

Enhanced *Innovized* Repair Operator for Evolutionary Multi- and Many-objective Optimization

Sukrit Mittal, Dhish Kumar Saxena, Kalyanmoy Deb, and Erik Goodman
COIN Report Number 2020020

Abstract—“*Innovization*” is a task of learning common relationships among some or all of the Pareto-optimal (PO) solutions in multi- and many-objective optimization problems. Recent studies have shown that a chronological sequence of non-dominated solutions obtained in consecutive iterations during an optimization run also possess salient patterns that can be used to learn problem features to help create new and improved solutions. In this paper, we propose a machine-learning- (ML-) assisted modeling approach that learns the modifications in design variables needed to advance population members towards the Pareto-optimal set. We then propose to use the resulting ML model as an additional *innovized* repair (IR2) operator to be applied on offspring solutions created by the usual genetic operators, as a novel mean of improving their convergence properties. In this paper, the well-known random forest (RF) method is used as the ML model and is integrated with various evolutionary multi- and many-objective optimization algorithms, including NSGA-II, NSGA-III, and MOEA/D. On several test problems ranging from two to five objectives, we demonstrate improvement in convergence behavior using the proposed IR2-RF operator. Since the operator does not demand any additional solution evaluations, instead using the history of gradual and progressive improvements in solutions over generations, the proposed ML-based optimization opens up a new direction of optimization algorithm development with advances in AI and ML approaches.

Index Terms—Multiobjective Optimization, Learning-based Optimization, Random Forest, Machine Learning, *Innovization*, Online *Innovization*

I. INTRODUCTION

OPTIMIZATION is an iterative process of arriving one or more optimal or near-optimal solutions for single- and multi-objective optimization problems using a point-based or a population-based approach. For multi- (having two or three objectives) and many-objective (having more than three objectives) optimization, the target is usually a set of trade-off Pareto-optimal (PO) solutions. Population-based methods become ideal for identifying a set of well-distributed and well-converged solutions in such problems. Since evolutionary algorithms use a population of solutions in each iteration, their extensions – evolutionary multi-objective optimization

(EMO) or evolutionary many-objective optimization (EMaO) algorithms – are extensively used for solving them [1, 2].

EMO/EMaO algorithms, including NSGA-II [3], NSGA-III [4, 5], MOEA/D [6], create new solutions (called offspring solutions) using bio-inspired genetic operators, like recombination and mutation operators. However, there are other algorithms, such as estimation of distribution algorithms (EDAs), that adopt a more involved approach and create new solutions by performing a sampling procedure from either a custom-built probabilistic model or other available probability models, such as Bayesian Networks, decision-trees, etc. EDAs, like BOA [7] and MONEDA [8], have shown that there is a potentially distinctive advantage in using the inter-variable relationships to generate new offspring solutions. This linkage-preserving advantage may be more important in problems with highly-linked variables. However, there are other problems where independent mating of variables through operators like SBX crossover causes a more efficient search [9].

Some recent studies in the EMO/EMaO domain have focused on an *online innovization* approach [10, 11, 12] which attempts to extract decision variable patterns or relationships that exist among the current non-dominated solutions and then apply this information as additional operators (apart from the crossover and mutation operators) in subsequent generations. These techniques explore the advantage of independent variable mating and exploit the learned inter-variable relationships at the same time. First, these studies have primarily been restricted to specific pre-fixed structures of relationships, such as power laws [13, 14]. Some limited studies of a more generic nature, using a genetic programming approach, have also been performed [15], which provide a promising direction for exploration and exploitation of these inter-variable relationships.

However, the scope of online innovization concept can also be extended to utilize the history of non-dominated solutions from earlier generations of an EMO/EMaO run to construct a *mapping model* which can improve an arbitrary variable-vector to map it closer to the current non-dominated set. A recent study [16, 9] has used artificial neural networks (ANNs) to learn a generic variable mapping between dominated and currently non-dominated target solutions and used the trained ANN model to repair the offspring solutions, thus providing offspring solutions to progress in the direction of the current non-dominated set. This learning process is discussed in Section II-B.

In this paper, we propose a random forest (RF) assisted

S. Mittal and D.K. Saxena are with the Department of Mechanical and Industrial Engineering, Indian Institute of Technology, Roorkee, Uttarakhand, 247667 India, e-mail: {smittal1,dhish.saxena}@me.iitr.ac.in.

K. Deb and E. Goodman are with the Beacon NSF Center and the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824 USA, e-mail: {kdeb,goodman}@egr.msu.edu.

machine learning (ML) method to capture the directional improvements in the optimization problem’s decision space in a progressive manner along with a few other improvements. Linking past dominated solutions with a specific target non-dominated solution, an RF is trained to develop a model, which is then used within the EMO/EMaO run to repair the generated offspring solutions into the *learned* directions. The proposed ML method is designed to focus on maintaining a diverse solution set at all times during the EMO/EMaO run, so that in addition to the convergence properties, a well-diverse set of solutions are also emphasized. The earlier naive approach [16] executed with ANN was reported to have difficulties in making useful repairs in certain problems. The proposed *innovized* repair operator with RF method (IR-RF) is reported to perform better on various problems covering different test suites, including ZDT, DTLZ, and WFG, with problems ranging from two to five objectives. Besides using RF over ANN, the basic IR operator is also modified with a number of other extensions which has produced an improved IR (IR2) operator. Extensive comparative evaluation of the proposed IR2-RF is made to reveal the benefits of the innovized repair operator concept.

In the remainder of this paper, we provide a brief overview of the existing learning-assisted EMO/EMaO algorithms including state-of-the-art methods for *online innovization*, and the past ANN-assisted repair operators in Sections II-A and II-B, respectively. The proposed IR-RF operator is outlined in Section III, followed by some comparative results. Section IV discusses modifications to IR operator by augmenting it with a number of specific features that allow the proposed IR2 operator to yield performance improvements. The differences between IR and IR2 operators are discussed in Section IV-F. Section V presents the results with new IR2 operator on a wide range of test problems, followed by a discussion of results in Section VI. The paper is concluded in Section VII.

II. EXISTING STATE-OF-THE-ART

In EMO/EMaO algorithms, the genetic operators—i.e., crossover and mutation—are not effective in reaching the optimal solution quickly for certain problems [17, 18]. Towards this goal, some approaches have been coupled with EMO/EMaO algorithms, such as local-search methods and surrogate-modeling techniques, to reduce the overall number of function evaluations. A recent study [9] discusses how an ANN-assisted repair operator that builds a foundation for the proposed repair operator is inherently different from local-search or surrogate methods. Further, there are Hybrid Genetic Algorithms (GAs) that learn from the evolution in past iterations and use that information to guide the evolution of new solutions, further discussed in the next paragraph. Apart from these, sections II-A and II-B discuss the idea of *innovized repair* and the ANN-assisted repair operator.

Hybrid Approaches with repair heuristics can be broadly classified into Lamarckian and Baldwinian approaches [19]. If the genotype or the decision variable vector is changed using a local repair heuristic, it is known as a Lamarckian algorithm. On the other hand, Baldwinian-learning-based heuristics

modify the fitness landscape rather than the genotype. [20] suggests that adaptive Baldwinian learning can do well in guiding the course of evolution in an EA, altering the shape of the search space and thus providing good evolutionary paths for the individuals. It may be noted that Baldwinian Learning is not a learning mechanism or method; rather, it designates the manner of learning. An example of this is the Multiobjective Immune Algorithm with Baldwinian Learning (MIAB) [21]. In MIAB, the learning operator at generation t captures the individual evolving directions (in decision-space) from $t - 1$ to t and uses it to decide the individual direction of evolution from t to $t + 1$. It may be noted that this learning-based evolution operator is used as an alternative to genetic operators—i.e., crossover and mutation—but our proposed approach repairs/evolves the offspring which have already been produced using genetic operators.

A. Innovization

Innovization is a knowledge-discovery task first proposed by Deb [22] and was later exemplified on several engineering design problems [23]. Even though *innovization* was proposed initially as a post-optimization process, a few studies [10, 12] later claimed that knowing these relationships through some intermediate-generation solutions during an optimization run and using this knowledge can enhance the convergence of the optimization algorithm. These relationships can be used to *repair* the solutions generated by EMO/EMaO operators, thus forcing these new solutions to possess some properties obtained from those good solutions, eventually accelerating the convergence. There are many issues and variables regarding learning at intermediate generations, such as the process/mechanism of learning, type/format of information to be extracted, and how this extracted information can be used to improve the existing solutions.

B. ANN-assisted Innovized Repair (IR-ANN) Operator

The study in [16] made the first attempt to capture the direction of evolution in a multi-dimensional search space using an ANN and applied it back as an offspring repair operator to enhance the speed of convergence. A more comprehensive study on the same IR operator was done in [9]. This IR operator has three processing steps, i.e., (i) Generation of training-dataset (in normal EMO/EMaO process), (ii) ANN-Training, and (iii) Repairing offspring solutions. Any intermediate generation t of the EMO/EMaO with the ANN-assisted repair operator is described in the steps below.

