





Identifying Pareto Fronts Reliably Using a Multi-Stage Reference-vector-based Framework

Kalyanmoy Deb , Fellow, IEEE, Claudio Lucio do Val Lopes , Flávio Vinícius Cruzeiro Martins ,
Elizabeth Fialho Wanner 

COIN Report 2023012

Abstract—Evolutionary multi-objective and many-objective optimization (EMO and EMaO) algorithms are increasingly being used to identify the true shape and location of the Pareto-optimal front using a few representative well-converged and well-distributed solutions. The reason for their popularity is due to their ability to provide a better understanding of objective relationships for optimal solutions, and also to facilitate the choice of a preferred solution using an interactive or post-optimal multi-criterion decision analysis. However, since EMO and EMaO algorithms are stochastic, a single application may not provide a true representative set with a desired number of Pareto solutions reliably in repetitive runs and importantly with a well-distributed set of solutions. In this paper, we propose a multi-stage framework involving reference-vector based evolutionary multi- and many-objective algorithms (MuSt-EMO and MuSt-EMaO) that attempts to recursively rectify shortcomings of previous stages by careful executions of subsequent stages so that a prescribed number of well-distributed and well-converged solutions are achieved at the end. The proposed multi-stage approach is implemented to a number of popular reference vector based EMO/EMaO algorithms and is applied on various multi- and many-objective test and real-world problems.

Index Terms—Multi-objective optimization, evolutionary algorithms, Pareto front, Gap-filling method.

I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) are regularly being used to find a set of trade-off Pareto solutions for multi- and many-objective optimization (EMO and EMaO) problems. However, there are a number of known shortcomings of EMO and EMaO algorithms which have been recognized in the literature in achieving adequate convergence and distribution of non-dominated solutions, but still stay as open issues required to be resolved using systematic approaches.

First, since EMO and EMaO algorithms are stochastic in nature, a single application may not always produce a reproducible well-distributed and well-converged set of non-dominated (ND) solutions (Pareto or near-Pareto solutions) over multiple applications. EMO researchers often use multiple runs, each starting with a different initial population and

present a median or average performing result. This is achieved by presenting the ND set obtained by the median or mean-performing hypervolume (HV) or inverse generational distance (IGD) run, or by using a more sophisticated attainment surface method [1].

Second, EMO and EMaO algorithms are expected to produce a prescribed number of ND solutions, usually dictated by the chosen population size N (and/or number of reference vectors R) [2], [3], although the number of solutions needed to adequately represent a ND set may not be known a priori. But every EMO or EMaO run may not produce the exact number of ND points as desired. This can be mainly due to two reasons. First, the stochasticity involved in an algorithm's operators may have failed to find an ND solution for every specified reference vector (RV). Second and more likely, since the exact location and shape of the Pareto front are not known before a run, all specified reference vectors may not associate with an Pareto solution, thereby causing less than $|R|$ ND solutions to be found at the end of the run. These issues make a comparison between two sets of ND solutions difficult. For example, a reliable comparison using the HV metric or any uniformity measure expects both sets to have an identical number of ND solutions. Moreover, since a RV-based EMO or EMaO algorithm is designed to process $|R|$ (usually equal to the population size N) reference vectors, fewer ND solutions in a population cause a waste of overall computational effort and memory. Third, a set of ND solutions may indicate certain gaps in the apparent ND front discovered by an EMO or EMaO algorithm. A gap in the ND front may artificially arise from discovering fewer points in a certain area of the Pareto front or a gap may truly exist in the Pareto front. Usually, a gap-confirming method [4] with a focused EMO [5] or focused EMaO [6] is employed to confirm the true existence of a gap. However, if the true or artificial existence of gaps can be confirmed and repaired during the optimization process, and not as a post-optimal process, the resulting EMO/EMaO application is expected to be more efficient and reliable.

Thus, although these known shortcomings are addressed with certain other additional execution of specific tasks, it would be desired if the EMO/EMaO algorithms are able to find a more reliable and reproducible ND set having exactly the desired number of a uniformly distributed set of points on the entire Pareto front with a clear indication of true gaps and holes, if any.

In the recent past, EMO researchers have emphasized making a balance between convergence and diversity in finding final ND solutions and proposed multi-phase EMO and EMaO

Corresponding author: K. Deb is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, USA (e-mail: kdeb@egr.msu.edu, see <http://www.coin-lab.org> for more information).

Cláudio Lúcio V. Lopes is with Postgraduate Program in Mathematical and Computational Modeling, CEFET-MG, Brazil (email: claudiolucio@gmail.com)

Flávio V. C. Martins is with the Computer Department at the Centro Federal de Educação Tecnológica de Minas Gerais, 30510-000 Brazil (e-mail: flaviocruzeiro@cefetmg.br).

Elizabeth F. Wanner is with the Computer Department at the Centro Federal de Educação Tecnológica de Minas Gerais, 30510-000 Brazil (e-mail: efwanner@cefetmg.br).

algorithms. Balanced NSGA-III (B-NSGA-III) [7] approach identified one of the three possible stages – extreme point identification phase, uniform diversity enforcement phase, and local improvement phase. A recent study [8] has followed three different distinct phases – convergence phase, diversity preservation phase, and convergence-diversity phase. These approaches focus on identifying gaps and holes within their obtained ND front either explicitly or through their innovative diversity evaluation metrics. But, none of these methods has focused on finding a pre-specified number of ND solutions, which are evaluated during the optimization process to possess the convergence and diversity properties without any artificial holes or gaps as a collection of final solutions with a given number of solution evaluations.

Recent archive-based EMO/EMaO methods propose to save all ND solutions from every generation from the start to the end of a run in an archive and then choose the desired number of ND solutions from the archive as the outcome of the overall algorithm [9], [10]. Expecting that the archive will grow into a huge set, selection of a few solutions from it will be time-consuming. While it may be useful for practical problems requiring to utilize a small budget of solution evaluations, such a strategy does not demand a careful balance between convergence and diversity among population members to be carried out at every iteration, making EMO/EMaO methods less appealing in an algorithmic sense.

In this paper, we attempt to address these issues by proposing a three-stage framework to improve the performance of existing evolutionary multi and many-objective optimization algorithms. The first stage attempts to find a set of ND solutions roughly on the entire Pareto front by using a standard EMO or EMaO algorithm. Based on the achieved distribution of solutions, the second stage focuses on to fill the apparent gaps and holes not covered by the first stage. In the third stage, solutions from both stages are combined, and a final EMO/EMaO algorithm is run in two iterations with a well-estimated number of RVs to arrive at the exact desired number of final ND solutions having a better convergence and uniformity. Later stages use populations obtained at the previous stages to maintain continuity and also to reduce overall computational complexity. Moreover, repeated emphases at critical regions of the search space ensure that the obtained solutions are reliably close to the true Pareto front.

In the remainder of the paper, we briefly review of the main focus in EMaO algorithms and motivate the need for a multi-stage framework involving EMaO algorithms for reliably finding a specific number of ND solutions in Section II. The multi-stage framework is presented in detail next in Section III with the help of a constrained test problem and a unconstrained practical problem. Results on a number of three to 10-objective problems are presented in Section IV. A parametric study of a single hyper-parameter of the proposed framework is presented in Section V. Finally, conclusions of this extensive study are summarized in Section VI.

II. EXISTING EVOLUTIONARY ALGORITHMS AND MOTIVATION FOR THIS STUDY

Evolutionary algorithms have a clear niche in finding multiple Pareto solutions for multi- (having two or three objectives) and many-objective (having more than three objectives) optimization problems in a single run [11], [12] compared to their traditional counterparts which mostly work by finding a single Pareto solution in a generational manner [13]. In EMO and EMaO algorithms, usually an emphasis for non-dominated solutions to achieve convergence and an emphasis for diverse objective solutions to achieve a uniform distribution are established either hierarchically or simultaneously. In EMaO algorithms, a set of pre-specified normalized RVs are used to help achieve a better distribution of solutions. Despite their ever-increasing popularity in both developmental and application studies, there are a few shortcomings which we highlight here, handling of which becomes the main motivation of this paper.

A. Reliable Convergence and Uniform Distribution of Solutions

EMO and EMaO algorithms start with an initial population and use stochastic search operators to iteratively improve two aspects – convergence towards the Pareto front and maintain a uniform distribution of solutions in the objective space. Convergence aspect is achieved mostly by providing more selection pressure for non-dominated solutions. Diversity of solutions is achieved by various means, including a crowding measure estimating the number of solutions in the neighborhood. Original MOEA/D [14] proposed the diversity maintenance through a pre-defined uniform set of reference vectors¹ in the objective space originating from the ideal point and preferring solutions that are close to RVs and also close to the ideal point. This revolutionary idea has been particularly found to work well for many-objective problems.

While two aspects – convergence and diversity – are undoubtedly the main emphasis of any EMO and EMaO algorithms, their order and extent of use within an algorithm are not yet settled. Some algorithms (such as, NSGA-II [16], NSGA-III [2], SPEA2 [17] and others) use a hierarchical ‘convergence first, diversity second’ strategy, whereas recent studies have raised questions about this strategy [18]. Too much focus on convergence early on may lead to a loss of diversity in population members, thereby making a diversity enhancement difficult in certain problems. It is clear that a ‘diversity first, convergence second’ strategy may not be that useful, as although a well-distributed set of solutions is possible to achieve early on the search space far away from the Pareto front, it may be difficult to maintain the diversity all along towards the convergence phase of the algorithm at the latter part of the run. However, a number of studies have proposed the need to strike a balance between convergence and diversity from start to finish during a run.

The two-archive evolutionary algorithm for constrained multi-objective optimization, C-TAEA, proposed in [19], tries

¹Recent studies suggested non-uniform distribution of reference vectors based on Pareto front shapes [15].

to balance convergence and uniformity of solutions in a two-archive procedure that simultaneously maintains two populations: the convergence archive (CA) and the diversity archive (DA). The former supports the convergence and feasibility in the evolutionary process while the latter explores areas that the convergence archive has not exploited yet. In this algorithm, the reference lines are also used in the update mechanism of the CA and DA. We use this RV-based algorithm as a potential method for improvement by our proposed multi-stage framework.

The MOEA/D-AWA [20] uses a two-stage strategy to deal with the generation of the weight vectors. In the first stage, pre-determined weight vectors are used until the population is considered converged. Then, in the second stage, some weight vectors are adjusted based on deleting the overcrowded sub-problems and inserting new ones in the sparse areas. However, such insertion may not satisfactorily adapt to the problems with disconnected Pareto fronts.

In [7], B-NSGA-III is presented as a multiphased many-objective evolutionary optimization algorithm capable of automatically balancing convergence and diversity of population members. The algorithm uses three phases dynamically. In phase 1, it looks for extreme points. In phase 2, it tries to cover gaps and extend extreme points found in the non-dominated front. And finally, in phase 3, it concentrates the efforts towards helping poorly converged non-dominated solutions. The algorithm can interchange from one phase to another if population properties trigger for such a change. Results show superior performance when compared to several state-of-the-art algorithms in a set of benchmark problems. However, this approach requires an internal local search method, making the overall algorithm not easy to compare with pure evolutionary algorithms for a fixed budget of solution evaluations.

