pareto fronts for most of the problems.**

***Index Terms*—multimodal multi-objective problems, Pareto dominance, Reference vectors, Mutation switching, Gaussian Mixture Model (GMM).**

I. Introduction

Multi-objective optimization problems may have more than one conflicting objectives which have to be simultaneously optimized. These types of problems can be stated as follows :

$$\begin{aligned} \min \quad & \mathbf{z} = F_M(\mathbf{y}) = [f_1(\mathbf{y}), f_2(\mathbf{y}), \cdots, f_M(\mathbf{y})] \\ s.t: \quad & g_i(\mathbf{y}) \leq 0, i = 1, 2, \cdots, p \\ & h_j(\mathbf{y}) = 0, j = 1, 2, \cdots, q \end{aligned} \tag{1}$$

where $\mathbf{y} = (y_1, y_2, \cdots, y_N) \in \mathbf{Y} \subset \boldsymbol{R}^N$ is a decision vector in N-dimensional space, $\mathbf{z} = (z_1, z_2, \cdots, z_M) \in \mathbf{Z} \subset \boldsymbol{R}^M$ is an objective vector in M-dimensional space and $F_M(\mathbf{y})$ denotes M number of functions mapped from $\mathbf{Y}$ to $\mathbf{Z}$, where each element in $F_M(\mathbf{y})$ is a single-objective problem and M is the number of objectives. $g_i(\mathbf{y}) \leq 0, i = 1, 2, \cdots, p$ denote p inequality constraints and $h_j(\mathbf{y}) = 0, j = 1, 2, \cdots, q$ denote q equality constraints where each element in $\mathbf{y}$ is a decision variable and N is the total number of decision variables.

In case of multi-objective optimization problems, it is really difficult to find a single best solution but we can compare different solutions by Pareto dominance. For minimization of a multi-objective optimization problem, suppose there are two feasible solutions y_1 and y_2 then it is said that y_2 is dominated by y_1 or $(y_1 \prec y_2)$ only if, $f_i(y_1) \leq f_i(y_2)$, for $i = 1, 2, \cdots, M$ and $f_j(y_1) < f_j(y_2)$ for at least one j, where $j \in \{1, 2, \cdots, M\}$. A solution is said to be non-dominated if no other solution dominates it and the set consisting of all solutions that are non-dominated to each other is known as Pareto-optimal set (PS) whereas the values of Pareto-optimal set in the objective space form the Pareto front (PF).

There are several multi-objective optimization problems which have two or more different Pareto sets with exact same Pareto front in the objective space, these particular types of problems are said to be multimodal multi-objective optimization problems (MMOPs). For this type of problems, finding one Pareto set is comparatively easy whereas finding all the Pareto sets is challenging. Also, providing all the possible solutions to a decision maker gives the flexibility to choose solutions and may obtain improved performances.

Niching techniques are used very frequently for solving multimodal problems. Various niching methods like fitness sharing [1], [2], crowding [3], clearing [4], speciation [5] have been proposed and shown to work well. Liang *et al.* proposed DN-NSGA II [6] which is a modification of NSGA II to solve MMOPs. It uses crowding distance in decision space as a niching technique for mating selection. Yue *et al.* proposed MO_Ring_PSO_SCD [7] which is a Particle swarm optimization algorithm using ring topology. This algorithm introduces a new concept of special crowding distance which is adopted both in decision and objective spaces.

In this work, we propose MM-NAEMO (Multimodal Neighborhood-sensitive Archived Evolutionary Many-objective Optimization Algorithm) which is an extension of NAEMO (Neighborhood-sensitive Archived Evolutionary Many-objective Optimization Algorithm) [8] for solving multimodal multi-objective optimization problems. NAEMO is a reference line based many-objective optimization algorithm which incorporates the neighborhood property of many-objective problems. During the mutation, NAEMO selects candidate solutions from local neighborhood and it always continues to keep at least one candidate to each reference line. This property of NAEMO monotonically improves the diversity which has been stated and proven in [8]. NAEMO adopts a periodic filtering scheme on the population to enhance convergence and diversity. When the population size goes beyond a specified value, a diversity-based filtering is carried out on the population which removes points with a larger PBI (Penalty-based Boundary Intersection) value [9] from a reference line having the highest number of associated candidates. Also, a convergence-based filtering is carried out when a newly generated candidate solution dominates other solutions in the population and it removes all dominated solutions. The use of PBI function along with selecting non-dominated solutions increases selection pressure and tries to push the solutions towards optimal values. NAEMO also adopts an improved mutation strategy which adapts hyper-parameter for mutation along with a concept of switching mutation operations depending on the probability values. This concept can combine the advantages of various mutation operations, proven in [8].