- Step 1: Update the Archive A_t with population members from t_{past} previous generations.
- Step 2: Check the population threshold, i.e., whether or not 50% population from the current population P_t is non-dominated. This is a deciding factor whether or not the learning and repair operation will happen at generation t .
- Step 3: Generate the training-dataset D_t from A_t . Towards this, every member in A_t is associated to the nearest reference vector (say Z_i), and then is paired with the

non-dominated solution (in current population P_t) nearest to that particular reference vector Z_i .

Step 4: Train an ANN using D_t .

Step 5: Generate offspring population Q_t using the crossover and mutation operators on P_t .

Step 6: Repair a randomly selected set of 50% of the total offspring solutions Q_t using the trained ANN model.

Step 7: Evaluate the offspring solutions.

Step 8: Select solutions from $P_t \cup Q_t$ to yield P_{t+1} .

The aforementioned process is guided by two user-defined parameters. The parameter t_{past} governs how many previous generations are considered in the archive A_t and the parameter t_{freq} governs the frequency of repairs, i.e., the repair takes place only after t_{freq} generations. The study [9] also discussed how the ANN-assisted repair operator differs from surrogate-modeling techniques or other local search operators for EMO/EMaO algorithms.

III. INNOVIZED REPAIR (IR) OPERATOR WITH RF

In this section, we analyze some results for how the performance changes when the associated learning method is changed from ANN to random forest (RF) method, referred to as IR-ANN and IR-RF, respectively. Since the learning is in the decision space, each training input and output is a vector of length n_{var} (number of variables). The RF has three critical parameters to be defined, i.e., the number of trees (n_{trees}), number of variables/features considered while splitting a node ($n_{features}$) and the splitting criterion. Considering that the size of the input archive is N_A , the following settings have been used for the RF:

$$n_{trees} = N_A, \quad n_{features} = n_{var}. \quad (1)$$

In addition to this, the splitting criterion used is *Mean Squared Error* (MSE), for a fair comparison with the ANN that uses MSE as the loss function. The rest of the implementation settings are taken as default¹. Some comparison results are shown in Table IV when NSGA-II is used to solve ZDT problems with the IR operator (proposed earlier in [9] and discussed in Section II-B) but using ANN and RF as learning methods. These are median hypervolume values at the end of 200 generations. The better results are marked in bold. The parameters used for NSGA-II and the IR operator are discussed later in Section V-B. This limited study indicates

TABLE I: Median Hypervolume on ZDT problems for NSGA-II with IR operator using ANN and RF as learning methods.

Problems	NSGA-II-IR-ANN	NSGA-II-IR-RF	p -value
ZDT1	0.679193	0.680198	8.98E-11
ZDT2	0.345448	0.346683	1.33E-11
ZDT3	0.535027	0.535561	1.62E-11
ZDT4	0.680704	0.681041	4.93E-04
ZDT6	0.333949	0.304714	1.40E-05

that there is a benefit of using RF over ANN in these kinds

¹The Random Forest Regressor used in this study has been taken from this Scikit-learn implementation (for python language). <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

of problems. It may be noted that for ZDT6, even though RF performed worse than ANN, both NSGA-II-IR-ANN and NSGA-II-IR-RF did better than NSGA-II (HV = 0.296750). In addition to changing the learning method from ANN to RF, this paper proposes some modifications in the core of IR operator design like the training dataset generation step and the repair step, which are further discussed in next section.

IV. MODIFIED INNOVIZED REPAIR (IR2) OPERATOR

Aligning with the existing IR operator, the new IR2 operator can be used with any suitable EMO/EMaO algorithm, such as NSGA-II [3], NSGA-III [4, 5], MOEA/D [6], etc. The main aim towards proposing the IR2 operator is to put emphasis on preserving the diversity in the EMO/EMaO population. The IR2 framework is broadly divided into three steps: (A) *Data Generation* Step, (B) *Training* Step, and (C) *Repair* Step. The overall framework is shown in Figure 1, where the three steps above are highlighted in gray. Step (A) is divided into two parts, i.e., updating target solutions and archive mapping, as explained in Sections IV-A1 and IV-A2, respectively. Steps (B) and (C), i.e., the RF training and repair steps, are detailed in Sections IV-B and IV-C, respectively, followed by their integration with multiple EMO/EMaO algorithms in Section IV-E. As mentioned earlier in Section II-B, this framework is also guided by two user-defined parameters, i.e., t_{past} and t_{freq} , along with a newly introduced parameter η to control the greediness of the repair operator. These steps are discussed in more detail in upcoming subsections, followed by the integration algorithm.

A. Data Generation Step

At any generation t of the EMO/EMaO algorithm solving an M -objective optimization problem, let the parent and offspring populations, each of size N , be referred to as P_t and Q_t , respectively. The *archive* A_t (of size N_A) consists of the parent population of the $(t - t_{past})^{th}$ generation and offspring populations from t_{past} previous generations. At the same time, a *target-archive* T_t (of size N) is also maintained, and is updated in each generation. The archive A_t and target-archive T_t together constitute the *input-output* or *input-target* dataset D_t for the RF training. The process of maintaining the target-archive and of mapping the archive population to these targets for generating the training dataset is discussed in detail in Sections IV-A1 and IV-A2, respectively.

1) *Updating the Target Solutions*: As mentioned earlier, this learning mechanism ensures that diversity is maintained along with convergence towards the optimal front; i.e., both goals of EMO/EMaO algorithms are pursued together. Towards this realization, a well-distributed target-archive T_t is created and maintained with the help of a uniformly-distributed set of reference points Z (size = N , dimensions = M). In this study, these reference points were generated using the Das-Dennis method [24], but they can alternatively be generated by other methods like a recently proposed generic sampling method on an M -dimensional unit simplex [25]. Since the sizes of T_t and Z are the same, ideally, there should be one target member from T_t associated with each reference

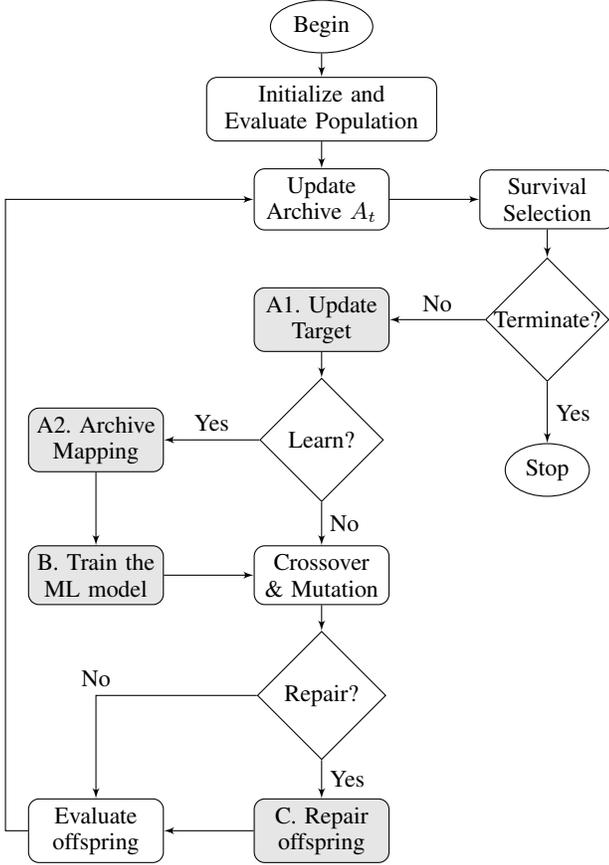


Fig. 1: EMO/EMaO algorithm with IR2 operator.

point. However, the case of a non-uniform distribution at any generation t has been discussed further.

In order to associate the current population members P_t with the reference points, a normalization procedure is performed in the objective space. Given that $F_k^{(i)}$ represents the k^{th} objective value for the i^{th} individual in P_t , $\bar{F}_k^{(i)}$ represents the k^{th} normalized objective value for the i^{th} individual in P_t , and Z_k^{ideal} and Z_k^{nadir} represent the ideal and nadir points in $P_t \cup T_t$, the normalization process is given by,

$$Z_k^{\text{ideal}} = \min_{i=1}^{N_A} F_k^{(i)}, \quad (2)$$

$$Z_k^{\text{nadir}} = \max_{i=1}^{N_A} F_k^{(i)}, \quad (3)$$

$$\bar{F}_k^{(i)} = \frac{F_k^{(i)} - Z_k^{\text{ideal}}}{Z_k^{\text{nadir}} - Z_k^{\text{ideal}}}, \quad \forall k = 1, \dots, M \quad (4)$$

In the normalized objective space, the association of an EMO/EMaO solution with any reference-point can be done in multiple ways, like using Achievement Scalarizing Function (ASF) [26], or Perpendicular Distance Matrix (PDM) [4], or Penalty-based Boundary Intersection (PBI) [6]. An earlier study [9] discussed the benefits of choosing ASF over PDM for an EMO/EMaO algorithm that does not use any support from reference points, like NSGA-II. For using the proposed operator with other EMO/EMaO algorithms that use a set of reference points, like MOEA/D, NSGA-III, etc., it would be ideal to use the same set of reference points and the same asso-

ciation method as used in the EMO/EMaO algorithm. Hence, we use ASF with a uniform weight vector $w = (1, 1, \dots, 1)^T$ of size $M \times 1$ for NSGA-II, PDM for NSGA-III and PBI for MOEA/D.