In [21], a multi-stage constrained multi-objective evolutionary algorithm, CMOEA-MS, is proposed. The algorithm exclusively focuses on constrained problems and balances objective optimization and constraint satisfaction, giving different priorities in each phase. The first phase indicates that most solutions are infeasible, and these solutions can support the population to leave the infeasible regions. The second phase points out that most solutions are feasible, so more feasible solutions can be found to help the population spread along the feasible boundaries. The general strategy uses different priorities of objectives and constraints in the two stages. The work presents some experiments based on constrained problem sets. It does not mainly address the diversity/uniformity of solution sets.

A many-objective evolutionary algorithm, dubbed as CLIA, combining two interacting processes, Cascade Clustering (CC) and Reference Point Incremental Learning, is proposed in [22]. The cascade clustering (CC) uses the non-dominated and dominated solutions to create clusters, sort, and select solutions for a better convergence and diversity. CC uses reference lines to create and sort the solutions. In the second process, a support vector machine (SVM) model is applied in an incremental learning approach, classifying the reference points as active/inactive and creating denser reference lines (generated on the unit simplex) and is used in the

optimization process. The incremental SVM model adjusts the RVs during the evolution process. By hybridizing an evolutionary approach with SVM, an adaptive adjustment of RVs provides a better distribution of ND solutions. We also integrate this algorithm with our multi-stage framework to improve its performance. RVRL-EA [23] also adapts reference vectors using reinforcement learning. However, it does not use a multi-stage framework and can only be integrated with decomposition-based algorithms. Experiments have shown that the algorithm was competitive compared to CLIA.

MSEA [8], a multistage EMaO algorithm, is designed for improving the diversity performance of EMaO. In Stage 1, the population is assessed based on a convergence metric. Then, in Stage 2, the diversity is assessed again; a particular selection algorithm generates offspring to replace one solution in the population. Finally, in stage 3, there is a mating selection for a parent with good convergence and diversity. The authors present results for multi-objective problems only, and the method presents a cubic computational complexity related to the number of solutions. The algorithm's extension to many-objective problems were not demonstrated and also MSEA does not use any RV. Thus, we exclude its comparison with our proposed framework.

These existing multi-stage EMaO algorithms attempt to stress convergence, diversity, or both, whenever needed during the evolutionary process, but lack significant actions towards reinforcing the true convergence and diversity of evolving solutions. In the single-objective evolutionary algorithm literature, *restarts* are often employed to help an algorithm to escape from premature convergence and also to reinforce convergence to true optimal solutions. However, due to multi-faceted goals in multi-objective optimization, restarts must have different goals. From a prematurely stuck existing ND solution set, how would a restart be initiated so that convergence, diversity, or both can be achieved in the next phase, depending on what is lacking in the stuck set of population members? Such a restart strategy should not only be disruptive to the previous stuck phase and, if done properly, must address various aspects of multi-objective problem solving in a systematic manner and help produce a more reliable set of ND solutions. Our proposed multi-stage framework is a step toward this effort.

B. Fewer than Expected Number of Solutions

In an EMO or EMaO, we usually target a set (say N) of well-distributed and well-converged ND solutions. The parameter N is user-supplied. If a run produces less than N ND solutions at the end, there are at least two difficulties. First, it does not fulfill the stated requirement and second, it may be difficult to compare or statistically come up with a single performance measure for two or more competing such sets having different number of points. No existing method guarantees this aspect and most often, solutions from a big archive are chosen to extract exactly N solutions [24]. Since an archive is not usually directly used in the evolutionary process, this may not produce the best possible distribution achievable with N ND points and, moreover, dilutes the spirit of algorithmic search for N well-distributed solutions.

C. Gaps in the Obtained Non-dominated Front

Some algorithms were proposed to identify apparent gaps in obtained Pareto front and employ additional tasks to fill the gaps. In [4], a three-step algorithm was proposed to deal with identifying and filling gaps in obtained ND fronts. In the first step, a well-distributed and converged solution set is obtained (using an EMO/EMaO algorithm). The second step tries to identify whether there are real gaps (based on inner problem characteristics) or not. Two metrics are proposed to help in this step. The third step attempts to fill the gap based on the outcome of the second step. This study only attempted to find and fill gaps, but did not focus on finding a pre-defined number of ND points, nor did the study make the steps use points from previous steps to make the overall approach more computationally efficient.

Based on the above discussions, we set our goal of arriving at a multi-stage framework involving an EMO/EMaO algorithm having the following properties:

- 1) A coherent and seamless three-stage algorithmic framework with a possibility of skipping latter stages based on the number of well-distributed ND solutions achieved,
- 2) A multi-stage framework that attempts to find as close to the desired number of ND points and having as uniformly distributed as possible on the entire feasible Pareto front,
- 3) A multi-stage framework that repeats its search among critical parts of the search space to ensure a reliable set of well-converged and well-distributed ND points, and
- 4) A multi-stage framework which is pragmatic in using existing seed solutions, implementable with any existing reference vector based EMO/EMaO algorithm, executed for a budget of solution evaluations, and controlled with a single well-tuned parameter.

III. PROPOSED MULTI-STAGE FRAMEWORK

Our proposal is a multi-stage framework that improves the performance of a specific reference vector based EMO/EMaO algorithm (such as, NSGA-III [2] or MOEA/D [14]) to find a prescribed number of repeatable, well-converged and well-distributed set of Pareto solutions (N). The chosen RV-based EMaO algorithm uses a set of reference vectors (\mathbf{R}) usually equal to the population size N , an initial seed population \mathbf{P}^{seed} of any size, and the preset number of solution evaluations (SEs) T , and produces a set of non-dominated solution set \mathbf{S} of size N . Note also that the size of the supplied initial population \mathbf{P}^{seed} need not be same as N or $|\mathbf{R}|$. The initial population generation operator creates exactly $N = |\mathbf{R}|$ population members at the first generation by the EMaO algorithm's operators from \mathbf{P}^{seed} . The proposed MuSt-EMaO uses three stages, dividing T SEs among them satisfying $T = T_1 + T_2 + T_3$, where T_i is the number of SEs for the i -th stage. The pseudo-code of the proposed MuSt-EMaO is presented in Algorithm 1.

Stage 1 of MuSt-EMaO framework makes the first attempt to generate an evenly distributed ND solution set using an existing reference vector based approach (the baseline EMaO algorithm). We apply the Riesz s-energy method to create exactly N reference lines (\mathbf{R}_1) by *GenerateReferenceVectors*

Algorithm 1: MuSt-EMaO Framework.

Input: EMaO (optimization algorithm), N (desired number of solutions), $T = (T_1, T_2, T_3)$ (number of solution evaluations), \mathbf{P}^{seed} (initial population)

Output: \mathbf{S} (a reliable and well-distributed ND solution set)

```

1 // Stage 1
2  $\mathbf{R}_1 \leftarrow \text{GenerateReferenceVectors}(N)$ ;
3  $\mathbf{S}_1 \leftarrow \text{EMaO}(\mathbf{R}_1, \mathbf{P}^{\text{seed}}, T_1)$ ;
4  $[\mathbf{S}_{1c}^{T_1}, \mathbf{R}_{1\bar{c}}] \leftarrow \text{Classify}(\mathbf{R}_1, \mathbf{S}_1)$ ;
5 if  $|\mathbf{S}_{1c}^{T_1}| < N$  then
6   // Stage 2
7    $\mathbf{S}_2 \leftarrow \text{EMaO}(\mathbf{R}_{1\bar{c}}, \mathbf{S}_1, T_2)$ ;
8    $\mathbf{S}_{12} \leftarrow \text{NonDominance}(\mathbf{S}_{1c}^{T_1} \cup \mathbf{S}_2)$ ;
9    $[\mathbf{S}_{3c}^0, \mathbf{R}_{12\bar{c}}] \leftarrow \text{Classify}(\mathbf{R}_1, \mathbf{S}_{12})$ ;
10 else
11    $\mathbf{S}_2 = \emptyset, \mathbf{S}_{3c}^0 = \mathbf{S}_{1c}^{T_1}$ ;
12    $T_3 \leftarrow T_2 + T_3$ ;
13 end
14 // Stage 3
15  $\mathbf{S}_3^0 \leftarrow \mathbf{S}_1 \cup \mathbf{S}_2$ ;
16  $N_3 = N$ ;
17 for  $i = 1:2$  do
18    $N_3 \leftarrow \text{int}(N(N_3/|\mathbf{S}_{3c}^{i-1}|))$ ;
19    $\mathbf{R}_3^i \leftarrow \text{GenerateReferenceVectors}(N_3)$ ;
20    $\mathbf{S}_3^i \leftarrow \text{EMaO}(\mathbf{R}_3^i, \mathbf{S}_{3c}^{i-1}, T_3/2)$ ;
21    $\mathbf{S}_3^i \leftarrow \text{NonDominance}(\mathbf{S}_{3c}^{i-1} \cup \mathbf{S}_3^i)$ ;
22    $[\mathbf{S}_{3c}^i, \mathbf{R}_{3\bar{c}}^i] \leftarrow \text{Classify}(\mathbf{R}_3^i, \mathbf{S}_3^i)$ ;
23 end
24 if  $|\mathbf{S}_{3c}^2| < N$  then
25    $\mathbf{S}_{3c}^2 \leftarrow \text{NonDominance}(\mathbf{S}_{3c}^1 \cup \mathbf{S}_{3c}^2)$ ;
26 end
27  $\mathbf{S} \leftarrow \text{Reduce}(\mathbf{S}_{3c}^2, N)$ ;

```

operator in Line 2. The procedure first creates N points on the unit-simplex using the *reduction* method outlined in [25] and then each point is adjusted using the Riesz-energy minimization through projected gradient-descent method, a code of which is available at `pymoo` distribution [26]. Then, we execute the EMaO algorithm for a maximum of T_1 SEs (Line 3). EMaO starts by creating an initial population of size N from the supplied \mathbf{P}^{seed} of any size. Applying the baseline EMaO algorithm, we obtain the first non-dominated solution set (\mathbf{S}_1 of size $N_1 (\leq N)$). The next task in Stage 1 is to classify the reference vector set \mathbf{R}_1 into two classes: (i) active reference vector (ARV) set and inactive reference vector (IRV) set by using the proposed *Classify* operator. This operator first associates every member of \mathbf{S}_1 with a reference line based on the shortest normalized Euclidean distance (identical to the d_2 operator in MOEA/D [3] or NSGA-III [2]). Then, all reference vectors for which there is an associated member from set \mathbf{S}_1 are saved in the ARV set. Let us say that the solution set $\mathbf{S}_{1c}^{T_1} (\subset \mathbf{S}_1)$ is associated with the ARV set \mathbf{R}_{1c} . The remaining ND solutions of \mathbf{S}_1 are saved as $\mathbf{S}_{1\bar{c}} = \mathbf{S}_1 \setminus \mathbf{S}_{1c}^{T_1}$ and the corresponding reference vectors are saved in the IRV set as $\mathbf{R}_{1\bar{c}} = \mathbf{R}_1 \setminus \mathbf{R}_{1c}$. Note that the IRV set indicates the region on the unit simplex for which no ND solutions are found in Stage 1.