The robust performance and strong theoretical background of NAEMO motivated us to modify it to MM-NAEMO. As NAEMO was not designed to identify and maintain points for multiple Pareto sets, we give special attention to solve this issue in MM-NAEMO. However, the multiple Pareto sets associated with multimodal problems can be viewed as separate clusters of points all corresponding to the same Pareto front. Thus, while searching for solutions, the algorithm must keep searching for clusters. Let us assume that the number of Pareto optimal sets is 2. If we draw multiple reference lines in the objective space, each reference line will have 2 associated points in the decision space corresponding to the associated optimal point in the objective space. Following this logic, we extend the filtering functions of NAEMO by incorporating Gaussian Mixture Models to perform the clustering. Unlike in the original NAEMO where we maintained at least one solution associated with each reference line, in MM-NAEMO we maintain at least 2 solutions for each of the reference line.

The key differences between NAEMO and MM-NAEMO are enumerated below:

- In convergence-based filtering of NAEMO we remove all the points which are dominated by a newly selected candidate vector from the archive whereas in MM-NAEMO, a Gaussian Mixture Model based clustering is performed on each sub-archive and two clusters are formed for each sub-archive. After that, we continue to remove dominated points from both clusters while maintaining at least one point in each cluster.
- In NAEMO, for divergence-based filtering, we remove points with larger PBI values from a reference line which has highest number of points associated with it. But in MM-NAEMO, we perform clustering on each sub-archive and form two clusters. Then we remove points from each cluster with larger PBI values from a reference line which has highest number of points.
- The other key difference is that while we continue to maintain at least one point associated with each reference line in NAEMO, we continue to maintain at least two points in MM-NAEMO.

There are 22 test problems in MMO test suite of CEC 2019 which are tested on MM-NAEMO. Out of which 18 problems (MMF1-13, SYM-PART simple and rotated, MMF1_e and MMF1_z) have 2 objectives and for the rest 4 problems (MMF14, 14_a, 15 and 15_a), the number of objectives is scalable. In case of Omni-test, the number of decision variables is also scalable. All the 3 objective problems have concave PFs. MMF14 and MMF15 have linear PSs whereas MMF14_a and MMF15_a have non-linear PSs. For 2 objectives MMF1-13 including SYM-PART simple and rotated, MMF1_e and MMF1_z have convex PFs except for MMF4 and MMF8 have concave PFs. Also, MMF9-12, SYM-PART simple and rotated, Omni-test have linear PSs and rest 2 objective problems have non-linear PSs. However, The geometry of Pareto sets and Pareto fronts are considered as unknown except the dimensionality of the problems during the design of the algorithm.

Here in our experiments, the number of decision variables and objectives for MMF14, MMF15, MMF14_a and MMF15_a are taken as 3 and for Omni-test, the number of decision variables is also taken as 3.

MM-NAEMO performs significantly better than MO_Ring_PSO_SCD [7] and DN-NSGAII [6] for almost all the test problems. Especially it outperforms them by obtaining all Pareto sets in decision space which can be observed from the values of 1/PSP and IGDX in Table I. Though MO_Ring_PSO_SCD and DN-NSGAII perform better than MM-NAEMO in terms of 1/HV value but MM-NAEMO shows better IGDF values for most instances which indicates it's effective performance in objective space.

Section II describes MM-NAEMO in detail along with pseudo-codes. This algorithm is then evaluated on the MMO test suite of CEC 2019. The results are presented in Section IV.

II. Algorithm Description

NAEMO at its core is based upon the 'Neighborhood Property' which has been stated and proven in [8]. It has also been proven theoretically and experimentally in [8] that the 'Neighborhood Property' improves the diversity monotonically in NAEMO. In this section, we start by explaining the basic framework of the original NAEMO algorithm. Then we move on to how we build MM-NAEMO upon NAEMO by using Gaussian Mixture Models to solve multimodal multi-objective optimization problems.

A. NAEMO Framework

NAEMO begins with initializing a global archive by randomly generating a predefined (L_{soft}) number of points. This archive comprises of several sub-archives. The number of sub-archives is equal to the number of reference lines and each sub-archive stores a population associated with each of the reference lines. To form the sub-archives, an association operation is performed (as described in II-B) for each randomly generated point. The i^{th} sub-archive (sub_arch_i) of the global archive ($arch$) can be shown as in Eq. (2).

$$arch = \{sub\_arch_1, sub\_arch_2, ..., sub\_arch_n\} \tag{2}$$

One single generation is completed by iterating through all the n sub-archives. Suppose $\mathcal{W}$ is the set of all reference lines. Then, $|\mathcal{W}| = n$. We select a point from each sub-archive(sub_arch_i) randomly as the parent vector, ($parent$). If this sub-archive is empty then we select another reference line from a set of k non-empty reference lines which are closest to the i^{th} reference line.

The intuition behind selecting a point from the local neighbor is that if we perform mutation of points from local neighbor of the empty sub-archive it will have a higher probability to generate a new point which will be associated with that empty reference line. After selecting the parent vector we apply mutation operator on it to generate a new candidate solution. However, to follow the 'Neighborhood Property' we constrain the mutation operation only to the k closest local neighbors who are non-empty, i.e., if nbr_i denotes the set of neighboring reference lines corresponding to the reference line, i, then the other parent vector comes from only sub_arch_j, such that $j \in nbr_i$.