Target association can be done by two methods, i.e., (i) by identifying the nearest candidate target solution to each reference-vector, and (ii) by identifying the nearest reference-vector to each candidate target solution. While the earlier IR operator uses the former method which ensures that each reference-vector is assigned a target solution (with possibility of duplicates), the proposed IR2 operator uses the latter method which ensures no duplicates in target solutions but may not be able to cover all the reference-vectors. For example, in Figure 2, there is no non-dominated solution associated with the vector R5 (as D dominates E). While the earlier IR operator would have chosen D, based on the smallest ASF value for R5 as the target for R5, the proposed IR2 operator would choose E (a dominated solution, but being close to R5) rather than choosing a solution that belongs to some other reference-vector. If there were no solutions near to the vector R5 at all, there would have been no target assignment for that vector.

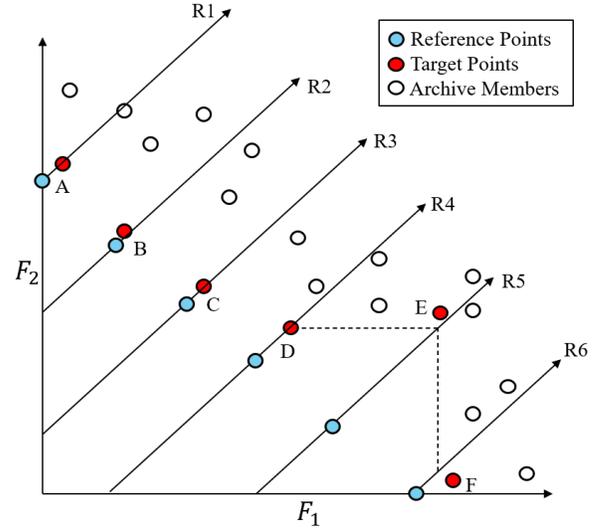


Fig. 2: A schematic for identifying target-points using ASF in a two-objective space.

The target archive T_t can have at most N solutions (one for each reference vector). However, as in the above stated example, it may happen that the EMO algorithm does not have any solutions associated with some reference-vectors in the early generations. Due to this, the target archive may have less than N solutions as well for a few generations at the beginning. However, for implementation of IR2 operator, T_t is initialized as an empty array of size N , where $T_t^{(i)}$ represents the target solution assigned to the reference point $Z^{(i)}$. The process of updating this target archive from T_{t-1} to T_t using the current parent population P_t is given in Algorithm 1. Each solution in P_t is evaluated using ASF (or some other metric) to associate it with some reference vector. Let α and β be two indices such that the solution $P_t^{(\alpha)}$ gets associated with the reference-vector $Z^{(\beta)}$. Now, if $T_t^{(\beta)}$ (the target assigned to

$Z^{(\beta)}$ is empty, that is, no target solution has been associated previously with $Z^{(\beta)}$, then the place is filled with $P_t^{(\alpha)}$. Else, if there is an existing solution at $T_t^{(\beta)}$, then their individual ASF values (with respect to $Z^{(\beta)}$) are compared to determine if the replacement is needed.

Algorithm 1 $T_t = \text{Update_Target}(P_t, T_{t-1}, Z)$

Require: Reference points Z , Target archive T_t , Parent Population $P_t = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})$

- 1: $F_T \leftarrow$ Objective function values in T_{t-1}
 - 2: $F_P \leftarrow$ Objective function values in P_t
 - 3: Compute Z^{ideal} & Z^{nadir} from F_P %Eqs. 2-3
 - 4: $\bar{F}_P \leftarrow$ Normalize F_P using Z^{ideal} and Z^{nadir} %Eq. 4
 - 5: $\bar{F}_T \leftarrow$ Normalize F_T using Z^{ideal} and Z^{nadir} %Eq. 4
 - 6: **for** $i = 1$ to N **do** % for each pt. in F_P
 - 7: **for** $j = 1$ to N **do** % for each pt. of Z
 - 8: $ASF_j \leftarrow \max_{k=1}^M (\bar{F}_{P,k}^{(i)} - Z_k^{(j)})$
 - 9: $[Index, ASF_P] \leftarrow \min_{j=1}^N ASF_j$
 - 10: **if** $T_t^{(Index)}$ is empty **then**
 - 11: $T_t^{(Index)} \leftarrow P_t^{(i)}$
 - 12: **else**
 - 13: $ASF_T \leftarrow \max_{k=1}^M (\bar{F}_{T,k}^{(Index)} - Z_k^{(Index)})$
 - 14: **if** $ASF_P < ASF_T$ **then**
 - 15: $T_t^{(Index)} \leftarrow P_t^{(i)}$
 - 16: **return** Updated target-archive T_t
-

2) *Archive Mapping*: The goal is to map the solutions in archive A_t to those in the target archive T_t to generate a training dataset that is then fed to train an ML model or a multi-target regressor.

This goal is realized by associating a solution in A_t with one of the reference points, say $Z^{(\beta)}$, for which a corresponding target solution $T_t^{(\beta)}$ already exists. However, it may be noted that to normalize the archive solutions, the Z^{ideal} and Z^{nadir} are calculated separately for A_t , and may be different from the ideal and nadir points of the target archive T_t . This helps in ensuring that if diversity has been lost in the past few generations, it can again be regained (partially) by this learning-repair process. Following this, each member in archive A_t is associated with one reference point—say $A_t^{(\alpha)}$ is associated with $Z^{(\beta)}$. Then, $A_t^{(\alpha)}$ goes to the *Input* and $T_t^{(\beta)}$ goes to the *Output*, which then forms one *Input-Output* sample for the training dataset D_t . This process is shown in Algorithm 2, which returns the dataset used for ML training.

B. Training Step

After the training dataset D_t is generated, the ML model is trained to capture the *Input-Output* relationships of the decision variable vectors. The training process is presented in Algorithm 3, which mainly consists of two steps, i.e., the *dynamic normalization* (discussed in our earlier work [9]) of training-dataset D_t (a pre-training step) and the training of the ML model itself.

Algorithm 2 $D_t = \text{Archive_Mapping}(A_t, T_t, Z)$

Require: Archive A_t , Target-archive T_t , Ref.-points Z .

- 1: $N_A \leftarrow$ Size of Archive A_t
 - 2: $F_A \leftarrow$ Objective function values in A_t
 - 3: Compute Z^{ideal} & Z^{nadir} from F_A %Eqs. 2-3
 - 4: $\bar{F}_A \leftarrow$ Normalize F_A using Z^{ideal} and Z^{nadir} %Eq. 4
 - 5: $Input \leftarrow \emptyset$
 - 6: $Output \leftarrow \emptyset$
 - 7: **for** $i = 1$ to N_A **do** % for each pt. in A_t
 - 8: **for** $j = 1$ to N **do** % for each pt. in Z
 - 9: $ASF_j \leftarrow \max_{k=1}^M (\bar{F}_{A,k}^{(i)} - Z_k^{(j)})$
 - 10: $Index \leftarrow \arg\min_{j=1}^N ASF_j$
 - 11: $Input \leftarrow Input \cup A_t^{(i)}$
 - 12: $Output \leftarrow Output \cup T_t^{(Index)}$
 - 13: $\mathbf{x}_{Input} \leftarrow$ Variable vectors in $Input$
 - 14: $\mathbf{x}_{Output} \leftarrow$ Variable vectors in $Output$
 - 15: $D_t \leftarrow [\mathbf{x}_{Input}, \mathbf{x}_{Output}]$
 - 16: **return** Training Dataset D_t
-

Algorithm 3 ($\text{Predict}(), [\mathbf{x}^{\min}, \mathbf{x}^{\max}] = \text{Training}(D_t, [\mathbf{x}^l, \mathbf{x}^u])$)

Require: Training dataset D_t , Lower & upper variable-bounds specified in the problem \mathbf{x}^l and \mathbf{x}^u .

- 1: $\mathbf{x}^{l,t} \leftarrow$ Min. variable vector in *Input-Output* data D_t
 - 2: $\mathbf{x}^{u,t} \leftarrow$ Max. variable vector in *Input-Output* data D_t
 - 3: **for** $k = 1$ to n_{var} **do** % No. of variables
 - 4: $x_k^{\min} = 0.5(x_k^{l,t} + x_k^l)$
 - 5: $x_k^{\max} = 0.5(x_k^{u,t} + x_k^u)$
 - 6: Normalize D_t using \mathbf{x}^{\min} and \mathbf{x}^{\max} as bounds
 - 7: Train the ML model using D_t to create $\text{Predict}()$
 - 8: **return** $\text{Predict}()$, Bounds $[\mathbf{x}^{\min}, \mathbf{x}^{\max}]$
-

C. Repair Step

The trained ML model is then used to repair the offspring solutions Q_t obtained after genetic recombination and mutation operations at generation t . This repair step consists of 4 different processes, i.e.,

- 1) Selecting 50% of offspring randomly and repairing them using the $\text{Predict}()$ function obtained after training the ML training.
- 2) *Near-bound Restoration* – Any variable near one of its bounds (before Repair) is restored to its original value.
- 3) *Boundary Repair* – Any variable that lies outside its respective permissible bounds $[\mathbf{x}_k^l, \mathbf{x}_k^u]$ is fixed to an inner value based in Inverse Parabolic Spread Distribution [27].
- 4) *Enhancement* of the repaired solutions.