If the cardinality of $\mathbf{S}_{1c}^{T_1}$ is less than the number of desired solutions (N), then the MuSt-EMaO framework goes to

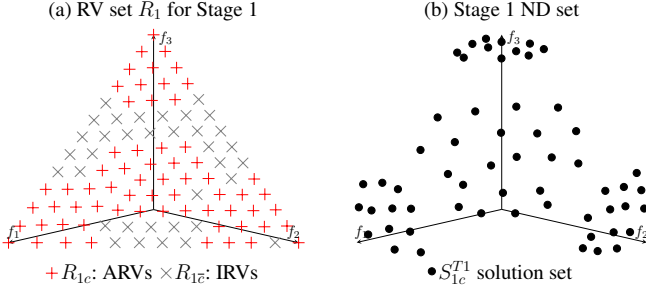


Figure 1: Stage 1 of MuSt-EMaO framework is illustrated on C2-DTLZ2 problem. Of $|\mathbf{R}_1| = 100$ RVs, 65 ND points are obtained with $T_1 = 5,000$ SEs in Stage 1.

Stage 2 (Line 5), in which a separate EMaO is run to find ND solutions for IRVs from the Stage 1 using $R_{1\bar{e}}$. Solutions from Stages 1 and 2 are then combined, *NonDominance* operator finds the ND set in Line 8, and the resulting ND set is classified to find associated points for the original R_1 RVs, generating S_{3c}^0 .

On the other hand, if Stage 1 is able to find an associated point for each R_1 RV, Stage 2 is omitted, and remaining T_2 and T_3 SEs are allocated to Stage 3.

Next, the MuSt-EMaO framework moves to Stage 3, irrespective of whether the total number of ND points obtained from combined Stages 1 and 2 is still less than N , or equal to N . In case of former, Stage 3 creates new ND points by increasing the number of RVs beyond N , so that N points are found by Stage 3 EMaO execution. In case of latter, Stage 3 uses the remaining SEs (T_3) with N RVs to improve the obtained solutions. The first task in Stage 3 is to estimate the number of required reference lines for this purpose. We suggest linearly scaling the population size by making the argument that if N reference vectors have produced S_{3c}^0 ND points in the combined Stages 1 and 2, how many reference vectors are needed to produce N ND points? We calculate this estimate as N_3 in Line 18. The first iteration of Stage 3 applies the same EMaO algorithm with a combined population S_1 and S_2 of all ND solutions found in Stages 1 and 2 as an initial population. The EMaO algorithm is run for N_3 number of reference vectors (set \mathbf{R}_3^1 created in Line 19) for half of the allocated SEs for Stage 3. This produces a new ND set S_3^1 in Line 20. The associated ND points are saved in set S_{3c}^1 . In order to not leave out any previously obtained better distributed ND points, S_{3c}^1 is combined with S_{3c}^0 , and a better set S_3^1 of size of $|\mathbf{R}_3^1|$ is chosen in Lines 21-22. Since at least N RVs are used in Stage 3, it is likely that two iterations will find N or more ND points at the end of Stage 3.

There is still a remote possibility that we are not able to have more than or equal N points. Line 24-25 treats this case. Solutions from S_{3c}^1 and S_{3c}^2 are combined, the *NonDominance* operator is used, generating a new S_{3c}^2 .

Finally, S_{3c}^2 set is then reduced to have exactly N final points in Line 27. The *Reduce* operator eliminates the most crowded solution according to the Riesz S-energy one by one until N solutions remain.

A. An Illustration

We illustrate the working of the proposed MuSt-EMaO framework with NSGA-III through a simulation on three-objective C2-DTLZ2 constrained problem. Figure 1(a) shows the active reference point set on the unit simplex by '+' that have an associated ND point from the final population of NSGA-III algorithm run with $N = |\mathbf{R}| = 100$ reference points and other standard parameter settings [2]. The respective ND objective vectors are shown with circles in Figure 1(b). A total budget of solution evaluations of $T = 20,000$ is divided as $T_1 = T_2 = 5,000$ and $T_3 = 10,000$. It is clear that each ARV has at least one ND solution assigned to it. For this problem, Stage 1 is able to find $|\mathbf{S}_{1c}^{T_1}| = 65$ ND solutions. Each RV in the IRV set (35 RVs represented with a 'x') does not have an ND solution associated with it in Stage 1. This can be a failure on the part of the NSGA-III algorithm or the C2-DTLZ2 problem does not possess any true Pareto solution associated with the IRVs. This is intended to be verified at Stage 2 by executing NSGA-III again with a special focus in attempting to find Pareto solutions corresponding to the IRV set only. This makes the MuSt-EMaO framework more reliable than a single algorithmic run with a focus on all RVs on the entire unit simplex.

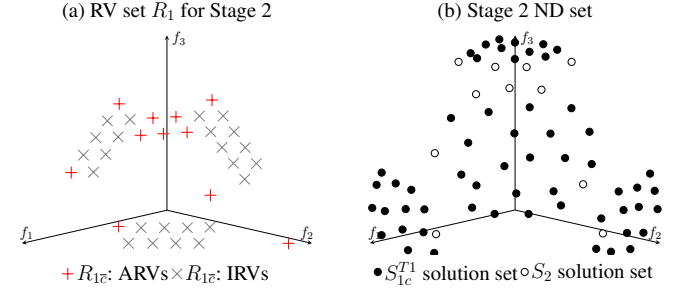


Figure 2: Stage 2 of MuSt-EMaO framework is illustrated on C2-DTLZ2 problem. 11 new ND points are found in Stage 2, making a total of 76 points (out of 100 original RVs) obtained with $T_1 + T_2 = 10,000$ SEs.

In Stage 2, the same EMaO algorithm is applied with the Stage 1 ND set S_1 as the initial population, but only with the inactive reference vector set $R_{1\bar{e}}$. This allows to focus the search on the part of the Pareto front left out at Stage 1 and finds a new ND set S_2 in Line 7. In the illustrative problem, $|\mathbf{R}_{1\bar{e}}| = 35$. Stage 2 NSGA-III execution finds $|\mathbf{S}_2| = 11$ new ND points that were not found in Stage 1, as shown in Figure 2. The non-dominated solutions of the ND sets of Stages 1 and 2 ($S_{1c}^{T_1}$ and S_2) are identified and stored in S_{12} in Line 8. 65 and 11 (or $|\mathbf{S}_{12}| = 76$) ND solutions are shown in Figure 2(b) with solid and open circles, respectively. The respective ARV points (for Stage 2 solutions) are shown in Figure 2(a) with a '+' symbol. This ND set is then classified using the original entire reference set \mathbf{R}_1 to identify the associated points S_{3c}^0 and the IRV set $\mathbf{R}_{12\bar{e}}$ in Line 9. An explicit focus in undiscovered region provides the reliability of the proposed framework, as depicted in the figure. The total number of ND points found after Stages 1 and 2 are $|\mathbf{S}_{3c}^0| = 76$ for the C2-DTLZ2 example.

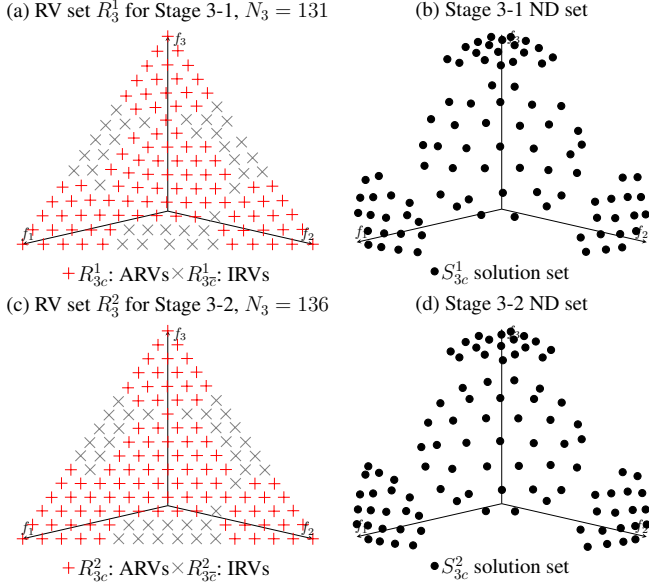


Figure 3: Two iterations of Stage 3 of MuSt-EMaO framework are illustrated on the C2-DTLZ2 problem. 20 more ND points are found on Stage 3, Iteration 1 ((a) and (b)), making a total of 96 points, and, finally, 4 more points are found on Stage 3, Iteration 2 ((c) and (d)), making 100 points at the end. Both iterations of Stage 3 use $T_3 = 10,000$ SEs.

The first iteration of Stage 3 estimates that $N_3 = \text{int}(100(100/76)) = 131$ reference vectors are needed. Figures 3(a) and (b) show that 96 ND solutions are found. These points are denser but are all within the feasible region of the Pareto front. However, the number of obtained points is close, but still not equal to the desired number $N = 100$.

The above linear estimation procedure for an increased number of reference vectors R_3^1 does not require any complicated information on the region where more RVs are required to be added, as proposed in the adaptive NSGA-III procedure [27]. More RVs are simply placed on the entire unit simplex. In the second iteration, the required number of RVs is estimated to be $N_3 = \text{int}(100(131/96)) = 136$. Figures 3(c) and (d) show the obtained 100 ND points (equal to the desired number of $N = 100$ points), which are slightly more packed compared to iteration 1 of Stage 3. It is clear that without the iterative process, it would have been difficult to predict exactly how many RVs were required to be chosen the desired number of Pareto solutions.

One can argue that instead of going through two iterations within Stage 3, one can double or triple the number of RVs and make a single iteration in Stage 3 and then use the reduce operator to adjust the final ND points to N . This alternate approach is not viable in a few ways. First, as argued above, it is difficult to know beforehand how many RVs are needed to produce N well-distributed ND points. Second, there is waste of computational effort in dealing with unnecessarily large RVs. Our two-iteration approach in Stage 3 makes a good compromise among the naive larger-than-required RV approach and many meticulous iterations to slowly reach the desired number of RVs.

Note that the proposed MuSt-EMaO framework does not demand too many critical parameters. The N is the desired number of ND solutions and is not a parameter. The total SEs, T , is also not a parameter, but two of the three SEs: T_1 , T_2 , and T_3 are the only required parameters to be set. In this sense, we intend to have only one parameter: γ , which is the proportion of T to be set for T_3 : $T_3 = \gamma T$. We assume $T_1 = T_2 = \frac{1-\gamma}{2}T$. We perform a parametric study in Section V to suggest a suitable value of γ .

IV. RESULTS

In this section, first, computational tests are designed aiming to assess the performance of the proposed multi-stage framework with NSGA-III [2] as an EMaO algorithm. In this case, well-known quality indicators are employed, and the MuSt-NSGA-III is compared with the baseline NSGA-III approach on a set of benchmark problems. Thereafter, the general applicability of the proposed MuSt-EMaO approach is tested on other EMaO algorithms. In this case, our goal is to investigate if the performance of different reference-based EMaO algorithms can be improved by the multi-stage framework proposed above.