When a new candidate solution or $child$ is generated it is selected for addition to a sub-archive only if the $child$ is not dominated by the $parent$. If it is selected, the $child$ is added to a sub-archive by performing an association operation on it. After adding it to a sub-archive, it checks for the points in the sub-archive which are dominated by the $child$, if dominated points are found they are removed from the sub-archive through convergence filtering. If the total size of the population exceeds L_{soft}, diversity filtering is used to reduce the size of the population to a hard limit L_{hard}.

B. Associating a Point to a Reference Line

To associate a point with the closest reference line, the association operation is performed. We find the distances between a point with every reference lines and then we choose the reference line having minimum distance as associated reference line. The formula (3) for calculating distance is as follows:

$$\begin{gathered} d_2(x, w_i) = \left\| x - \frac{x \cdot w_i}{\|w_i\|^2} \cdot w_i \right\| \\ x \in sub\_arch_i, \text{ when } i = \underset{w_i \in \mathcal{W}}{\arg\min}\, d_2(x, w_i) \end{gathered} \tag{3}$$

where d_2 is the distance between a point (x) and i^{th} reference line, $w_i \in W$

Algorithm 1 Main Framework of MM-NAEMO

Input: tot_itr: Total number of generations; n: Total number of reference lines; L_{soft} and L_{hard}: soft and hard limits for population in the archive; $flag1$, $flag2$, mut_prob and η_m: parameters for mutation-switching.
Output: $\mathcal{A}$: final population of archive

```
Initialize set of reference lines W using the method of Das and Dennis [10]
Initialize population of archive arch of size L_soft randomly.
Association ∀x ∈ arch and w_i ∈ W and form sub_arch_i using Eq. (3).
for itr = 1 to tot_itr do
    S_ηc = φ
    S_F = φ
    S_CR = φ
    for j = 1 to n do
        index = j
        if sub_arch_index = φ then
            I_rand ← random(1, |nbr_j|)
            index = I_rand
        end if
        par_ind ← random(1, |sub_arch_index|)
        parent ← sub_arch_index[par_ind]
        η_c = Gaussian(μ_ηc, 5)
        F = Gaussian(μ_F, 0.1)
        CR = Gaussian(μ_CR, 0.1)
        child = Perform Mutation on parent
        if child is not dominated by parent then
            Associate child to a line (Eq. (3))
            sub_arch_line = sub_arch_line ∪ child
            Modified Convergence-based filtering (Algorithm 2)
            if |arch| > L_soft then
                Modified Diversity-based filtering (Algorithm 3)
            end if
            S_ηc = S_ηc ∪ η_c
            S_F = S_F ∪ F
            S_CR = S_CR ∪ CR
        end if
    end for
    μ_ηc = mean(S_ηc)
    μ_F = mean(S_F)
    μ_CR = mean(S_CR)
end for
A = arch
```

C. Original Convergence-based Filtering

In NAEMO, convergence-based filtering is performed once a new child is selected. This filtering method finds all the points from the archive which are dominated by the new child and eliminates them from the archive. But it never eliminates a point if it is the only point corresponding to its associated sub-archive. Maintaining at least one point for each reference line increases the probability for NAEMO to get points in difficult regions for some problems.

D. Original Diversity-based Filtering

As an open size to a population will need a large amount of computations, we perform diversity-based filtering on the total population when the size of the population goes beyond the predefined value L_{soft}. In this filtering, we find the reference line which has the highest number of points associated with it. After that, we find the point which has the highest value of PBI function corresponding to that reference line and eliminate it. We keep performing this process until the total population size becomes equal to the hard limit, L_{hard}. This diversity-based filtering produces selection pressure and pushes the points towards the optimal values.

E. Modified Convergence-based Filtering for multimodal Problems

In MM-NAEMO, the convergence-based filtering scheme has been modified (Algorithm 2), so that it can solve multimodal problems. Unlike NAEMO which eliminates all dominated points by a new child from the population of the archive, MM-NAEMO performs a Gaussian Mixture Model based clustering on sub_arch_i and forms two clusters. After that, it continues to remove the dominated points from each cluster while maintaining a minimum population of size 1 for each cluster in the reference line. The modified convergence-based filtering maintains a minimum of two points for each reference line, thus preserving the diversity as well as points for each of the Pareto sets. It must be noted that in this case, we assume only 2 clusters. However, it can be increased to any number. In cases where we do not know about the number of Pareto optimal sets, we can simply choose a high value of number of clusters.

F. Modified Diversity-based Filtering for multimodal Problems

Similar to the Modified Convergence-based filtering in MM-NAEMO, the concept of clustering is applied to Diversity-based filtering (Algorithm 3). If the total size of the archive overtakes the value L_{soft}, we perform Gaussian Mixture Model based clustering on those sub-archives which have at least two associated points. Then we select the reference line which has the highest number of points associated with it. Then we sort points in each cluster according to their PBI values corresponding with that reference line. Next, we remove points having the highest PBI values from each cluster. We continue removing such points from the archive until the total size of the archive becomes equal to the value of L_{hard}. This filtering strategy using the value of PBI function and clustering increases selection pressure as well as maintains diversity in each reference line. This increases the probability of obtaining different Pareto sets.