Process-4 is newly introduced, as explained further in the next subsection.

Enhancement of Repaired solutions: A new parameter governs enhancement of the repaired solutions, hereafter referred to as η . At any intermediate generation, the ML model would learn the *Input-Output* relationships, which could have been learned from the best solutions belonging to the already explored search space. However, the global optima may lie in

regions of the search space yet to be explored, implying that the relationships learned until now are prematurely restrictive. Given that, for an offspring X and a repaired offspring Y , a greedy algorithm would exploit this possibility by further extrapolating the solution in the decision space. This exploitation is done using parameter η by defining the enhanced solution Y' as,

$$Y' = X + \eta(Y - X), \quad (5)$$

where $\eta = 1$ is equivalent to no enhancement, and any value of $\eta > 1$ will potentially enhance the repaired solutions. However, too much greediness can be disadvantageous as well, a study of which is done in V-B. In the future, this parameter can be made adaptive as the EMO/EMaO algorithm progresses. However, in this paper it is kept fixed to the value defined while initializing the EMO/EMaO algorithm. The complete repair process is presented in Algorithm 4.

Algorithm 4 $Q_t = \text{Repair}(Q_t, \eta, [\mathbf{x}^{\min}, \mathbf{x}^{\max}], [\mathbf{x}^l, \mathbf{x}^u], \text{Predict})$

Require: Offspring Q_t , Enhancement factor η , Bounds from Algorithm 3 $[\mathbf{x}^{\min}, \mathbf{x}^{\max}]$, Variable bounds in problem $[\mathbf{x}^l, \mathbf{x}^u]$, $\text{Predict}()$ function.

```

1:  $I \leftarrow$  Randomly selected 50%  $Q_t$ 
2:  $range_k \leftarrow (x_k^u - x_k^l)$ ,  $k = 1, \dots, n_{var}$ 
3: for  $i = 1$  to  $N/2$  do
4:    $temp \leftarrow$  Normalize  $I^{(i)}$  using  $\mathbf{x}^{\min}$  and  $\mathbf{x}^{\max}$ 
5:    $R^{(i)} \leftarrow \text{Predict}(temp)$ 
6:    $R^{(i)} \leftarrow$  Denormalize  $R^{(i)}$  using  $\mathbf{x}^{\min}$  and  $\mathbf{x}^{\max}$ 
7:    $R^{(i)} \leftarrow I^{(i)} + \eta(R^{(i)} - I^{(i)})$ 
8:   for  $k = 1$  to  $n_{var}$  do
9:      $vicinity \leftarrow \min(|I_k^{(i)} - x_k^{\min}|, |x_k^{\max} - I_k^{(i)}|)$ 
10:    if  $vicinity \leq 0.01 * range_k$  then
11:       $R_k^{(i)} \leftarrow I_k^{(i)}$ 
12:    Replace  $I^{(i)}$  by  $R^{(i)}$  in  $Q_t$ 
13:  Boundary Repair on  $I^{(i)}$ 
14: return Repaired offspring  $Q_t$ 

```

D. Learning Termination

The kind of learning-based repair operator described above can help gain faster convergence by improving the fitness of the offspring generated by crossover and mutation operators [16, 9]. However, the effect of repairs is bound to decrease in later generations when the population is nearly converged, as compared to earlier generations. Hence, a termination criterion is incorporated, which monitors the improvement in solutions and adaptively turns off the learning when the improvement becomes relatively smaller. Since convergence is the only criterion for termination here, the metric used is GD (Generational Distance), which is evaluated at generation t for P_d (where $d = t - t_{past}$) with the current generation's population as the reference set. This running metric will be denoted by G , given as,

$$G_t(P_d) = \frac{1}{|P_d|} \sqrt{\sum_{i=1}^{|P_d|} d_i^2}, \quad (6)$$

where d_i represents the euclidean distance (in F -space) from a member in P_d to its nearest population member in P_t . This parameter G is monitored at every generation after the repair operator is activated. If the value of G_t drops below $g_{th}\%$ of the maximum value of G until then, the learning and repair steps are switched off, and the EMO/EMaO algorithm resumes its original operation. However, if G_t becomes higher than this threshold at some later generation, then the learning and repair operations resume. This is a probable scenario in problems where the base EMO/EMaO algorithm usually gets stuck in a local optimum. The effect of this threshold parameter for termination is discussed in Section VI-E.

E. Integration with EMO/EMaO Algorithms

1) *NSGA-II and NSGA-III*: The integration of the proposed IR2 operator with NSGA-II and NSGA-III is presented in Algorithm 5. The pseudo-code represents an intermediate generation of the optimization algorithm along with IR2-related steps. First, the target archive T_t is updated using the parent population P_t . Lines 2–7 check whether to use the IR2 operator as per parameter t_{freq} and the termination criterion given in Section IV-D. If repair is to be done, the *flag* is marked *True*, else *False*. In the learning step, mapping is done using the `Archive_Mapping` function (Algorithm 2) and then training is done using the `Training` function (Algorithm 3), as given in lines 8–10. Following this, mating is done using genetic operators (recombination and mutation, line 11) along with the repair process using the `Repair` function (Algorithm 4). The repair process (line 13) is done before the offspring fitness evaluation step (line 14) so that the offspring are evaluated only once, in line with the EMO algorithm. Line 15 shows the update of archive A_t as explained initially in Section IV-A, followed by survival selection in line 16.

2) *MOEA/D*: The integration of the proposed IR2 operator with MOEA/D is presented in Algorithm 6. The pseudo-code represents an intermediate generation of the optimization algorithm along with IR2 related steps. First, the target archive T_t is updated using the parent population P_t . Lines 2–7 check whether to use the IR2 operator as per the parameter t_{freq} and the termination criterion given in Section IV-D. If repair is to be done, the *flag* is marked *True*, else *False*. In the learning step, mapping is done using the `Archive_Mapping` function (Algorithm 2) and then training is done using `Training` function (Algorithm 3), as given in lines 8–10. Following this, 50% of solutions are randomly selected (line 12), and their indices are stored in I . Lines 14–15 represent the selection and mating process—i.e., whether the parents are selected from the neighborhood or the entire population—followed by creation of a single offspring. If the index of this offspring was selected in I , it is repaired using the `Repair` function (Algorithm 4). The offspring are then evaluated, and neighborhood solutions are updated according to the MOEA/D algorithm. In the end, line 20 shows the update of Archive A_t .

Integration with other EMO/EMaO algorithms can also be suitably achieved.

Algorithm 5 Generation t of NSGA-II and NSGA-III with IR2 operator.

Require: Reference Set Z , Parent population P_t , Archive A_t , Previous Target-archive T_{t-1} , Collection t_{past} , Frequency t_{freq} , Enhancement Factor η , Variable bounds defined in problem $[\mathbf{x}^l, \mathbf{x}^u]$.

- 1: $T_t \leftarrow \text{Update_Target}(P_t, T_{t-1}, Z)$
- 2: $count \leftarrow \text{rem}(t, t_{freq})$
- 3: $G \leftarrow G \cup G_t$ % Equation 6
- 4: **if** $G_t > g_{th}\%$ of $\max(G)$ & $count = 0$ **then**
- 5: $flag \leftarrow \text{True}$
- 6: **else**
- 7: $flag \leftarrow \text{False}$
- 8: **if** $flag = \text{True}$ **then**
- 9: $D_t \leftarrow \text{Archive_Mapping}(A_t, T_t, Z)$
- 10: $(\text{Predict}, [\mathbf{x}^{\min}, \mathbf{x}^{\max}]) \leftarrow \text{Training}(D_t, [\mathbf{x}^l, \mathbf{x}^u])$
- 11: $Q_t \leftarrow \text{Recombination} + \text{Mutation}(P_t)$
- 12: **if** $flag = \text{True}$ **then**
- 13: $Q_t \leftarrow \text{Repair}(Q_t, \eta, [\mathbf{x}^{\min}, \mathbf{x}^{\max}], [\mathbf{x}^l, \mathbf{x}^u], \text{Predict})$
- 14: Evaluate Q_t
- 15: $A_{t+1} \leftarrow (A_t \cup Q_t \cup P_{t+1-t_{past}}) \setminus [P_{t-t_{past}} \cup Q_{t-t_{past}}]$
- 16: $P_{t+1} \leftarrow \text{Survival selection of NSGA-II/III with } P_t \cup Q_t$
- 17: **return** Next Parent Population P_{t+1} , Next Archive A_{t+1} , Target-archive T_t

F. Differences with earlier IR operator

Apart from changing the associated learning technique from ANN to RF, there are certain other differences between the core of the IR and IR2 operators with a primary focus on preserving diversity in population and some other issues, as enumerated below.

- 1) *Archive Members* – The IR operator has an archive A_t that contains solutions from t_{past} previous generations. Since not all parents P_t are replaced by offsprings Q_t in the survival selection, this archive will have multiple copies of same solution, more so in later generations where only a small fraction of offsprings might get selected. This results in duplicates in the training dataset D_t , undermining its generality. To address this, the archive A_t in the IR2 operator stores offspring solutions from t_{past} previous generations and one parent population $P_{t-t_{past}}$. This new archive covers all the solutions in the earlier definition of archive and has possibly no duplicates, advocating for a better learning process.
- 2) *Target Archive* – Unlike the IR operator that chooses the target solutions from the non-dominated solutions in the current population P_t , the IR2 operator maintains a separate target-archive of the N best solutions found across each of the N reference vectors. The target-archive T_t is updated at every generation, from the solutions in P_t if found better than any of the existing solutions in T_t . This ensures that each part of the obtained front contributes to the training dataset, even though some part may be dominated from some other target solutions. With this modification, the requirement of 50% of P_t to

Algorithm 6 Generation t of MOEA/D with IR2 operator.