A. Proof-of-Principle Results of MuSt-NSGA-III Approach

First, we integrate the proposed MuSt-EMaO framework with NSGA-III and attempt to solve two-objective and three-objective problems to gain insights into the working principle of the proposed multi-stage framework. To demonstrate the distribution of final ND points obtained from NSGA-III and MuSt-NSGA-III algorithms, we plot the objective values of Pareto solutions from the median performing runs for the three-objective MaF07 problem in Figure 4. In all cases, N is set to 100. Both algorithms are implemented using the pymoo framework [26]. In all tests, both algorithms are executed with $T = 20,000$ for all problems and, for multi-stage algorithms, we use $T_1 = T_2 = 5,000$, so that $\gamma = 1/2$. Other NSGA-III parameters are set to their standard settings: $p_c = 1$, $p_m = 1/n$, $\eta_c = 30$, and $\eta_m = 20$ [16]. It is clear from the figure that MuSt-NSGA-III is able to find a better distributed set of Pareto solutions compared to NSGA-III. This visual comparison is aided with quantifiable indicators in Table I. There are exactly 100 ND points spread almost uniformly on the four clusters of the Pareto front. Despite many reference vectors in the original set R_1 not conforming to any Pareto solution, our MuSt-NSGA-III is able to find more active reference vectors in Stage 3 and provide a well-distributed Pareto front with the same number of SEs ($T = 20,000$) as the baseline algorithm.

Another example is the crashworthiness problem depicted in Figure 5. The non-dominated solutions from the median HV over the 50 runs are shown. It is clear from the figure that MuSt-NSGA-III procedure is able to find a better distribution with more solutions than NSGA-III. Similar behaviour is also observed for other problems, supporting the superiority of the proposed multi-stage framework from the five uniformity indicators used here. Clearly, with 100 RVs on the entire unit

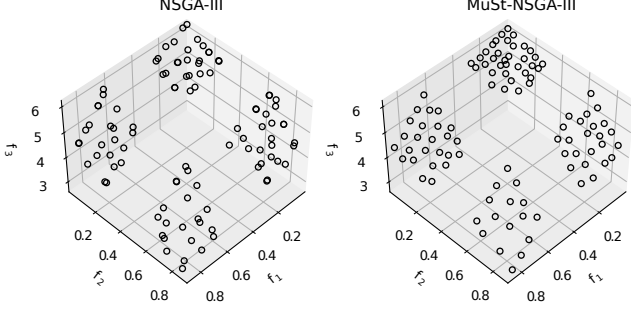


Figure 4: NSGA-III and MuSt-NSGA-III results for the median HV run for MaF07 problem show a better distribution obtained by the latter with identical SEs.

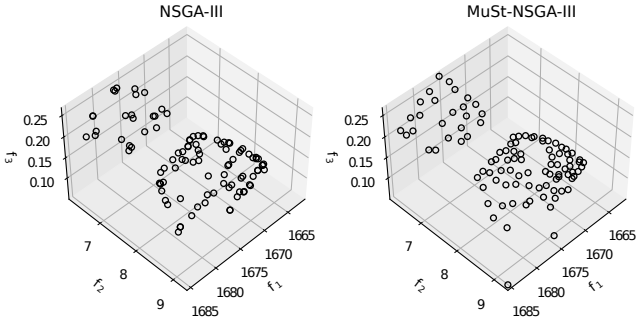


Figure 5: NSGA-III and MuSt-NSGA-III sample results for the crashworthiness problem. 50 runs are executed and the ND points from the median HV run are plotted for each case.

simplex in NSGA-III, only 38 RVs are found to have associated population members. No matter how many generations are executed, this number will not change much, but the objective vectors may come closer to the RVs making the overall distribution better with generations. However, NSGA-III does not put any specific emphasis in arranging the distribution of other ND solutions, which are not the closest associated RV solutions. Thus, locations of these additional ND solutions are not expected to make the overall distribution uniform. The top plot in Figure 6 marks each of the active reference points on the unit simplex. First, notice that there are many inactive reference points on the simplex, shown with a ‘—’. Second, notice from the shape of the markers indicating the number of ND population members that are associated with each ARV is that some reference points have more associated population members and some are not. All 100 ND points obtained by NSGA-III are also shown in left plot in Figure 5, clearly depicting the non-uniform density of points. This is expected to happen to problems in which not all RVs of the unit simplex are expected to have an associated point.

However, in Iteration 2 of Stage 3, 335 RVs were estimated to obtain 100 ND solutions. The left plot in Figure 6 shows all 335 reference points on the unit simplex and associated reference points are marked with different markers according

to the number of neighboring ND population members associated with them for the median HV run. Only 59 of 335 reference points are active. However, when the active reference points are marked with MuSt-NSGA-III ND points, there are 87 reference points with one population member, 9 reference points with two population members and two reference points with 3 population members found. This makes a total of 100 points having a much closer to uniform distribution than NSGA-III-obtained points. Since our density adjustments are made on the entire unit simplex uniformly by increasing the number of RVs, the uniformity in the distributed ND points is also expected to be better on the Pareto front.

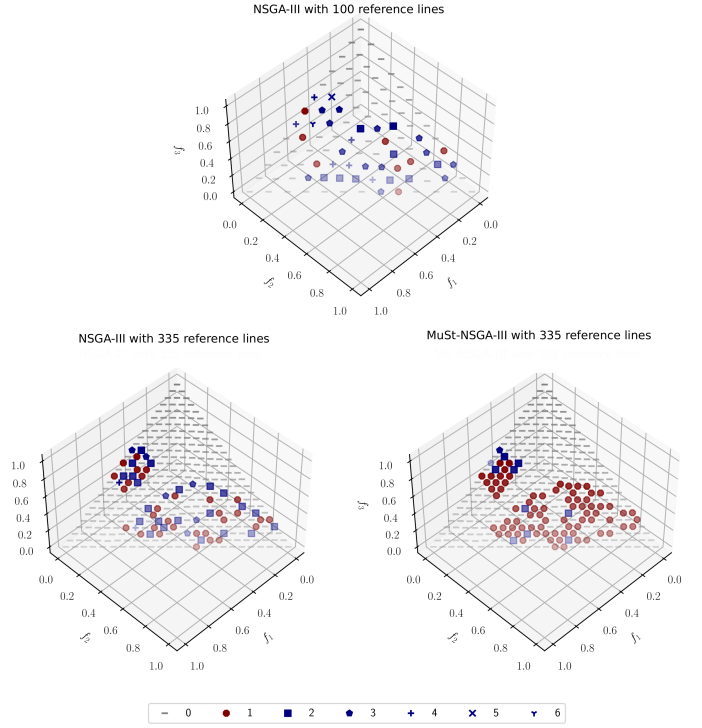


Figure 6: Active RVs from NSGA-III on unit simplex with 100 and 335 reference points, respectively, at the top and bottom-left. Associated RVs for MuSt-NSGA-III ND solutions are shown at the bottom-right, which uses the same 335 reference points on the entire unit simplex. Each symbol on the plots indicates the number of solutions assigned to each RV. MuSt-NSGA-III finds more active RVs.

A convergence behavior through HV can give us another perspective of how the three stages of our proposed framework work compared to the baseline algorithm. Figure 7 shows the average HV variation over a number of SEs from 50 runs for three different cases of HV computation executed after every generation of each algorithm and not from an accumulated archive with (i) the closest MuSt-NSGA-III ND population member to each associated RV, marked as ‘MuSt-NSGA-III’, (ii) all NSGA-III ND population members, marked as ‘Extended NSGA-III’, and (iii) the closest NSGA-III ND population member to each associated RV, marked as ‘NSGA-III’. The number of such population members are also shown with values shown on the right vertical axis. For MuSt-NSGA-III,

the final population from the previous stages and the current population are combined to find associated active points, and HV is computed from them. Thus, a solution evaluation-wise HV variation is expected to be mostly monotonic, as can be seen in Figure 7. It can also be seen that very early on, NSGA-III is able to find 100 ND population members (shown by the dashed line). NSGA-III and MuSt-NSGA-III find a similar number of active RVs until about 10,000 SEs (until the end of Stage 2). This is not surprising because both these algorithms employ the same baseline NSGA-III until then, and Stage 1 is able to capture most of the active RVs, and Stage 2 does not add many new ARVs. However, due to the increase in total RVs using our proposed estimation procedure, the number of ARVs drastically increase in Iteration 1 of Stage 3 and some more in Iteration 2. Stage 1 was saturated with 38 ARVs (out of 100 RVs), and without these iteration-wise increases in RVs, as in NSGA-III shown with dotted line, the algorithm would not have increased the number of ARVs close to the desired number ($N = 100$). It is also clear that Iteration 2 of Stage 3 is not able to produce exactly 100 ARVs, but Line 24 to 27 of Algorithm 1 allows a combination of both iteration's points together to produce exactly 100 solutions at the end.

The HV variations reveal that NSGA-III and MuSt-NSGA-III perform almost in the same manner during Stages 1 and 2. However, MuSt-NSGA-III improves the HV by adding more ND points in both iterations of Stage 3, whereas NSGA-III's HV does not improve due to lack of denser active ND solutions in the population. But, if all ND solutions are used to compute HV, marked with 'Extended NSGA-III', a slight increase in HV from the HV computed with single active ND point, marked with 'NSGA-III' is observed. The HV for 'MuSt-NSGA-III' is better due to a more uniformly distributed set of increasingly denser points obtained by our multi-stage framework.

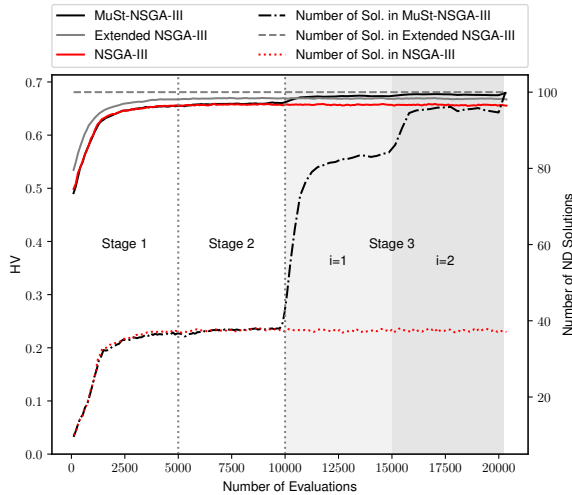


Figure 7: Average hypervolume (solid lines) and the number of active RVs (dotted lines) obtained using NSGA-III calculated from two sets – closest to each RV and all ND solutions – and MuSt-NSGA-III ND set for the crashworthiness problem.

The above deeper analysis on HV (and the same on other

metrics shown in the supplementary document) of the behavior of the proposed MuSt-NSGA-III compared to the original NSGA-III makes a better understanding of the working principle of the proposed framework. Furthermore, it provides us with confidence to use the framework to solve other problems and integrate with other EMaO algorithms.

B. MuSt-NSGA-III Applied to Multi-objective Test Problems

Next, we apply MuSt-NSGA-III to a number of multi-objective test problems. But before that, we briefly describe the quality indicators that are used in this study. For more detailed information about the indicators, the original references provided below can be referred. The quality indicators (with more information in the supplementary document) have been chosen taking into account their ability in measuring convergence and uniformity in their distribution.