The main framework for MM-NAEMO is described in Algorithm 1.

Algorithm 2 Modified Convergence-based Filtering

Input: $arch$: unfiltered archive; sub_arch_i: i^{th} sub-archive; n: number of reference lines or number of partitions of $arch$; $child$: newly added point
Output: $arch$: filtered archive

```
for i = 1 to n do
    l ← Set of points in sub_arch_i dominated by child
    if |l| ≠ φ then
        if |sub_arch_i| > 2 then
            clusters ←GMM(sub_arch_i, clusters_no=2)
            n_0← no of points in cluster 1
            n_1← no of points in cluster 2
            n=[n_0, n_1]
            for k=1 to |l| do
                ind=l[|l| − k − 1]
                if n[clusters[ind]]>1 then
                    {sub_arch_i}←{sub_arch_i}−
                        sub_arch_i[ind]
                    n[clusters[ind]]=n[clusters[ind]] − 1
                end if
            end for
        end if
    end if
end for
```

III. Experimental Settings

In this section, we give a brief description of the comparing algorithms, performance indicators and discuss the experimental settings for MM-NAEMO and the comparing algorithms.

A. Comparing Methods

In our experiment, we have selected two state of the art algorithms to evaluate the performance of MM-NAEMO.

The first algorithm is MO_Ring_PSO_SCD [7], which is a particle swarm optimization algorithm. It uses an index-based ring topology method to form stable niches which help to find multiple Pareto optimal solutions. It also adopts a new concept of special crowding distance as a density metric which is employed in both decision and objective spaces to maintain the diversity.

The second algorithm is DN-NSGA II [6]. This algorithm is a modification of NSGA II to find multiple PSs. This algorithm first sorts all the solutions using a non-dominated sorting method and then a decision space based niching (crowding distance) is used for sorting the solutions in the same front. As a result this algorithm always gives preference to the solutions which are non-dominated and come from a less crowded region in the decision space.

B. Performance Indicators

Four performance indicators, reciprocal of Pareto Sets Proximity (1/ PSP) [7], Inverted General Distance (IGD [11]) in Decision space (IGDX [12]), the reciprocal of Hypervolume (1/HV) [13] and IGD in objective space IGDF [12] are employed for comparing the performances between the

Algorithm 3 Modified Diversity-based Filtering

Input: $arch$: unfiltered archive; sub_arch_i: i^{th} sub-archive; n: number of reference lines or number of partitions of $arch$; pop_arr: array of length n to store population sizes of each sub-archive; L_{hard}: minimum size of archive
Output: : $arch$: filtered archive
$l\_sum = 0$
$pop\_arr \leftarrow \left[0, \overset{n}{\cdots}, 0\right]$
for $i = 1$ to n **do**
  if $|sub\_arch_i| > 2$ **then**
    clusters $\leftarrow GMM(sub\_arch_i, clusters\_no = 2)$
  end if
  $pop\_arr[i] \leftarrow |sub\_arch_i|$
end for
$l\_sum = sum(pop\_arr)$
while $l\_sum > L_{hard}$ **do**
  $ind \leftarrow$ Index of maximum value from pop_arr
  $C1 = \phi, C2 = \phi$
  for $j = 1$ to $|sub\_arch_{ind}|$ **do**
    if $|sub\_arch_{ind,j}| \in$ cluster 1 **then**
      $C1 \cup sub\_arch_{ind,j}$
    else
      $C2 \cup sub\_arch_{ind,j}$
    end if
  end for
  $C1 \leftarrow$ Sort $C1$ by PBI value
  $C2 \leftarrow$ Sort $C2$ by PBI value
  if $|C1| > |C2|$ **then**
    Remove last element from $C1$
  else
    Remove last element from $C2$
  end if
  $sub\_arch_{ind} \leftarrow C1 \cup C2$
  $pop\_arr_{ind} = pop\_arr_{ind} - 1$
  $l_{sum} = l_{sum} - 1$
end while

algorithms. The performance in decision space is evaluated by 1/PSP and IGDX whereas 1/HV and IGDF evaluate the performance in objective space. Smaller values of all four indicators are expected which indicate better performance. The Pareto sets Proximity or PSP is used to measure the similarity between the true PSs and the obtained PSs.

Indicators, IGDX and IGDF, are the *Inverted General Distance* in Decision space and Objective space respectively. These two indicators are used to provide the information of convergence as well as the diversity of the obtained solutions in the decision space and objective space respectively.

Hypervolume(HV) [13] is used to measure convergence as well as diversity of the obtained solutions. Given a reference point, we expect a larger value of HV which indicates solutions of better quality.

C. Experimental Setting

Here total population size is kept equal to $100 * n\_var$ and maximal fitness evaluations $(MaxFES)$ is $5000 * n\_var$. Each benchmark problem has been evaluated on three different algorithms. Four performance indicators $1/PSP, 1/HV, IGDX, IGDF$, as discussed in the Section III B are used to evaluate the performance of the algorithms on different test problems.