Require: Reference Set Z , Parent population P_t , Neighborhood solutions B , Neighborhood size N_S , Population Size N , Archive A_t , Previous Target-archive T_{t-1} , Collection t_{past} , Frequency t_{freq} , Enhancement Factor η , Variable bounds defined in problem $[\mathbf{x}^l, \mathbf{x}^u]$.

- 1: $T_t \leftarrow \text{Update_Target}(P_t, T_{t-1}, Z)$
- 2: $count \leftarrow \text{rem}(t, t_{freq})$
- 3: $G \leftarrow G \cup G_t$ % Equation 6
- 4: **if** $G_t > g_{th}\%$ of $\max(G)$ & $count = 0$ **then**
- 5: $flag \leftarrow \text{True}$
- 6: **else**
- 7: $flag \leftarrow \text{False}$
- 8: **if** $flag = \text{True}$ **then**
- 9: $D_t \leftarrow \text{Archive_Mapping}(A_t, T_t, Z)$
- 10: $(\text{Predict}, [\mathbf{x}^{\min}, \mathbf{x}^{\max}]) \leftarrow \text{Training}(D_t, [\mathbf{x}^l, \mathbf{x}^u])$
- 11: $P_{t+1} \leftarrow P_t$
- 12: $I \leftarrow 50\%$ Randomly selected indices from N
- 13: **for** $i = 1$ to N **do**
- 14: $j, k \leftarrow \text{Two selected indices from } B(i) \text{ or } P_t$
- 15: $Q_t^{(i)} \leftarrow \text{Recombination} + \text{Mutation}(P_t^{(j)}, P_t^{(k)})$
- 16: **if** $flag = \text{True}$ and $i \in I$ **then**
- 17: $Q_t^{(i)} \leftarrow \text{Repair}(Q_t^{(i)}, \eta, [\mathbf{x}^{\min}, \mathbf{x}^{\max}], \text{Predict})$
- 18: Evaluate $Q_t^{(i)}$
- 19: Update the neighborhood solutions $B(i)$ in P_{t+1}
- 20: $A_{t+1} \leftarrow (A_t \cup Q_t \cup P_{t+1-t_{past}}) \setminus [P_{t-t_{past}} \cup Q_{t-t_{past}}]$
- 21: **return** Next Parent Population P_{t+1} , Next Archive A_{t+1} , Target-archive T_t

be non-dominated for learning is removed from the IR2 operator, which was there earlier in the IR operator.

- 3) *Target Association* – In the IR operator, each reference vector is supplied with its nearest non-dominated solution as the target, even though that solution may belong in the region of some other vector. In case the EMO algorithm has no non-dominated solution for a particular vector, the IR operator will map the archived solutions on that vector to the assigned target solution (which would belong to some other reference vector). This may create some noise in the dataset and may contribute in further widening the gap in the front. Towards this effect, the IR2 operator checks the association of a current solution with a reference vector, and if found better, then assigns it to that particular vector only. This would mean that some reference-vectors may not have a target solution in the initial generations, but then they simply would not be considered for training unless they have some representative solutions.
- 4) *Boundary repair* – In the IR operator, if any variable goes out of its respective bounds during the repair step, that variable is fixed at the respective boundary value. However, in case of IR2, the boundary repair value is determined using an inverse parabolic spread distribution [27]. This would be beneficial since fixing a variable to its boundary in multiple solutions may cause

inefficiencies for the genetic mating operators.

- 5) *Enhancement of repaired offsprings* – This is a new proposition in the IR2 operator, which did not exist in the IR operator earlier. This enhancement contributes directly towards greediness of the repair operator, an effect of which is visualized through a small parametric study in Section V-B.

It is evident that points 2–4 mentioned above focus on maintaining the diversity in either F - or X -space, which advocates for the main aim of designing this modified IR2 operator. These inclusions make IR2 a more robust operator to be used for a wide range of optimization problems. The efficacy of this operator is discussed in the following sections.

V. RESULTS WITH IR2 OPERATOR

In this section, the performance of the IR2 operator is analyzed for three EMO/EMaO algorithms, i.e., NSGA-II [3], NSGA-III [4, 5] and MOEA/D [6]. These algorithms, integrated with the IR2 operator and RF as the ML technique, are hereafter called NSGA-II-IR2-RF, NSGA-III-IR2-RF and MOEA/D-IR2-RF. The main aim of the proposed IR2 operator is to assist the underlying EMO/EMaO algorithm to converge faster, while ensuring that the diversity in F -space is maintained. In other words, the IR2 operator is expected to achieve a result in, say, t iterations, for which the EMO/EMaO algorithm without repair would have taken $t + \Delta t$ iterations. Since the IR2 operator, and even the IR operator, do not require any extra function evaluations, the Δt iterations and corresponding function evaluations can be seen to be a direct saving. This benefit is bound to decrease as the optimization progresses and the population reaches the vicinity of the PO front. However, in real-world problems, the number of allowed function evaluations are usually limited because they are computationally expensive. Hence, we try to assess the efficacy of the proposed IR2 operator on several 2–5-objective benchmark test problems, analyzing the results at intermediate generations.

For this analysis, we primarily use the hypervolume (HV) metric, since it is indicative of both convergence and diversity of the solution set. At any intermediate generation t for a particular test instance, the median hypervolume (from multiple independent seed runs) is recorded for both algorithms, an EMO/EMaO algorithm with and without the IR operator. Let us call these HV values as H_R^t and H^t , respectively. Since these values are different, let us say that the EMO/EMaO algorithm without repair takes $t + \Delta t$ iterations to reach to an equivalent HV, i.e., $H_R^t = H^{t+\Delta t}$. Δt may be negative if the IR operator makes the EMO/EMaO algorithm perform worse than the original algorithm. The percentage savings (S) in the function evaluations can be calculated as,

$$S = \frac{\Delta t}{t} \times 100\% \quad (7)$$

The S -metric is a direct indicator of the benefits of using the proposed IR2 or earlier IR operator. The comparison on different problem instances is done only with the same underlying algorithm, since we propose this as a generic learning-based repair operator which can be used with any

other learning algorithm (than RF) or any other EMO/EMaO algorithm. Apart from this, the final generation results are often similar with/without the IR/IR2 operator, since both algorithms approach the Pareto front. However, the end results are also better with the IR/IR2 operator in some instances where the genetic operators are simply unable to converge.

For performance comparison, hypervolume (HV) has been used to quantify the measure of convergence and diversity. For some problems, Generational Distance (GD) and Inverted Generational Distance (IGD) has also been computed. The HV-computation has been done with reference point as $[P]_{(1 \times M)}$ for M -objective problems, where $P = \frac{N}{N-1}$ (N is the population size). The GD and IGD computation has been done with a Pareto front of 500 equally space data points for 2-objective problems. Also, p -values have been calculated using the Wilcoxon Rank-Sum test to supplement the visually apparent differences with statistical reasoning. For WFG problems, where the PO front does not lie between 0 and 1, the i^{th} objective is normalized with bounds $[0, 2 * i]$ before hypervolume is computed.

A. Test Suite

Several two to five-objective optimization problems have been used in this paper, in order to establish the improved performance of an EMO/EMaO algorithm using the IR2 operator as compared to the base EMO/EMaO algorithm. Before starting with the extensive results, a small parametric study for the enhancement parameter η is presented in Section V-B, for which two problems L1 and L2 have been taken from [9]. The dependency on the learning termination threshold is discussed in Section VI-E.

For demonstrating the efficacy of the proposed IR operator, several benchmark test suites have been used, namely ZDT [28], WFG [29] and DTLZ [30]. In the ZDT problems, their respective $g(X)$ -functions have been modified to have PO solutions at $x_k = 0.5 \forall k = 2, \dots, 30$. Further, WFG problems are solved with $M = 3, 4$ and total 24 variables. For these problems, the position-related variables k are fixed as $k = 2 \times (M - 1)$. The difficulty of these problems is scalable and parameter dependent, and these values are generally taken as the defaults from [29]. In some instances, these parameters are modified to increase the difficulty (discussed in Section VI). For the DTLZ problems, the number of objectives is varied as $M = 3, 4, 5$ while the number of variables is fixed at 15.