- We use well-known *hypervolume* metric with nadir point as the reference point. Higher the HV value, better is the set in terms of convergence and diversity.
- *MIP-DoM* [28] is a binary quality indicator that tries to express Pareto front features, such as, proximity, cardinality, spread, and uniformity. It considers the minimum overall movement of points in one set needed to dominate each and every member of the other set. If $MIP-DoM(A,B) < MIP-DoM(B,A)$, then set A is better than B and vice versa.
- To measure uniformity of ND solutions, we use several metrics. As a naive measure, we use the mean and standard deviation of the k -th nearest neighbor ($k = \lfloor \sqrt{M} - 1 \rfloor$) values [17] of ND solutions. A large value of mean and a small value of standard deviation are desired.
- *Spacing* (SP), proposed in [29], measures how uniform the ND solutions are according to the standard deviation of Euclidean distance from their nearest neighbors. Smaller spacing values indicate better distribution.
- *Uniform Distribution* (UD), proposed in [30], is a niching-based unary quality indicator that measures the standard deviation in niche count values of solutions. Larger the metric, more uniform is the distribution.
- Finally, we use the *Evenness* (ξ) metric [31] as an indicator to measure gaps, if any, by a coefficient of variance measure of two critical neighbor distances of all ND points. Smaller the value, the better is the evenness.

Using the above quality indicators, the computational experiments are designed to assess the performance of the MuSt-NSGA-III for a comparison to its baseline version on 12 multi-objective problems. ZDT3 and MaF07 present disjoint Pareto fronts having natural gaps. Crashworthiness and car side-impact problems [2] are practical problems and involve non-linearity and non-convexity features of the Pareto front. Other problems from the MaF benchmark suite (designed for CEC'2018 MaOP competition) are selected: MaF01, MaF10, MaF11, and MaF12, which present linear, non-separable, biased, convex/disconnected, and concave features. A few DAS-CMOP family problems having multiple disjointed clusters in their Pareto fronts [32] are also selected. For DAS-CMOP7 to DAS-CMOP9 problems, the difficulty triplets are set to

Table I: Results of baseline NSGA-III with complete ND population ($N = 100$) and MuSt-NSGA-III. 7 performance metrics statistically compare both algorithms over 50 runs. Symbols \uparrow and \downarrow indicate better metric values for large and small values, respectively. MIP-DoMs from top to bottom are for pairs (NSGA-III, MuSt-NSGA-III) and (MuSt-NSGA-III, NSGA-III).

Problem	Algorithm	M	N	HV		MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	ξ \downarrow	#RVs Stage 3, $i=2$
				mean \uparrow	std dev \downarrow		mean \uparrow	std dev \downarrow				
ZDT3	NSGA-III	2	100	0.4806	0.0038	0.1086	0.0071	0.0048	0.0068	0.6983	1.6641	-
	MuSt-NSGA-III		100	0.4816	0.0006	0.1594	0.0083	0.0028	0.0040	0.9634	1.6300	177
DTLZ2	NSGA-III	3	100	0.7917	0.0000	0.7201	0.1049	0.0034	0.0052	1.0000	0.0858	-
	MuSt-NSGA-III		100	0.7916	0.0002	0.7253	0.1031	0.0037	0.0061	0.9975	0.0902	100
MaF01	NSGA-III	3	100	0.0146	0.0002	1.7711	0.0175	0.0147	0.0227	0.5329	0.8066	-
	MuSt-NSGA-III		100	0.0162	0.0001	1.7393	0.0439	0.0073	0.0102	0.9748	0.2392	343
MaF07	NSGA-III	3	100	0.2503	0.0150	<i>1.4809</i>	0.0254	0.0179	0.0279	0.6400	0.9236	-
	MuSt-NSGA-III		100	0.2551	0.0092	1.4508	0.0498	0.0133	0.0190	0.8781	0.6991	222
MaF10	NSGA-III	3	100	0.5153	0.0893	1.0939	0.0290	0.0171	0.0266	0.5922	0.8987	-
	MuSt-NSGA-III		100	0.6177	0.0289	<i>1.2600</i>	0.0500	0.0141	0.0218	0.9359	0.7319	187
MaF11	NSGA-III	3	100	0.7273	0.0635	<i>1.5877</i>	0.0855	0.0198	0.0282	0.9030	0.2285	-
	MuSt-NSGA-III		100	0.7371	0.0614	1.5133	0.0868	0.0146	0.0204	0.9868	0.2098	106
MaF12	NSGA-III	3	100	0.6640	0.0019	2.2017	0.0837	0.0192	0.0272	0.9049	0.2311	-
	MuSt-NSGA-III		100	0.6646	0.0020	<i>2.2286</i>	0.0858	0.0141	0.0194	0.9951	0.2119	102
Carside impact	NSGA-III	3	100	0.2288	0.0005	1.9857	0.0506	0.0334	0.0498	0.6919	0.4876	-
	MuSt-NSGA-III		100	0.2303	0.0009	<i>2.0307</i>	0.0733	0.0118	0.0163	0.9671	0.2117	143
Crashworthiness**	NSGA-III	3	100	0.1102	0.0027	1.4844	0.0195	0.0166	0.0260	0.5424	0.9322	-
	MuSt-NSGA-III		100	0.1140	0.0017	1.0885	0.0404	0.0112	0.0169	0.7971	0.6208	325
	Archive-NSGA-III		100	0.1117	0.0018	1.3647	0.0263	0.0284	0.0418	0.3648	0.9981	-
DAS-CMOP7	NSGA-III	3	100	0.7331	0.0074	0.9922	0.0778	0.0244	0.0359	0.8402	0.3194	-
	MuSt-NSGA-III		100	0.7330	0.0060	0.9918	0.0794	0.0158	0.0237	1.0000	0.2881	109
DAS-CMOP8	NSGA-III	3	100	0.7148	0.0065	0.9917	0.0837	0.0199	0.0292	0.9017	0.2955	-
	MuSt-NSGA-III		100	0.7156	0.0027	0.9963	0.0834	0.0161	0.0239	0.9975	0.2974	111
DAS-CMOP9	NSGA-III	3	100	0.2744	0.0513	0.4712	0.0139	0.0135	0.0209	0.4625	0.9744	-
	MuSt-NSGA-III		100	0.2912	0.0509	<i>0.5330</i>	0.0194	0.0122	0.0183	0.4766	0.8128	396

The Wilcoxon signed-rank test with a significance level of 0.05 was applied to perform a statistical comparison between the approaches. Thus, the data in italics means that it is not possible to detect differences between NSGA-III and MuSt-NSGA-III. Data in bold indicates the best value between the approaches;

** MIP-DoM from top to bottom for pairs: (NSGA-III, MuSt-NSGA-III), (MuSt-NSGA-III, Archive NSGA-III), (Archive NSGA-III, MuSt-NSGA-III).

(0.5,0.5,0.5). Results on many other problems are shown in the supplementary document, including DTLZ5.

Table I presents the average quality indicator values obtained by NSGA-III and MuSt-NSGA-III over 50 runs of both algorithms on 12 problems. In all cases, N is set to 100. In all tests, MuSt-NSGA-III is executed with $T = 20,000$ for all problems, except for DAS problems, for which we use $T = 100,000$. For all problems, $\gamma = 1/2$ is chosen. The Wilcoxon signed-rank test with a significance level of 0.05 is applied to assess the statistical significance of the obtained values. Data in italics means that the p-value is more than 0.05, and it is not possible to reject the null hypothesis that the distribution of the differences between NSGA-III and MuSt-NSGA-III is symmetric at about zero. Data in bold indicates the best value between the approaches.

It is important to observe that in some problems, the base algorithm is not able to generate N ND solutions, mainly due to the non-existence of a Pareto solution for every chosen RV. To make a fair comparison with our multi-stage framework, we use an extended ND solution set with population members that are beyond the closest point to each RV. This enables to have a total of N ND points, but they need not have a good distribution.

Analyzing Table I, some important features of the proposed multi-stage framework can be highlighted. First, we argue that the MuSt-EMaO maintains the inner characteristics of

the baseline reference-vector-based algorithm. If the problem allows finding all N solutions in Stage 1, like in ZDT3 and DTLZ2 problems, MuSt-NSGA-III and the baseline NSGA-III perform precisely the same way. In three-objective DTLZ2, there is no hole or clusters in the Pareto front. Then, only Stage 1 of the multi-stage framework is executed for the entire T SEs. Thus, the table shows that all quality indicators present no statistical difference among the results. This degeneracy of our proposed algorithm to the baseline algorithm was a desired feature.

Second, observing the hypervolume values, it is possible to observe that the MuSt-NSGA-III always produce a larger hypervolume value than its baseline version, except for DAS-CMOP7. However, the statistical test indicates that there is no statistical difference between the algorithms on this problem as well. However, MuSt-NSGA-III is statistically better on MAF01, MaF07, car side-impact, crashworthiness, DAS-CMOP8 and DAS-COMP9 problems. For the MIP-DoM metric, except in ZDT3, MuSt-NSGA-III is statistically better or equivalent to NSGA-III on all 11 problems. All other indicators capture the reliability and uniformity of the final ND solution set. MuSt-NSGA-III clearly performs better than the baseline algorithm considering all the indicators. The only exceptions are the k -nearest neighbor distance mean and evenness indicators for DAS-CMOP8 problem; however, the tests indicate no statistical difference between their values.

Third, to demonstrate the reproducibility aspect, the standard deviation on HV shows that, in most problems, the variation of HV over 50 independent runs has a much smaller standard deviation compared to NSGA-III, meaning that the multiple MuSt-NSGA-III runs produce almost a similar set of 100 ND points. The coefficient of variation (standard deviation of HV divided by the mean HV) varies between 0.0000 and 0.1748 with a median of 0.0072, while NSGA-III values vary between 0.0001 to 0.187 with a median of 0.0119.

Fourth, the last column of the table presents the number of RVs needed in Iteration 2 of Stage 3 of MuSt-NSGA-III to obtain $N = 100$ ND solutions. As it can be seen, a maximum of $4N$ RVs are needed to obtain N ND points for these problems. Interestingly, for the DTLZ5 problem shown in the supplementary document, about $19N$ RVs are needed. These numbers are difficult to predict at the beginning for a problem and our proposed multi-stage framework with two iterations adaptively estimates the requisite number of RVs needed to find the desired number of ND points.

The crashworthiness result in Table I shows that the archive-based NSGA-II method [9], [10] may not produce a better distribution of solutions, despite choosing the desired number of solutions from a large archive of generation-wise collected ND solutions. To achieve a well-distributed and well-converged set of solutions, algorithms need to focus their search in multiple diverse regions in a coordinated manner parallelly at every generation and allow current good solutions to compete with newly created solutions, rather than expecting a few sampled ND solutions, passively collected from multiple generations, to automatically produce a good distribution. However, a more comprehensive investigation is warranted as a future study.

C. Multi-stage Framework in Other EMO Algorithms

The computational results so far were achieved using the NSGA-III as the baseline reference-vector based algorithm. However, the multi-stage framework can also be integrated with any other reference-based EMO as long as a requirement is met. Since in the multi-stage framework, the initial population from Stage 1 must be used in Stage 2, and in Stage 3, the baseline algorithm must allow the user to set an initial population instead of always a random one. Three additional reference-based algorithms are included in this study: MOEA/D [3], C-TAEA [19], and CLIA [22]. The algorithms have been implemented using pymoo and PlatEMO frameworks, [26], [33]. The performance indicators – HV, k-neighbors distance mean, k-neighbors distance standard deviation, Spacing, UD, and Evenness – are used as quality indicators. The Wilcoxon signed-rank test with a significance level of 0.05 was employed to assess the statistical significance of the values. Data in italics means the p-value is more than 0.05, and it is not possible to detect difference between the values. Data in bold indicates the best value between the multi-stage framework and its baseline.