The parameter settings of MM-NAEMO are: the values of $\mu_{\eta_c}, \mu_F, \mu_{CR}$ are initialized to 30, 0.5, 0.2 respectively, where the parameters η_c, F, CR adapt over generations. The value of η_m is kept 20. The values of $flag1$ and $flag2$ are set as 'True' and 'False' respectively, for all cases since the problems here are multimodal type. The value of θ for calculating PBI function is fixed to 5.0. The size of the neighborhood is taken as 20% of the total no of reference lines. Also, the value of L_{hard} equals to double of the number of reference lines. The test problems for which the number of decision variables is 2, the values of L_{hard} and L_{soft} are taken as 200 and 300 respectively and for the test problems with 3 decision variables, the values of L_{hard} and L_{soft} are 300 and 450 respectively. The number of $divisions$ for 2 and 3 objective cases are taken as 100 and 15 respectively except for MMF13 and Omni-test where it is taken as 150. The required number of reference lines is calculated using the formula given in non-dominated sorting genetic algorithm III (NSGA-III) [14],

$$W = \binom{M + p - 1}{p} \tag{4}$$

where M denotes the number of objectives, p denotes the number of divisions and W is the number of reference lines. For MM-NAEMO, the number of gen is taken as 93 and 104 for the test problems having number of decision variables 2 and 3 respectively except for MMF13 and Omni-test where it is considered as 94, so that the total number of evaluations does not exceed the specified $MaxFES$, which are $10,000$ and $15,000$ for the problems having 2 and 3 decision variables respectively. The number of decision variables for MMF13 and Omni-test is 3 in our experiment.

For the comparing algorithms, we have taken the population size and maximum number of evaluations same as MM-NAEMO as stated in the CEC 2019 MMO test problems suite.

Here each algorithm is executed for 21 times on each test problem and the mean values of 1/PSP, 1/HV, IGDX and IGDF are calculated. In addition, t-test is performed to check whether the performance of a particular algorithm is significantly different than those by other algorithms for each indicator and the null hypothesis is rejected at 0.05 significance level.

IV. Discussion of Results

A. Results on MMO Test problem suite in CEC 2019

The proposed MM-NAEMO has been tested on all the 22 problems from the MMO Test problems suite of CEC 2019. The corresponding results have been presented in Table I and II.

TABLE I: Mean value of 1/PSP & IGDX.

Test Problems	1/PSP					IGDX				
	MM-NAEMO	MO_Ring_PSO_SCD		DN-NSGA-II		MM-NAEMO	MO_Ring_PSO_SCD		DN-NSGA-II	
MMF1	**0.0487**	0.0488	=	0.0958	+	**0.0487**	0.1485	=	0.0940	+
MMF2	**0.0118**	0.0447	+	0.1403	+	**0.0118**	0.0419	+	0.1141	+
MMF3	**0.0139**	0.0299	+	0.1429	+	**0.0139**	0.0280	+	0.1029	+
MMF4	0.0312	**0.0274**	-	0.0857	+	0.0312	**0.0271**	-	0.0852	+
MMF5	0.0872	**0.0864**	-	0.1798	+	0.0871	**0.0857**	=	0.1771	+
MMF6	0.0743	**0.0741**	=	0.1466	+	0.0742	**0.0736**	=	0.1447	+
MMF7	**0.0229**	0.0265	+	0.0537	+	**0.0228**	0.0263	+	0.0526	+
MMF8	0.3950	**0.0681**	-	0.3070	-	0.3527	**0.0675**	-	0.2967	=
MMF9	**0.0051**	0.0079	+	0.0233	+	**0.0051**	0.0079	+	0.0233	+
MMF10	**0.0124**	0.4885	=	1.6604	+	**0.0123**	0.1709	+	0.1496	+
MMF11	**0.0419**	0.4798	+	1.6544	+	**0.0418**	0.2052	+	0.2502	+
MMF12	**0.0050**	0.5956	+	2.2163	+	**0.0050**	0.1912	+	0.2467	+
MMF13	**0.2725**	0.3508	+	0.6233	+	**0.1878**	0.2377	+	0.2877	+
MMF14	**0.0466**	0.0529	+	0.0973	+	**0.0466**	0.0529	+	0.0973	+
MMF15	**0.0528**	0.1555	+	0.2473	+	**0.0528**	0.1555	+	0.2298	+
MMF1_z	**0.0346**	0.0354	=	0.0819	+	**0.0346**	0.0352	=	0.0804	+
MMF1_e	**0.4543**	0.5512	+	1.8701	+	**0.4258**	0.4744	=	1.1916	+
MMF14_a	0.0662	**0.0618**	-	0.1199	+	0.0662	**0.0617**	-	0.1199	+
MMF15_a	**0.0870**	0.1663	+	0.2234	+	**0.0865**	0.1641	+	0.2096	+
SYM-PART simple	**0.1136**	0.1757	+	4.2022	+	**0.1130**	0.1750	+	4.0889	+
SYM-PART rotated	0.9724	**0.3541**	-	6.0170	+	0.8196	**0.3253**	-	3.8438	+
Omni-test	**0.1457**	0.3934	+	1.4287	+	**0.1456**	0.3897	+	1.4042	+

[1] The mean value is calculated over 21 runs for each Test Problem.