B. Parameter Settings

The base EMO/EMaO algorithms deployed with the RF-assisted IR2 operator are NSGA-II (for 2-objective problems), and NSGA-III and MOEA/D (for problems with more than two objectives). NSGA-III is chosen over NSGA-II for three-objective problems since studies suggest that NSGA-III outperforms NSGA-II on these problems [31]. For all problems, SBX crossover ($p_c = 0.9$ and $\eta_c = 10$), and polynomial mutation ($p_m = 0.1$ and $\eta_m = 1/n_{var}$) is used. The population size N is kept different, varying with the number of objectives M as given in Table II, along with number of gaps

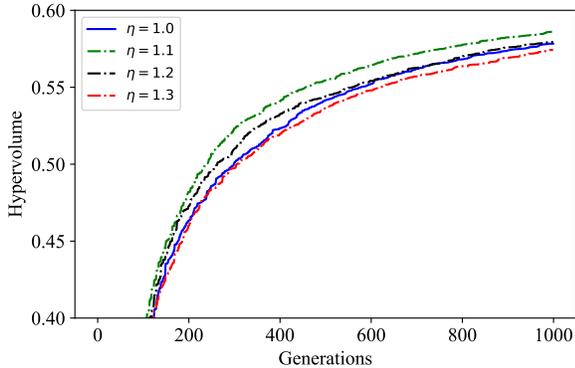
TABLE II: Population Size N and gaps for Das-Dennis method p for problems with M objectives.

M	2	3	4	5
N	100	105	286	495
p	99	13	10	8

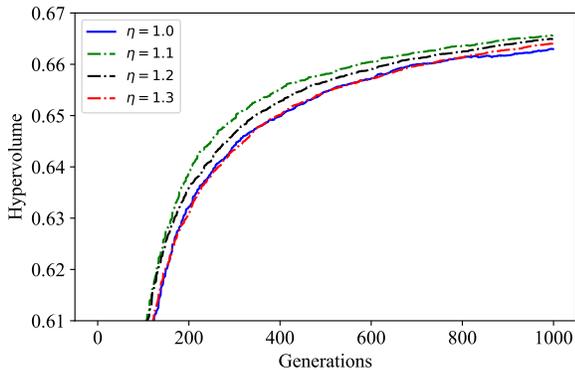
p for the Das-Dennis method to generate exactly N reference-points.

From [9], the archive collection parameter t_{past} and the frequency of repairs t_{freq} are both fixed at 5. The parameters used for RF training have been discussed earlier in Section III.

We investigate the behaviour of the IR2-RF operator with respect to the newly introduced enhancement operator η . For the parametric study, we explore the values of η as 1.0 (i.e., no enhancement), 1.1, 1.2 and 1.3. The median hypervolumes from 16 independent runs over 1,000 generations are shown in Figure 3 for two problems, L1 and L2 [9]. It can be observed that there is a consistent advantage at almost all generations with $\eta = 1.1$. Table III also shows the median HV values at the final generation and their respective p -values. However, even though $\eta = 1.1$ produces the best results, it is noteworthy that solutions with $\eta > 1$ are better than no enhancement ($\eta = 1$) at all. To conclude, this preliminary parametric study motivates us to use $\eta = 1.1$ for all subsequent experiments in this paper.



(a) Problem L1.



(b) Problem L2.

Fig. 3: Median HV for different η on problems L1 and L2.

C. ZDT problems

First, we discuss and analyze the performance of the proposed IR2 operator (NSGA-II-IR2-RF) as compared to the base algorithm NSGA-II and NSGA-II-IR-RF (discussed earlier in Section III). For a fair comparison, RF is chosen as the learning method for both IR and IR2 operators. Their median hypervolume values at generation 100 are shown in Table IV. It is evident from the table that NSGA-II-IR2-RF performs either better, or, at par in all test instances. The better performance of NSGA-II-IR2-RF over NSGA-II-IR-RF implies that the modified IR operator is more effective, considering that the learning method (RF) is the same in both algorithms.

Further, the same ZDT problems are used to compare MOEA/D and MOEA/D-IR2-RF on two-objective problems. At generation 50, the HV, IGD and GD values are shown in Table V along with their respective p -values. MOEA/D-IR2-RF either outperforms, or, is at par with MOEA/D on all problems, except ZDT4. In ZDT4, the GD for MOEA/D is better but the results are not statistically different.

D. Other Multi- and Many-Objective Problems

In this subsection, we analyze the performance of NSGA-III-IR2-RF and MOEA/D-IR2-RF on WFG and DTLZ problems ranging from 3 to 5 objectives. Earlier studies [16, 9] clearly indicate that the hypervolume improvement is significant in early generations but the difference reduces as the solutions approach the Pareto front. It is relevant to discuss the convergence in early generations as well, since the number of available function evaluations is limited in many real-world problems. Table VI shows the analysis at generation $t = 40$ with median hypervolume values H^t and H_R^t for NSGA-III and NSGA-III-IR2-RF, respectively. Another column, named *Recovery*, depicts the generation ($t + \Delta t$) in which NSGA-III achieves a hypervolume equivalent to that of NSGA-III-IR2-RF at generation t . Then, the percentage savings in function evaluations (S) by the IR2 approach is calculated using Equation 7.

The results in Table VI indicate that the effectiveness of the proposed IR2 operator when used with the NSGA-III algorithm. The proposed algorithm performs worse only in problem WFG5 ($M = 3$). This problem is extremely difficult and has deception, which makes it fundamentally difficult for any kind of learning-based repair to yield better performance. However, it is also evident that the degradation in performance on that instance is much lower than the average improvements in most other test instances. A similar study is done for MOEA/D for three- and four-objective problems to demonstrate the effectiveness of the proposed IR2 operator when integrated with MOEA/D. The results are tabulated in Table VII, where the trend is in line with the results on NSGA-III mentioned in Table VI. The only difficulty MOEA/D-IR2-RF faces is on solving WFG5. However, the difference is not statistically significant, as indicated by the p -value.

Table VIII shows the median hypervolume results obtained from NSGA-III and NSGA-III-IR2-RF on three- to five-objective problems. The results are recorded at an intermediate

TABLE III: Effect of repair enhancement parameter η on two difficult problems L1 and L2.

Problem	$\eta=1.0$		$\eta=1.1$		$\eta=1.2$		$\eta=1.3$	
	median-HV	p-value	median-HV	p-value	median-HV	p-value	median-HV	p-value
L1	0.578317	5.11E-06	0.586048	-	0.57954	5.28E-03	0.574252	6.47E-05
L2	0.662975	1.40E-04	0.665669	-	0.664926	2.56E-04	0.664079	1.41E-06

TABLE IV: Hypervolume values for two-objective problems, comparing NSGA-II without repair, and with RF-assisted IR and IR2 operators. Best performing framework is marked in bold.

Problem	Generation	NSGA-II		NSGA-II-IR-RF		NSGA-II-IR2-RF	
		median-HV	p-value	median-HV	p-value	median-HV	p-value
ZDT1	100	0.675528	1.33E-11	0.678876	1.63E-02	0.679119	-
ZDT2	100	0.340607	1.33E-11	0.344396	1.97E-07	0.345401	-
ZDT3	100	0.532092	1.33E-11	<i>0.534475</i>	1.26E-01	0.534578	-
ZDT4	100	<i>0.673986</i>	1.06E-01	<i>0.675198</i>	3.49E-01	0.676978	-
ZDT6	100	0.159995	6.34E-09	0.169844	5.19E-07	0.230231	-

TABLE V: Performance metrics for two-objective problems, comparing MOEA/D without repair, and with repair (MOEA/D-IR2-RF). Best performing framework is marked in bold.

Problem	Generation	median-HV			median-IGD			median-GD		
		MOEA/D	MOEA/D-IR2-RF	p-value	MOEA/D	MOEA/D-IR2-RF	p-value	MOEA/D	MOEA/D-IR2-RF	p-value
ZDT1	50	0.534166	0.592069	2.04E-06	0.105212	0.085742	1.04E-02	0.117213	0.056263	1.40E-06
ZDT2	50	0.162505	0.208247	2.91E-03	0.161240	0.109551	3.17E-02	0.163532	0.086702	2.43E-05
ZDT3	50	0.900764	0.951958	3.96E-04	0.085657	0.055298	7.45E-03	0.065226	0.045721	1.64E-04
ZDT4	50	<i>0.183939</i>	0.254727	9.10E-01	<i>0.521270</i>	0.405933	7.34E-01	0.636865	<i>0.786373</i>	9.70E-01
ZDT6	50	<i>0.000000</i>	0.001219	2.00E-01	<i>0.668449</i>	0.581419	1.32E-01	0.725946	0.659610	3.82E-02

TABLE VI: Median Hypervolume metric and associated percentage savings in function evaluations (S -metric) on 3- and 4-objective WFG problems, solved using NSGA-III and NSGA-III-IR2-RF. Best performing framework is marked in bold.