Table II presents seven quality indicators calculated for five EMO algorithms with and without the multi-stage framework on three-objective MaF01 problem (similar to the inverted DTLZ1 problem). As described in Section II, CLIA [22] is an

algorithm which combines two interactive processes: Cascade Clustering (CC) and Reference Point Incremental Learning. CC uses reference vectors to create and sort the solutions, while the SVM model adjusts these references throughout the evolutionary process. Therefore, we want to apply CLIA as a baseline algorithm in MuSt-EMaO in two different contexts. Initially, CLIA is applied as the reference-based algorithm in MuSt-EMaO just like other EMO algorithms (NSGA-III, MOEA/D, and C-TAEA). Thereafter, we integrate our multi-stage framework coupled with the Cascade Clustering (CC) of CLIA. The rationale behind this idea is that although the SVM model in the incremental learning phase improves the uniformity of adjusting the reference vectors, it presents a quadratic computational cost related to the number of objectives. We argue that the proposed multi-stage framework can improve the reliability and uniformity, when coupled with CC. In this context, to assess the effectiveness of the reference vector adjustments, we turn off the incremental learning approach from CLIA and integrate our MuSt-EMaO framework with the Cascade Clustering process (MuSt-CC). Interestingly, in all performance measures, the respective multi-stage version has obtained statistically better or equivalent performance.

Table II: MaF01 problem set with $M = 3$. The experiment involves 50 trials, and each algorithm is run with $T = 20,000$. Some selected quality indicators are tabulated.

Algorithm	N	HV \uparrow	MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	$\xi \downarrow$
				mean \uparrow	std dev \downarrow			
NSGA-III	100	0.0146	1.7711	0.0175	0.0147	0.0227	0.5329	0.8066
MuSt-NSGA-III	100	0.0162	1.7393	0.0439	0.0073	0.0102	0.9748	0.2392
MOEA/D	97	0.0129	1.7946	0.0261	0.0269	0.0409	0.4709	0.8589
MuSt-MOEA/D	100	0.0161	1.7473	0.0424	0.0085	0.0122	0.9282	0.2808
C-TAEA	100	0.0153	1.7449	0.0289	0.0130	0.0205	0.5552	0.5800
MuSt-C-TAEA	100	0.0162	1.7770	0.0450	0.0079	0.0108	0.9646	0.2233
CLIA	100	0.0164	1.8025	0.0345	0.0128	0.0198	0.7148	0.4338
MuSt-CLIA	100	0.0164	1.7332	0.0450	0.0085	0.0110	0.9687	0.2318
CLIA	100	<i>0.0163</i>	1.8537	0.0344	0.0128	0.0199	0.7123	0.4323
MuSt-CC	100	0.0164	1.6558	0.0438	0.0087	0.0120	0.9798	0.2466

To show the distribution of obtained points, we plot MOEA/D obtained ND points in the left plot of Figure 8. These points come from the median HV run. The ND points from MuSt-MOEA/D are shown in the right plot. Clearly, the multi-stage framework is able to produce a much better distribution of points on the MaF01 problem. Note from Table II that MOEA/D could not produce 100 ND points. We compute the quality indicators if at least 95% of prescribed ND points are found and ascertain that any fewer number of ND solutions will not make the comparison between two vastly unequal sets fair. The experiment is extended to two more problems: MaF07 and MaF11 in Tables III, and IV, respectively. It is important to note that MOEA/D can find only 85 ND solutions with 100 RVs on the entire unit simplex for MaF07 and MaF11 problems. The final population of MOEA/D does not have any more ND points to increase the final number of ND points to the desired number (100, in our case). Thus, we do not compare MOEA/D's performance with MOEA/D's multi-stage version.

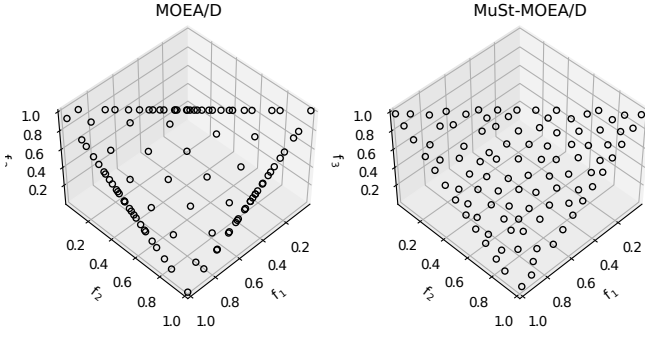


Figure 8: MOEA/D and MuSt-MOEA/D ND solutions from the median HV run of 50 runs for MaF01 problem.

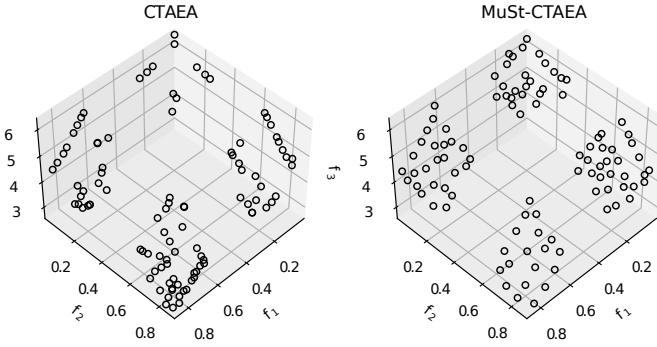


Figure 9: C-TAEA and MuSt-C-TAEA ND solutions from the median HV run of 50 runs for MaF07 problem.

Analyzing the results from MaF01 problem, presented in Table II, a similar pattern, observed in Table I, also appears here. There is a significant statistical difference favoring the multi-stage framework when the uniformity indicators are computed. However, for MuSt-CLIA and MuSt-CC, regarding the HV indicator, there is no statistical difference between the

Table III: MaF07 problem set with $M = 3$. The experiment involves 50 trials, and each algorithm has set with $T = 20,000$. Some quality indicators are presented. MOEA/D algorithm is not able to generate 100 solutions; hence, we do not compute and compare the quality indicators.

Algorithm	N	HV \uparrow	MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	ξ \downarrow
				mean \uparrow	std dev \downarrow			
NSGA-III	100	0.2503	1.4809	0.0254	0.0179	0.0279	0.6400	0.9236
MuSt-NSGA-III	100	0.2551	1.4508	0.0498	0.0133	0.0190	0.8781	0.6991
MOEA/D	85	-	-	-	-	-	-	-
MuSt-MOEA/D	100	0.2205	1.1091	0.0374	0.0148	0.0229	0.7366	0.7080
C-TAEA	100	0.2603	1.3762	0.0272	0.0224	0.0335	0.4616	1.1244
MuSt-C-TAEA	100	0.2702	1.3036	0.0467	0.0143	0.0211	0.8916	0.7503
CLIA	100	0.2076	1.5099	0.0314	0.0221	0.0341	0.4909	0.9842
MuSt-CLIA	100	0.2218	1.3957	0.0441	0.0110	0.0164	0.8425	0.7239
CLIA	100	0.2066	1.6067	0.0310	0.0225	0.0347	0.5065	0.9870
MuSt-CC	100	0.2162	1.2410	0.0436	0.0110	0.0163	0.8264	0.7390

baseline algorithm and the multi-stage framework. Figure 9 shows the ND points from C-TAEA and MuSt-C-TAEA. A better distribution of ND points can be observed with the multi-stage framework.

In Table IV, the results on MaF11 problem are presented. In this case, there is a statistical difference in two CLIA versions related to HV. However, considering the k -neighbor distance mean, the pair-wise values are very close, and there is no statistical difference. Figure 10 shows the CLIA and MuSt-CLIA solution sets.

Table IV: MaF11 problem set with $M = 3$. The experiment involves 50 trials, and each algorithm has set with $T = 20,000$. Some quality indicators are presented. MOEA/D algorithm is not able to generate 100 solutions; hence, we do not compute and compare the quality indicators.

Algorithm	N	HV \uparrow	MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	ξ \downarrow
				mean \uparrow	std dev \downarrow			
NSGA-III	100	0.7273	1.5877	0.0855	0.0198	0.0282	0.9030	0.2285
MuSt-NSGA-III	100	0.7371	1.5733	0.0868	0.0146	0.0204	0.9868	0.2098
MOEA/D	85	-	-	-	-	-	-	-
MuSt-MOEA/D	100	0.3593	0.7533	0.0535	0.0103	0.0162	0.6854	0.5377
C-TAEA	100	0.5782	1.3290	0.0608	0.0297	0.0446	0.7901	0.6010
MuSt-C-TAEA	100	0.7500	1.3526	0.0675	0.0174	0.0264	0.9696	0.4102
CLIA	100	0.7389	1.5326	0.0727	0.0264	0.0408	0.9729	0.4795
MuSt-CLIA	100	0.7406	1.5397	0.0737	0.0195	0.0303	0.9803	0.3794
CLIA	100	0.7398	1.5356	0.0723	0.0258	0.0399	0.9819	0.4827
MuSt-CC	100	0.7421	1.5423	0.0743	0.0188	0.0293	0.9893	0.3746

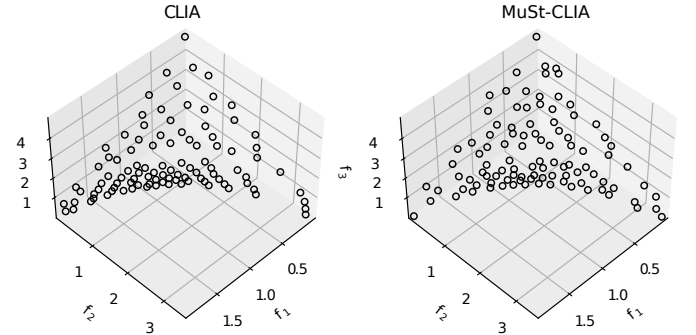


Figure 10: CLIA and MuSt-CLIA ND solutions from the median HV run of 50 runs for MaF11 problem.

In the supplementary document, more results on other problems, such as, ZDT3, DTLZ5, C2-DTLZ2, MW14, more MaF and DAS-CMOP problems, are provided.

D. MuSt-NSGA-III Applied to Many-objective Problems

Next, to analyze the effect of the number of objective functions on the proposed framework, MaF07 and C2-DTLZ2 are considered. For both problems, the number of objective functions is set to $M = 3, 5, 8$, and 10 using $N = 100, 200, 400$, and 600, and $T = 20,000, 40,000, 80,000$, and 120,000, respectively. Only NSGA-III and MuSt-NSGA-III are used in

this study. Table V shows the average of performance measures over 50 runs on MaF07 problem. Considering HV and MIP-

Table V: Number of objective functions is set to $M = 3, 5, 8$, and 10 , with $N = 100, 200, 400$, and 600 , and, $T = 20,000, 40,000, 80,000$, and $120,000$, respectively, for MaF07 problem. Average of quality indicators for 50 runs are presented.