In this section, the performance of MM-NAEMO is evaluated by comparing the values of different indicators with the values of MO_Ring_PSO_SCD and DN-NSGAII empirically. Table I and Table II present the mean values of 1/PSP , IGDX and 1/HV, IGDF respectively, obtained by three algorithms on the different test problems. Here, '+' and '-' sign beside values show that MM-NAEMO performs significantly better and worse respectively. '=' sign indicates that there is no statistically significant difference in performance. We also present obtained solution sets of some test problems in decision and objective spaces to investigate their performances visually in Figs. 1-9.

Table I presents a comparison of the Pareto Sets obtained from the algorithms. It can be observed that MM-NAEMO outperforms DN-NSGA-II and MO_Ring_PSO_SCD for majority of the cases. In terms of IGDX, MM-NAEMO is outperformed in only 4 cases out of 22. This shows that MM-NAEMO effectively combines the concept of NAEMO with GMMs and produces a better Pareto Set with higher convergence as well as coverage.

Table II compares the final Pareto fronts obtained by different algorithms. It can be observed that MM-NAEMO outperforms the other 2 algorithms for a majority of the cases with respect to the IGDF value. This indicates that MM-NAEMO produces a well converged as well as diverse Pareto Front.

To further show the effectiveness of MM-NAEMO, we present figures comparing it with DN-NSGA-II and MO_Ring_PSO_SCD in the next sub-section.

B. Visual Comparison of MM-NAEMO with Other State of the Art Algorithms

Here, in this section, we show a visualization of performance of MM-NAEMO through plots of Pareto sets and Pareto fronts obtained for MMF4, MMF9 and MMF14 test problems. We also provide a comparison of MM-NAEMO with DN-NSGAII and MO_Ring_PSO_SCD through the plots.

Figure 1, figure 2 and figure 3 show the performance of MM-NAEMO, DN-NSGAII and MO_Ring_PSO_SCD respectively on MMF4 test problem. We can observe that the Pareto set obtained from DN-NSGAII has missed large parts. The Pareto sets attained by MM-NAEMO and MO_Ring_PSO_SCD are very similar. However, MM-NAEMO has a better and more converged Pareto optimal front in comparison with other two algorithms.

MM-NAEMO visibly outperforms DN-NSGAII and MO_Ring_PSO_SCD in terms of converging to the Pareto optimal sets for MMF9 test problem. This can be observed

TABLE II: Mean value of 1/HV & IGDF.

Test Problems	1/HV					IGDF				
	MM-NAEMO	MO_Ring_PSO_SCD		DN-NSGA-II		MM-NAEMO	MO_Ring_PSO_SCD		DN-NSGA-II	
MMF1	1.3287	**1.1484**	-	1.1495	-	0.0040	**0.0037**	-	0.0043	+
MMF2	1.2678	**1.1848**	-	1.1953	-	**0.0083**	0.0208	+	0.0338	+
MMF3	1.7075	**1.1741**	-	1.1918	-	**0.0085**	0.0155	+	0.0307	+
MMF4	1.8597	1.8620	+	**1.8577**	-	0.0034	0.0037	+	**0.0032**	-
MMF5	1.1519	**1.1485**	=	1.1490	=	**0.0037**	0.0037	=	0.0040	=
MMF6	1.2215	**1.1484**	-	1.1487	-	0.0036	**0.0035**	=	0.0036	=
MMF7	1.1833	**1.1484**	-	1.1498	-	**0.0035**	0.0037	+	0.0040	+
MMF8	2.3841	2.4077	+	**2.3813**	-	**0.0037**	0.0048	+	0.0040	+
MMF9	0.1053	**0.1034**	-	0.1034	-	0.0479	0.0160	-	**0.0141**	-
MMF10	0.0803	**0.0796**	-	0.0818	+	**0.0659**	0.2073	+	0.1856	+
MMF11	0.0707	0.0690	-	**0.0689**	-	0.0931	**0.0881**	=	0.0979	=
MMF12	0.8625	0.6387	-	**0.6361**	-	**0.0196**	0.0631	+	0.0833	+
MMF13	0.0550	0.0544	-	**0.0542**	-	0.1170	**0.1004**	=	0.1516	+
MMF14	0.3476	0.3476	=	**0.3246**	-	0.0810	**0.0798**	-	0.1071	+
MMF15	0.2512	0.2348	-	**0.2316**	-	**0.1113**	0.1734	+	0.2155	+
MMF1_z	1.2367	**1.1483**	-	1.1484	-	**0.0035**	0.0036	+	0.0037	=
MMF1_e	6.1684	1.1865	=	**0.7085**	=	**0.0056**	0.0120	+	0.0301	+
MMF14_a	0.3438	0.3383	=	**0.3270**	-	0.0791	**0.0790**	=	0.1210	+
MMF15_a	0.2485	0.2393	-	**0.2345**	-	**0.1263**	0.1743	+	0.2260	+
SYM-PART simple	**0.0599**	0.0605	+	0.0601	=	0.0471	0.0419	-	**0.0128**	-
SYM-PART rotated	0.0603	0.0606	+	**0.0601**	-	0.0397	0.0471	+	**0.0152**	-
Omni-test	0.0191	0.0190	=	**0.0189**	-	0.0127	0.0422	+	**0.0080**	-

[1] The mean value is calculated over 21 runs for each Test Problem.

from Figure 4, Figure 5 and Figure 6. We see that DN-NSGAII yields a Pareto set with large gaps. The Pareto set obtained by MO_Ring_PSO_SCD is comparatively much better. However, the Pareto set obtained by MM-NAEMO has less variation and is much more converged than those obtained by the other 2 algorithms.