Problem-(M)	Generation (t)	NSGA-III (H^t)	NSGA-III-IR2-RF (H_R^t)	p-value	Recovery w/o repair ($t + \Delta t$)	Saving ($S\%$)
WFG1-3	40	<i>0.005261</i>	0.005278	4.51E-01	40	0.0
WFG2-3	40	<i>0.781701</i>	0.793153	2.91E-01	47	17.5
WFG3-3	40	0.503857	0.519115	1.73E-05	55	37.5
WFG4-3	40	0.302729	0.336154	1.41E-06	75	87.5
WFG5-3	40	0.252149	0.248024	4.18E-02	39	-2.5
WFG6-3	40	0.225263	0.273883	5.11E-06	69	72.5
WFG7-3	40	0.275361	0.309183	1.41E-06	57	42.5
WFG8-3	40	0.224801	0.240901	1.90E-04	55	37.5
WFG9-3	40	0.284691	0.327777	2.87E-05	72	80.0
WFG1-4	40	0.057422	<i>0.052123</i>	6.51E-01	38	-5.0
WFG2-4	40	0.863800	0.887670	7.57E-05	52	30.0
WFG3-4	40	0.507060	0.529724	1.41E-06	65	62.5
WFG4-4	40	0.408224	0.458472	1.41E-06	79	97.5
WFG5-4	40	0.347052	<i>0.344402</i>	9.73E-02	40	0.0
WFG6-4	40	0.315560	0.407118	1.41E-06	93	132.5
WFG7-4	40	0.379668	0.446049	1.41E-06	68	70.0
WFG8-4	40	0.303973	0.331764	1.41E-06	60	50.0
WFG9-4	40	0.410231	0.423810	1.04E-02	48	20.0

generation and the final generation when the optimizer is terminated. Here, the intermediate generation is chosen as $t + 100$, where t is the generation at which the first non-zero hypervolume is achieved. The table clearly shows that the performance of NSGA-III-IR2-RF is better at the intermediate generation, however the performance is either better, or, at par in the final generation. These results support two important arguments.

- 1) For a problem in which PO solutions can be achieved by NSGA-III (the base EMO/EMaO algorithm), NSGA-III will eventually catch up with NSGA-III-IR2-RF in terms of any performance metric measured. However, there are problems like WFG6, WFG9, etc. where the population may stagnate before achieving the PO solution set. In these problems, NSGA-III-IR2 achieve equivalent or better results than NSGA-III.

- 2) For real-world problems where the available function evaluations are limited, it is important to build an optimizer that can produce a better solution set (more converged) quickly (or, at any intermediate generation). The performance at the intermediate generations, as shown on ZDT, WFG and DTLZ problems, reveals that the proposed learning based-IR2 operator achieves better convergence in most test instances, thus demonstrating its efficacy.

VI. DISCUSSION ON IR2 OPERATOR

Based on the results of multiple two- to five-objective problems, it is evident that algorithms employing the IR2 operator perform much better than the original algorithms in most cases; however, there are some instances where the IR operator does not change the original algorithm's performance.

TABLE VII: Median Hypervolume metric and associated percentage savings in function evaluations (S -metric) on 3- and 4-objective WFG problems, solved using MOEA/D and MOEA/D-IR2-RF. Best performing framework is marked in bold.

Problem- (M)	Generation (t)	MOEA/D (H^t)	MOEA/D-IR2-RF (H^t_R)	p -value	Recovery w/o repair $(t + \Delta t)$	Saving $(S\%)$
WFG1-3	40	0.022009	0.170737	1.23E-05	83	107.5
WFG2-3	40	<i>0.576799</i>	0.582327	5.97E-01	46	15.0
WFG3-3	40	<i>0.430079</i>	0.439573	3.46E-01	47	17.5
WFG4-3	40	0.203112	0.231028	6.11E-06	78	95.0
WFG5-3	40	0.141332	<i>0.132609</i>	3.27E-01	32	-20.0
WFG6-3	40	0.110332	0.138521	6.65E-03	65	62.5
WFG7-3	40	0.150090	0.192004	4.26E-06	134	235.0
WFG8-3	40	0.128467	0.145331	1.95E-02	59	47.5
WFG9-3	40	0.134948	0.188236	1.60E-04	99	147.5
WFG1-4	40	0.069468	0.196936	1.41E-06	156	290.0
WFG2-4	40	<i>0.591580</i>	0.607856	3.46E-01	53	32.5
WFG3-4	40	0.424510	0.452893	6.90E-04	56	40.0
WFG4-4	40	0.271415	0.300305	2.20E-04	68	70.0
WFG5-4	40	0.194265	<i>0.183675</i>	5.22E-01	36	-10.0
WFG6-4	40	0.156274	0.196048	7.90E-04	61	52.5
WFG7-4	40	0.206851	0.249464	2.04E-06	87	117.5
WFG8-4	40	0.162210	0.184629	5.20E-04	63	57.5
WFG9-4	40	0.177231	0.248910	2.05E-05	96	140.0

TABLE VIII: Hypervolume values for DTLZ1–4 ($M=3,4,5$), comparing NSGA-III without repair and with the proposed IR2 repair operator. The intermediate generation is chosen as 100 generations after the first non-zero hypervolume is achieved. Best performing framework is marked in bold and statistically similar results are marked in italics.

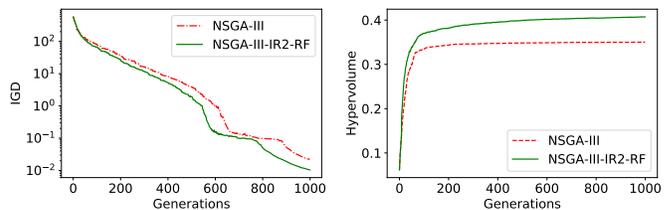
Problem- (M)	Intermediate Generation HV				End Generation HV			
	Generation	NSGA-III	NSGA-III-IR2-RF	p -value	Generation	NSGA-III	NSGA-III-IR2-RF	p -value
DTLZ1-3	550	<i>0.067417</i>	0.315313	3.27E-01	1000	<i>1.001629</i>	1.002444	1.52E-01
DTLZ2-3	103	0.433458	0.438385	1.41E-06	1000	0.447479	<i>0.447471</i>	8.51E-01
DTLZ3-3	687	<i>0.203485</i>	0.361433	5.72E-01	1000	<i>0.434177</i>	0.439590	3.86E-01
DTLZ4-3	109	0.435713	0.439147	3.48E-02	1000	0.447480	<i>0.447469</i>	1.87E-01
DTLZ1-4	443	<i>0.032100</i>	0.299945	1.18E-01	1000	<i>1.009436</i>	1.009460	4.74E-01
DTLZ2-4	100	0.589208	0.601760	1.41E-06	1000	0.618473	<i>0.618437</i>	7.34E-01
DTLZ3-4	643	0.195940	0.534411	5.94E-03	1000	0.606276	0.612803	2.27E-03
DTLZ4-4	107	0.595864	0.609001	1.41E-06	1000	<i>0.618445</i>	0.618466	9.40E-01
DTLZ1-5	386	<i>0.657787</i>	0.830186	4.29E-01	1000	1.009373	<i>1.009368</i>	7.92E-01
DTLZ2-5	100	0.702793	0.721317	1.41E-06	1000	<i>0.743545</i>	0.743554	9.10E-01
DTLZ3-5	585	<i>0.526269</i>	0.567091	3.86E-01	1000	<i>0.736317</i>	0.737042	8.80E-01
DTLZ4-5	105	0.709817	0.733610	1.41E-06	1000	<i>0.743692</i>	0.743728	1.42E-01

There are certain attributes of problems that can make it harder for an EMO/EMaO without learning-based repair to perform well. In the following subsections, we discuss some of these characteristics, where a modified IR2 based algorithms yield considerable improvement, like multi-modality, bias and variable linkages. Further, we discuss the effect of boundary points in many-objective problems, and the effect and importance of early termination of the learning algorithm.

A. Multi-Modality of Pareto-optimal Set

Some test problem problems exhibiting multi-modality are DTLZ3, WFG4 and WFG9 [29]. The generation-wise median IGD plot of DTLZ3 and median hypervolume plot of WFG9 are shown in Figures 4a and 4b, respectively. For DTLZ3, IGD is chosen since the median hypervolume remains zero until almost 600 generations in a 1,000 generation run. In Figure 4a, it is evident that the performance of NSGA-III-IR2-RF is better than that of NSGA-III at all times from start to end. Similarly, Figure 4b shows that the use of the IR2 operator helps NSGA-III converge to a better solution for problem WFG9. It is clear that without the IR2 operator, NSGA-III converges to a smaller hypervolume value.

Motivated from results on these and other problems, we further increase the difficulty of WFG4 problem. The multi-



(a) DTLZ3 ($M = 4$)

(b) WFG9 ($M = 3$)

Fig. 4: Median IGD plot for DTLZ3 and median HV plot for WFG9 (Multi-modal problems).

modality of the problem is dependent on three parameters: A , B and C , whose standard values are 30.0, 10.0 and 0.35, respectively. Earlier study [29] suggested that higher values of A and smaller values of B will create more difficult problems. Hence, we define the new problem as modified-WFG4 with parameter values as 70.0, 5.0 and 0.35. The HV plots for both WFG4 and modified-WFG4 are shown in Figure 5. Both hypervolume values in Figure 5b are comparatively lower than their corresponding values in Figure 5a, but NSGA-III-IR2-RF is able to yield a performance improvement even in the more difficult version of WFG4.

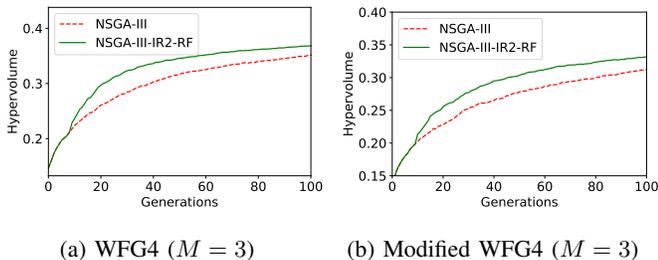


Fig. 5: Median HV plots for WFG4 (with standard and difficult versions).