Algorithm	M	N	HV \uparrow	MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	ξ \downarrow
					mean \uparrow	std dev \downarrow			
NSGA-III	3	100	0.2503	1.4809	0.0254	0.0179	0.0279	0.6400	0.9236
MuSt-NSGA-III		100	0.2551	1.4508	0.0498	0.0133	0.0190	0.8781	0.6991
NSGA-III	5	200	0.2284	3.3726	0.0970	0.0606	0.0760	0.6224	0.7188
MuSt-NSGA-III		200	0.2300	3.0803	0.1323	0.0392	0.0585	0.9212	0.4425
NSGA-III	8	400	0.0528	5.4310	0.1489	0.0700	0.1234	0.6698	0.6098
MuSt-NSGA-III		400	0.0526	5.5817	0.1731	0.0518	0.0864	0.8119	0.4571
NSGA-III	10	600	0.0323	5.5446	0.1982	0.0662	0.1382	0.6721	0.5454
MuSt-NSGA-III		600	0.0319	5.6760	0.2167	0.0359	0.0672	0.9500	0.3827

DoM, the results of MuSt-NSGA-III for $M = 8$ and 10 are not statistical different from NSGA-III. All other results bring evidence favoring the multi-stage framework. In Figure 11, a comparison of final ND solutions for eight-objective MaF07 problem from NSGA-III and MuSt-NSGA-III are shown on parallel coordinate plots (PCPs). Visually, it is possible to observe a better regularity in the position of the lines for MuSt-NSGA-III, particularly for objective functions f_3 , f_7 , and f_8 .

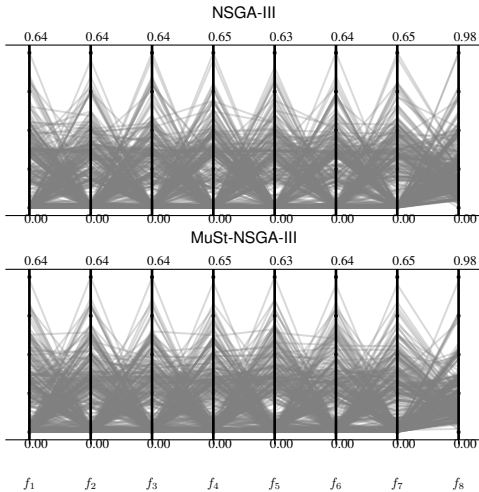


Figure 11: Parallel coordinate plots for NSGA-III and MuSt-NSGA-III for the MaF07 problem with $M = 8$. 50 runs are made and the presented results come from the median HV run.

Table VI shows the results for C2-DTLZ2. There is no statistical difference for HV, MIP-DoM, and k -neighbors distance mean indicators, as the number of objectives increase. All other indicators show a more uniform distribution for MuSt-NSGA-III, as observed for the MaF07 problem.

V. PARAMETRIC STUDIES

Algorithm 1 shows that four input are needed to execute the MuSt-EMaO framework. Of them, the EMaO algorithm,

Table VI: Number of objectives is set to $M = 3, 5, 8$, and 10 , $N = 100, 200, 400$, and 600 , and, $T = 20,000, 40,000, 80,000$, and $120,000$, respectively, for C2-DTLZ2 problem. Average of quality indicators for 50 runs are presented.

Algorithm	M	N	HV \uparrow	MIP-DoM \downarrow	k -neighbors distance		SP \downarrow	UD \uparrow	ξ \downarrow
					mean \uparrow	std dev \downarrow			
NSGA-III	3	100	0.7379	0.9331	0.0573	0.0365	0.0540	0.6965	0.4780
MuSt-NSGA-III		100	0.7476	0.9221	0.0710	0.0140	0.0199	1.0000	0.3506
NSGA-III	5	200	0.9603	1.0005	0.1647	0.0459	0.0812	0.7818	0.4874
MuSt-NSGA-III		200	0.9607	1.0004	0.1695	0.0268	0.0340	1.0000	0.4197
NSGA-III	8	400	0.9812	1.0007	0.2036	0.0732	0.1199	0.7741	0.5038
MuSt-NSGA-III		400	0.9811	1.0013	0.2042	0.0464	0.0563	0.9969	0.4798
NSGA-III	10	600	0.9612	1.0002	0.2255	0.0765	0.1247	0.7921	0.4951
MuSt-NSGA-III		600	0.9612	1.0007	0.2314	0.0698	0.0815	0.0849	0.4437

desired number of ND solutions, and the overall number of SEs are supplied by the user and are not algorithm parameters which can be set in any way by the developer. The above section has shown how the proposed MuSt-EMaO framework can be integrated with a number of EMaO algorithms to improve their performance in terms of finding a well-distributed set of a pre-specified number of desired ND point reliably. The use of three stages adaptively determines if the first stage will continue until all pre-specified number of solution evaluations are completed, or if the algorithm has to move to Stage 2 to ensure if any active RVs was left out in Stage 1, or if the algorithm needs to produce denser points to come up the desired number of ND points. The P^{seed} is the initial supplied random population. The proposed multi-stage framework requires only one parameter: γ , which determines the proportion of total T SEs to be dedicated to Stage 3, while the remaining SEs are equally distributed between Stages 1 and 2. The number of desired ND points (N) is truly not a parameter for the algorithm, but is a desired number that the user needs to provide. But, in the next section, we show that the MuSt-NSGA-III works well with different N values.

A. Effect of Number of Desired Solutions

The previous experiments have attempted to find $N = 100$ ND solutions. Here, we investigate the effect of number of desired solutions (N) on the performance of our proposed multi-stage framework. Table VII presents the results on the MaF07 problem for three different N values: 50, 100 and 200. We use a linearly proportionate SEs: 10,000, 20,000 and 40,000 for the three scenarios, respectively. We have also set $\gamma = 0.5$ in this study. It is clear from the table that MuSt-NSGA-III is able to produce a more uniformly distributed set of Pareto solutions compared to NSGA-III for the same number of respective SEs over 50 runs. In most cases, MuSt-NSGA-III is statistically superior to NSGA-III.

Next, we consider the crashworthiness problem and show the results for $N = 50, 100$ and 200 as well in Table VIII. For this problem, MuSt-NSGA-III performs statistically superior to NSGA-III over 50 independent runs, clearly indicating the advantage of the multi-stage framework.

Table VII: Results using MaF07. Number of solutions are set to $N = 50, 100, 200$ and $M = 3$, using $T = 10,000, 20,000$, and $40,000$, respectively. Average of quality indicators for 50 runs are presented.

Algorithm	N	HV \uparrow	MIP-DoM \downarrow	k-neighbors distance		SP \downarrow	UD \uparrow	$\xi \downarrow$
				mean \uparrow	std dev \downarrow			
NSGA-III	50	0.2416	1.1875	0.0446	0.0367	0.0557	0.6489	0.7807
MuSt-NSGA-III		0.2421	1.0841	0.0692	0.0229	0.0337	0.7988	0.5837
NSGA-III	100	0.2503	1.4809	0.0254	0.0179	0.0279	0.6400	0.9236
MuSt-NSGA-III		0.2551	1.4508	0.0498	0.0133	0.0190	0.8781	0.6991
NSGA-III	200	0.2546	1.5469	0.0170	0.0118	0.0183	0.5922	1.1038
MuSt-NSGA-III		0.2565	1.4665	0.0335	0.0078	0.0113	0.8990	0.8840

Table VIII: Results using Crashworthiness. Number of solutions are set to $N = 50, 100, 200$ and $M = 3$, using $T = 10,000, 20,000$, and $40,000$, respectively. Average of quality indicators for 50 runs are presented.

Algorithm	N	HV \uparrow	MIP-DoM \downarrow	k-neighbors distance		SP \downarrow	UD \uparrow	$\xi \downarrow$
				mean \uparrow	std dev \downarrow			
NSGA-III	50	0.1091	1.4982	0.0275	0.0246	0.0388	0.5625	0.9016
MuSt-NSGA-III		0.1132	1.3401	0.0592	0.0187	0.0283	0.7846	0.5409
NSGA-III	100	0.1101	1.4844	0.0195	0.0166	0.0260	0.5440	0.9333
MuSt-NSGA-III		0.1139	1.0682	0.0403	0.0112	0.0169	0.7991	0.6205
NSGA-III	200	0.1135	1.4164	0.0138	0.0110	0.0172	0.5268	1.0229
MuSt-NSGA-III		0.1162	0.9642	0.0281	0.0074	0.0108	0.8226	0.7266

B. Effect of the Solution Evaluation Budget on Each Stage

Next, we perform a parametric study with the γ parameter, that decides the individual SEs for each stage. We assume that $T_1 = T_2$, providing two equal chances for the EMaO algorithm to reliably find a set of ND solutions by first focusing on the entire Pareto front in Stage 1 and then focusing on the IRVs in Stage 2, that could not be covered in Stage 2. Using T as the total number of solution evaluations, we define $T_3 = \gamma T$ and $T_1 = T_2 = \frac{1-\gamma}{2}T$. Based on this setting, three different configuration tests are designed with $\gamma = 1/3, 1/2$, and $2/3$. Therefore, for example, considering 10,000 total solution evaluations, a configuration set with $\gamma = 1/2$ will set $T_1 = 2,500, T_2 = 2,500$, and $T_3 = 5,000$. Note that Stage 3 involves two iterations.

Our goal is to determine the statistically best setting considering all quality indicators and all problems used in this study. However, the usual multiple pairwise comparisons, such as the ones done in the Wilcoxon test, suffers from the multiplicative effect to control the family-wise error rate [34]. We adopt two non-parametric statistical tests that are able to make multiple comparisons. The first test, the Friedman test, ranks the algorithm's performance for each problem and computes the average performance among problems. It uses these rankings to test the null hypothesis, the equivalence of median metric value for different γ values, versus the alternative hypothesis that two or more medians are different. The second test, the Quade test [34], considers the difference among the problems. It calculates the range of the problems as the maximum differences between the samples and ranks

the k problem ranges. Then, these ranks are assigned to the problems and represent a weight to the ranks obtained by the Friedman method. Finally, it uses the same null and alternative hypotheses as in the Friedman test.

Table IX shows the Friedman test results using the HV values in the top part of the table. The mean HV value and the ranking of each γ are shown for each problem in brackets. The mean HV-based ranking is tabulated in the last row. While $\gamma = 1/2$ performs the best, the other two values are also statistically equivalent. Similar tables are also produced for all other indicators. The complete Friedman table for each quality indicator is shown in the supplementary document. Here, we show only the final ranking for the other indicators.

Table IX: Friedman ranks for HV mean for all config sets using $\gamma = 1/3, 1/2$, and $2/3$. For k-neighbors distance mean, k-neighbors distance std dev, SP, UD, and ξ quality indicators, only the final ranking are shown for brevity.