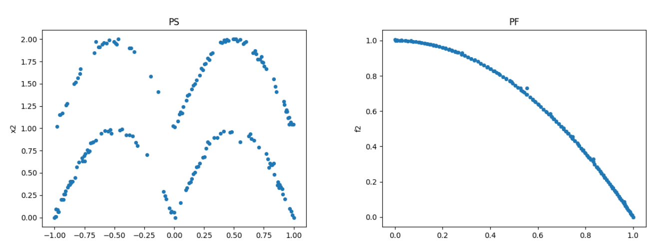
MM-NAEMO clearly outperforms the other two algorithms for MMF14 test problem as well. We can observe from Figure 7, Figure 8 and Figure 9 that DN-NSGAII produces a Pareto optimal front with poor diversity and a comparatively sparse Pareto optimal set although points on both the Pareto sets have been obtained. MO_Ring_PSO_SCD performs better in comparison. However, MM-NAEMO yields an extremely well defined Pareto optimal front as well as Pareto set.

The performance of MM-NAEMO is similar for the other problems as well and outperforms the DN-NSGAII and MO_Ring_PSO_SCD in most of the cases.

V. Conclusion

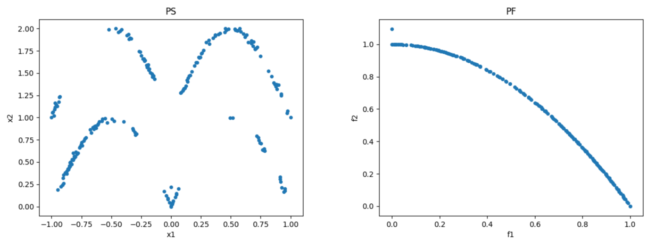
In this paper, we propose a reference point based evolutionary multimodal multi-objective optimization algorithm named as Multimodal Neighborhood-sensitive Archived Evolutionary Many-objective Optimization Algorithm (MM-NAEMO). It is built upon the original framework of NAEMO. MM-NAEMO has been tested on MMO test problems suite of CEC 2019. Results show that MM-NAEMO performs significantly well on these problems and outperforms some previous state-of-the-art algorithms for multimodal multi-objective optimization.

MM-NAEMO still has a large scope for improvement. There are two points that shall be addressed in future. First is improving the convergence further. The second is adaptively choosing the number of clusters for performing the Gaussian Mixture Model based clustering. In this case, we have assumed



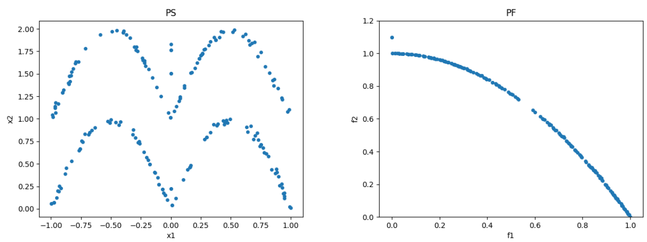
(a) (b)

Fig. 1: (a) Obtained Pareto sets for MMF4 by MM-NAEMO (b) Obtained Pareto front for MMF4 by MM-NAEMO



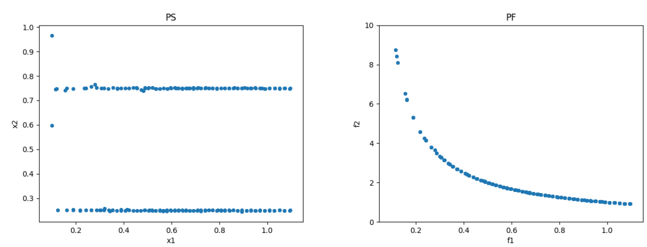
(a) (b)

Fig. 2: (a) Obtained Pareto sets for MMF4 by DN-NSGAII (b) Obtained Pareto front for MMF4 by DN-NSGAII



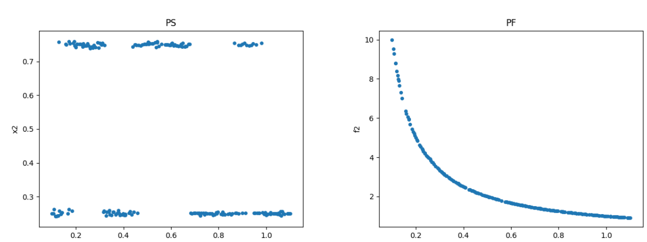
(a) (b)

Fig. 3: (a) Obtained Pareto sets for MMF4 by MO_Ring_PSO_SCD (b) Obtained Pareto front for MMF4 by MO_Ring_PSO_SCD



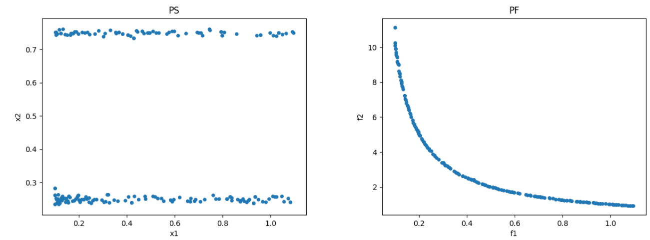
(a) (b)

Fig. 4: (a) Obtained Pareto sets for MMF9 by MM-NAEMO (b) Obtained Pareto front for MMF9 by MM-NAEMO



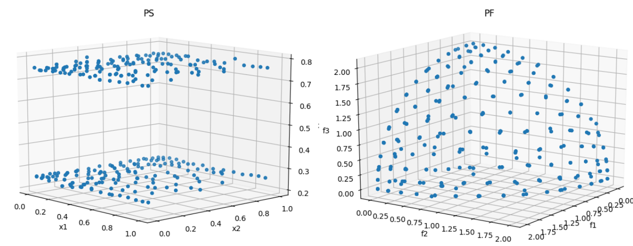
(a) (b)

Fig. 5: (a) Obtained Pareto sets for MMF9 by DN-NSGAII (b) Obtained Pareto front for MMF9 by DN-NSGAII



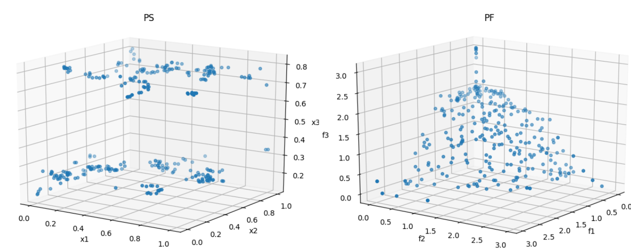
(a) (b)

Fig. 6: (a) Obtained Pareto sets for MMF9 by MO_Ring_PSO_SCD (b) Obtained Pareto front for MMF9 by MO_Ring_PSO_SCD



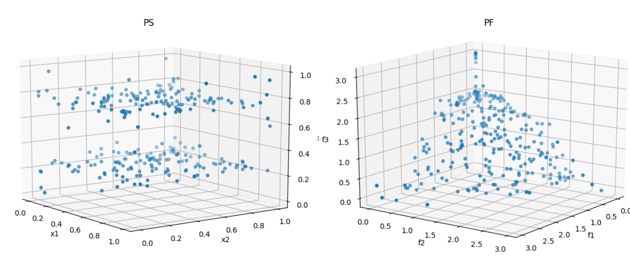
(a) (b)

Fig. 7: (a) Obtained Pareto sets for MMF14 by MM-NAEMO (b) Obtained Pareto front for MMF14 by MM-NAEMO



(a) (b)

Fig. 8: (a) Obtained Pareto sets for MMF14 by DN-NSGAII (b) Obtained Pareto front for MMF14 by DN-NSGAII



(a) (b)

Fig. 9: (a) Obtained Pareto sets for MMF14 by MO_Ring_PSO_SCD (b) Obtained Pareto front for MMF14 by MO_Ring_PSO_SCD

2 clusters for all the problems. However, it is necessary that a system should be developed to detect the right number of clusters automatically while executing the algorithm.

REFERENCES

[1] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.

[2] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[3] Franciszek Seredynski. *Evolutionary Paradigms*, pages 111–145. Springer US, Boston, MA, 2006.

[4] N. N. Glibovets and N. M. Gulayeva. A review of niching genetic algorithms for multimodal function optimization. *Cybernetics and Systems Analysis*, 49(6):815–820, Nov 2013.

[5] Michael Scott Brown, Michael J. Pelosi, and Henry Dirska. Dynamic-radius species-conserving genetic algorithm for the financial forecasting of dow jones index stocks. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 27–41, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] B. Y. Qu, Y. S. Zhu, Y. C. Jiao, M. Y. Wu, Ponnuthurai N. Suganthan, and J. J. Liang. A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. *Swarm and Evolutionary Computation*, 38:1–11, 2018.

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

[8] Raunak Sengupta, Monalisa Pal, Sriparna Saha, and Sanghamitra Bandyopadhyay. Naemo: Neighborhood-sensitive archived evolutionary many-objective optimization algorithm. *Swarm and Evolutionary Computation*, 2018.

[9] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec 2007.

[10] Indraneel Das and J Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8, 07 2000.

[11] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *Evolutionary Computation, IEEE Transactions on*, 12:41 – 63, 03 2008.

[12] Aimin Zhou, Qingfu Zhang, and Yaochu Jin. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *Evolutionary Computation, IEEE Transactions on*, 13:1167 – 1189, 11 2009.

[13] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

[14] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, Aug 2014.