Problem sets	Config sets using γ		
	1/3	1/2	2/3
HV Friedman test rank			
ZDT3	0.4817 (1)	0.4816 (2)	0.4815 (3)
MaF01	0.7275 (2)	0.7278 (1)	0.7272 (3)
MaF07	0.6813 (3)	0.6819 (2)	0.6823 (1)
MaF10	0.6792 (1)	0.6330 (2)	0.5543 (3)
MaF11	0.7386 (3)	0.7422 (1)	0.7392 (2)
MaF12	0.7700 (3)	0.7709 (1)	0.7701 (2)
Crashworthiness	0.7447 (3)	0.7460 (1)	0.7458 (2)
Carside impact	0.6823 (2)	0.6823 (3)	0.6826 (1)
DAS-CMOP7	0.7808 (1)	0.7794 (2)	0.7755 (3)
DAS-CMOP8	0.7884 (1)	0.7880 (2)	0.7845 (3)
DAS-CMOP9	0.5108 (1)	0.5015 (2)	0.4944 (3)
Ranking	1.90	1.72	2.36
k-neighbors distance mean Friedman test			
Ranking	2.18	1.81	2.00
k-neighbors distance std dev Friedman test			
Ranking	2.72	1.72	1.54
Spacing Friedman test			
Ranking	2.90	1.54	1.54
UD Friedman test			
Ranking	2.27	1.86	1.86
ξ Friedman test			
Ranking	2.45	1.64	1.90

The numbers indicate that it is not possible to reject the null hypothesis, concluding that there is an equivalence among all three γ values for the hypervolume metric. We conduct the same test using the Quade ranking test, and even considering the different problem difficulties, it is again impossible to reject the null hypothesis. Therefore, the experiments present no statistical difference related to this quality indicator. The same phenomenon happens to k-neighbors distance mean, UD, and Evenness, in which it is not possible to observe statistical differences among different γ values. Contrarily, k-neighbors distance std-dev and spacing have produced slightly different conclusions. Results indicate statistically significant differences among γ values, preferring $\gamma = 1/2$ and $2/3$ over $1/3$.

The main drawback of the Friedman and Quade tests is that they can only detect significant differences over multiple comparisons. A general recommendation approach is: if the null hypothesis is rejected, we should proceed with a post-hoc procedure to characterize these differences. Then, applying a

Table X: HV Shaffer’s adjusted p-values for tests considering multiple comparisons among all methods.

Hypothesis	Shaffer
k-neighbors distance std dev	
Config test $\gamma = 1/3$ vs Config test $\gamma = 2/3$	0.0426
Config test $\gamma = 1/3$ vs Config test $\gamma = 1/2$	0.0426
Config test $\gamma = 1/2$ vs Config test $\gamma = 2/3$	0.7060
Spacing	
Config test $\gamma = 1/3$ vs Config test $\gamma = 2/3$	0.0126
Config test $\gamma = 1/3$ vs Config test $\gamma = 1/2$	0.0200
Config test $\gamma = 1/2$ vs Config test $\gamma = 2/3$	0.5930

post-hoc test, we can obtain a p-value that determines the degree of rejection of each pairwise hypothesis. Therefore, we conduct a post-hoc analysis using the adjusted p-values calculated by Shaffer’s procedure [35]. Table X makes a pairwise comparison of three γ configurations for k-neighbors distance standard deviation and spacing indicators on all problems. Considering a significance level of 0.05, it can be concluded that there is always a difference in performance with $\gamma = 1/3$ compared to $1/2$ or $2/3$. However, the comparisons between $\gamma = 1/2$ and $\gamma = 2/3$ do not present a statistical difference for both quality indicators. It leads us to conclude that a budget with a number of solution evaluations greater than $\gamma = 1/2$ performs better. Since the final adjustments with increased number of RVs in Stage 3 determines the quality of the overall framework, the parametric study finds that allocating more SEs at Stage 3 is a better strategy.

VI. CONCLUSIONS

Evolutionary algorithms are stochastic; hence they produce different sets of solutions in different runs. To make evolutionary multi-objective optimization algorithms reproduce a desired number of non-dominated (ND) solutions reliably, this paper has proposed a three-stage reference vector (RV) based framework that takes multiple attempts to focus on various key aspects of multi-objective problem solving. The first stage attempts to find a skeleton of active RVs that contain at least one associated ND solution. The second stage attempts to focus its search on inactive RVs to make sure no true associated ND solutions are left out. Having identified all active RVs, the third stage focuses on searching the desired number of ND points in two iterations by progressively increasing the number of RVs. While each of the three stages has their individual role, if an earlier stage is able to find the desired number of ND points, the algorithm consumes the remaining budget of solution evaluations to make the convergence and distribution better without truly moving to the latter stages, thereby degenerating to a single-stage original algorithm, if adequate. A later stage starts from the final populations of the previous stages to make the overall algorithm more efficient.

The working principle of the multi-stage NSGA-III has been demonstrated on 12 two-objective and three-objective problems, clearly showing its superiority in terms of achieving a better distribution of points measured through seven

performance indicators. The multi-stage approach has been integrated with MOEA/D, C-TAEA, and CLIA algorithms as well, with an improved performance in each case. Thereafter, the multi-stage NSGA-III is applied to many-objective problems having 5-10 objective problems. Better performance has been reported compared to the original NSGA-III for the same number of solution evaluations.

A parametric study has been performed for the parameter defining the proportion of SEs dedicated to each stage. With a statistical analysis, it has been observed that the multi-stage approach works better with 50% or more total SEs assigned to two iterations of Stage 3. The multi-stage EMO has also shown to work well with different number of desired ND solutions, thereby making the overall framework generically applicable.

This extensive study for finding a reliable set of desired number of ND points can be extended in various ways. First, the multi-stage approach can be extended to algorithms that do not use RVs, by identifying inactive regions in Stage 2 and the number of uniformly distributed ND points using other indicators. Second, instead of prefixing a γ parameter to allocate a fixed number of SEs for each stage, indicators for stability in convergence and diversity measures can be used to terminate a stage. Care must be taken to ensure sufficient SEs are kept for later stages, despite stability conditions are not met for earlier stages. Third, effect of more iterations in Stage 3 can be tried. Fourth, comparisons of our approach with other shape-invariant EMO algorithms [36], [15] and archive-based ND solution accumulation strategies [9], [10] may be studied. Nevertheless, the multi-stage philosophy proposed here makes the framework practical by finding an exact number of desired uniformly distributed ND solutions reliably through repetitive searches in stages with varying focus.

REFERENCES

- [1] H. Ishibuchi, H. Masuda, and Y. Nojima, “Comparing solution sets of different size in evol. many-objective optimization,” in *2015 IEEE Congress on Evolutionary Computation*, 2015, pp. 2859–2866.
- [2] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [3] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *Trans. Evol. Comp.*, vol. 11, no. 6, p. 712–731, Dec. 2007.
- [4] P. V. Pellicer, M. I. Escudero, S. F. Alzueta, and K. Deb, “Gap finding and validation in evolutionary multi- and many-objective optimization,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, ser. GECCO ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 578–586.
- [5] K. Deb and J. Sundar, “Reference point based multi-objective optimization using evolutionary algorithms,” in *Proceedings of the 8th Annual Conf. on Genetic and Evol. Comput.*, ser. GECCO’06. New York, NY, USA: ACM Press, 2006, p. 635–642.
- [6] Y. Vesikar, K. Deb, and J. Blank, “Reference point based nsga-iii for preferred solutions,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1587–1594.
- [7] H. Seada, M. Abouhawwash, and K. Deb, “Multiphase balance of diversity and convergence in multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 503–513, 2019.
- [8] Y. Tian, C. He, R. Cheng, and X. Zhang, “A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 9, pp. 5880–5894, 2021.

- [9] L. Chen, L. M. Pang, H. Ishibuchi, and K. Shang, "Periodical generation update using an unbounded external archive for multi-objective optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1912–1920.
- [10] O. Schütze and C. Hernández, *Archiving Strategies for Evolutionary Multi-objective Optimization Algorithms*. Springer, 2021.
- [11] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.
- [12] P. Shukla and K. Deb, "Comparing classical generating methods with an evolutionary multi-objective optimization method," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-2005)*, 2005, pp. 311–325.
- [13] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer, 1999.
- [14] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [15] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.
- [16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [17] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou et al., Ed. Int. Center for Numerical Methods in Engg (CIMNE), 2001, pp. 95–100.
- [18] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [19] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.
- [20] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [21] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.
- [22] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. L. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Trans. on Evol. Comput.*, vol. 23, no. 4, pp. 572–586, 2019.
- [23] L. Ma, N. Li, Y. Guo, X. Wang, S. Yang, M. Huang, and H. Zhang, "Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.
- [24] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking multi- and many-objective evolutionary algorithms under two optimization scenarios," *IEEE Access*, vol. 5, pp. 19 597–19 619, 2017.
- [25] J. Blank, K. Deb, Y. Dhebar, S. Bandaru, and H. Seada, "Generating well-spaced points on a unit simplex for evolutionary many-objective optimization," *IEEE Trans. on Evol. Comput.*, vol. 25, no. 1, pp. 48–60, 2020.
- [26] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.
- [27] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. on Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.
- [28] C. L. d. V. Lopes, F. V. C. Martins, E. F. Wanner, and K. Deb, "Analyzing dominance move (MIP-DoM) indicator for multi-and many-objective optimization," *IEEE Trans. on Evolutionary Computation*, pp. 1–1, 2021.
- [29] J. R. Schott, "Fault tolerant design using single and multi-criteria genetic algorithms," Master's thesis, Boston, MA: MIT, 1995.
- [30] K. Tan, T. Lee, and E. Khor, "Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 2, 2001, pp. 979–986.
- [31] A. Messac and C. A. Mattson, "Normal constraint method with guarantee of even representation of complete pareto frontier," *AIAA journal*, vol. 42, no. 10, pp. 2101–2111, 2004.
- [32] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit," *Evol. Comput.*, vol. 28, no. 3, pp. 339–378, 2020.
- [33] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [34] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010, special Issue on Intelligent Distributed Information Systems.
- [35] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [36] J. G. Falcón-Cardona, C. A. Coello Coello, and M. Emmerich, "Crimoa: A Pareto-front shape invariant evolutionary multi-objective algorithm," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 307–318.



Kalyanmoy Deb is Fellow of IEEE, University Distinguished Professor and Koenig Endowed Chair Professor with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, Michigan, USA. He is largely known for his seminal research in evolutionary multi-criterion optimization. He has published over 600 international journal and conference research papers to date. He received IEEE EC Pioneer award, Infosys prize, ACM, IEEE and ASME Fellow recognition.



Cláudio Lúcio V. Lopes is a computer scientist and Ph.D. student at Computational and Mathematical Modelling program at CEFET-MG, Belo Horizonte, Brazil. He is a teaching assistant at PUC-MINAS, CTO, and founder at A3Data. His research interests include evolutionary computation, multi and many-objective optimization, quality indicators, machine learning and artificial intelligence.



Flávio V. Cruzeiro Martins received the B.S. degree in Computer Science (2007), M.Sc. degree (2009), and a Ph.D. degree (2012) in Electrical Engineering from Universidade Federal de Minas Gerais. He is an assistant professor at Computer Engineering Department in the Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, Brazil. His research interests include evolutionary algorithms, multi-objective and combinatorial optimization, and real-world optimization.



Elizabeth F. Wanner received the B.S. (1994) and M.Sc. (2002) degrees in mathematics, and Ph.D. degree in electrical engineering (2006) from the Universidade Federal de Minas Gerais. She is an Associate Professor in the Department of Computer Engineering, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, Brazil. Her research areas include evolutionary computation, global optimization, constraint handling, and multiobjective optimization.