B. Bias in Pareto-optimal Set

Further, some known problems with bias in the objective space are ZDT6, DTLZ4, and WFG7 [29]. The generation-wise median hypervolume plots for problems ZDT6 and DTLZ4 are shown in Figures 6a and 6b, respectively. The usefulness of the proposed IR2 operator is evident from these figures. Theoretically, it makes sense to have the advantage of a learning-based operator in biased problems, since the learning will help reach the optimal distribution of solutions faster once the right extreme points are identified.

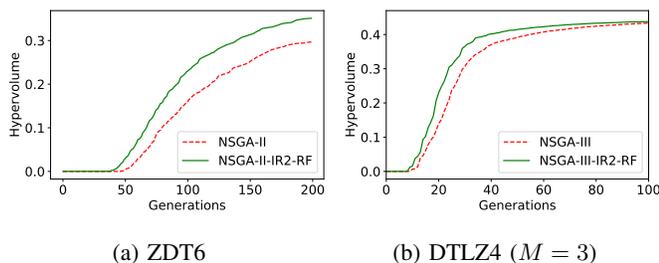


Fig. 6: Median HV plots for biased problems: ZDT6 and DTLZ4.

As done for the multi-modal problem WFG4, the parameters of WFG7 problem are also modified since it has a parameter-dependent bias. The parameter C in the WFG7 problem, which is suggested to be 50 in [29], is now increased to 100. We call this problem with $C = 100$ the modified-WFG7. The HV plots of both versions of WFG7 are shown in Figure 7.

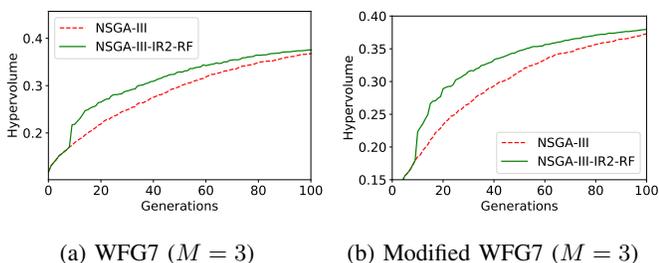


Fig. 7: Median HV plots for WFG7 (with standard and difficult versions).

C. Linked Variable Values in Pareto-optimal Set

It is known that the common genetic operators, like SBX crossover and polynomial mutation operators, do not account for any inter-variable relationships or linkages. A crossover operator that can preserve variable linkages is the operator from *differential evolution* (DE). However, in problems with linked as well as in problems with independent variables, DE is not as efficient in gaining convergence [9]. In both of these cases, the SBX crossover operator with a learning-based IR operator proves to be effective. Two problems L1 and L2 [9] with linked variables are solved using NSGA-II and NSGA-II-IR2-RF. The generation-wise hypervolume plots are presented in Figures 8a and 8b. These plots clearly depict that the IR operator proves effective in preserving the variable linkages.

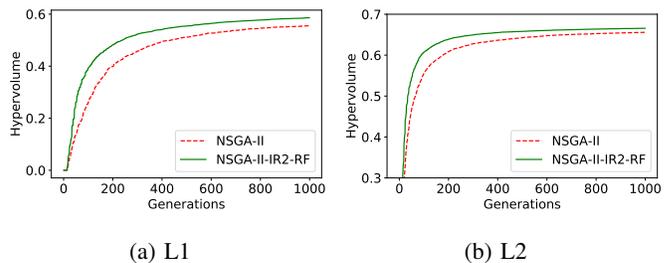


Fig. 8: Median HV plots for L1 and L2 problems with NSGA-II and NSGA-II-IR2-RF.

D. Energy Method for Generating Reference Points

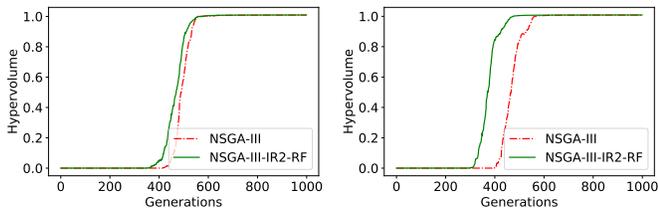
Using the Das-Dennis method to generate reference points (parameters M and p taken from Table II), the percentage of boundary points to the all reference points is tabulated in Table IX. It is evident from the table that the percentage of boundary points escalates quickly with the number of objectives. This can potentially affect the quality of learning as well. In order to explore this effect, the recently-proposed

TABLE IX: Percentage boundary Das-Dennis points with M objectives.

M	2	3	4	5
Boundary Points	2%	37%	70%	93%

energy method [32] is used here to generate reference points and the benefit of learning is compared. Using the energy method in a layer-wise manner, we can control the proportion of boundary points. With $M = 4, 5$ layers of reference points are generated with gaps in each layer as [16, 13, 11, 8, 2] (from outer to inner layer). By this method, 290 points are generated with only 32% boundary points. This reduction from 70% to 32% boundary points (shifting from Das-Dennis to Energy Method) in turn reflects that the proportion of interior points has increased.

It can be seen in Figure 9 that the performance difference is much larger in the case of the energy method compared to the Das-Dennis method, even though the population size is quite similar in both cases (286 and 290 in Figures 9a and 9b, respectively). This motivates us to explore the effect of



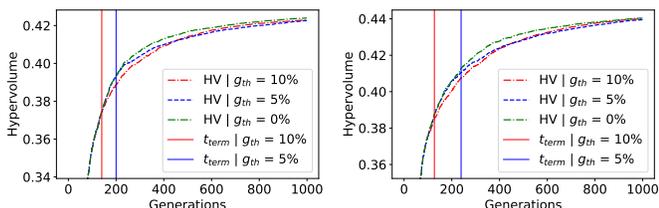
(a) Das-Dennis method. (b) Layer-wise energy method.

Fig. 9: Median HV plots for DTLZ1 ($M = 4$) problem solved by NSGA-III and NSGA-III-IR2-RF using reference points generated by Das-Dennis method and layer-wise energy Method.

boundary points and interior points on quality of learning in future studies.

E. Limited Use of Learning

In this subsection, the effect of terminating the RF-based learning process with a threshold g_{th} and its importance are discussed. Figure 10 shows the results of varying g_{th} to be 10%, 5% and 0% (no termination) on some problems. It can be seen that all three variants are the same until the vertical red line, where learning is terminated and $g_{th} = 10\%$ starts showing a little worse performance than the others. Similarly, after the vertical blue line, the one with $g_{th} = 5\%$ starts performing worse. However, it may be noted that the performance difference is only minor. This is because the learning-based repair does not provide large improvements as the solution nears the Pareto-optimal set. Additional repair will reduce diversity in non-dominated solutions and may not help the search process. Thus, terminating the learning-based repair operator allows original EMO/EMaO algorithms to work well and additionally saves the extra computational effort spent on learning the IR2 operator. Similar trends are observed in other problems as well. However, the early terminations depicted here clearly occur before the beneficial effects of the learning and repair operation are exhausted. At 300-800 generations on either function, the results of continuing learning/repair are found to be beneficial.

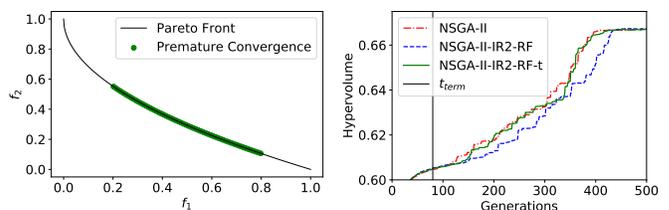


(a) WFG6 ($M = 3$) (b) WFG7 ($M = 3$)

Fig. 10: Limited learning on two WFG problems.

There can be some problems where the idea of termination becomes vital, where the algorithm first converges to only a portion of the PO front and then spreads along it. For example, the constrained imbalance problem CIBN2 [33], for which the

PO front is shown with a solid black line in Figure 11a is such a problem. The green dots represent the premature front where the optimizer converges first. The proposed IR2 operator may not help in this case (after the green-part in Figure 11a is achieved), since there are no representative solutions in other parts of the front to learn from. Also, the repair operator (which learns from the existing good solutions only) may also repair some potentially good offspring in the unexplored regions back to the existing front, which is not favorable. Hence, terminating the repair operator can help in reducing the chances of such performance degradation. Figure 11b shows the median hypervolume plot as the solutions evolve from the premature front to the PO front. It can be seen that the NSGA-II-IR2-RF (without termination) performs worse than NSGA-II, but that NSGA-II-IR2-RF with termination (NSGA-II-IR2-RF-t) performs at par with NSGA-II.



(a) Premature and final Front (b) Hypervolume plot

Fig. 11: Analysis on constrained imbalance problem CIBN2.

VII. CONCLUSIONS

In this paper, we have proposed an enhanced *Innovized* repair operator, named IR2, for potentially improving some of the current offspring solutions, so they become equivalent to or better than the current non-dominated solutions. To fulfill this aim, we have first paired the (archived) solutions from the past few generations with the *best* non-dominated solution found till the current generation, along each reference vector. The dataset constituted by such paired solutions is used to train a Random Forest model which in turn is utilized, as a repair operator. The latter ensures transition of an offspring X to a better converged solution Y .

The proposed IR2 operator has been integrated with NSGA-II, NSGA-III and MOEA/D and tested on multiple benchmark problems ranging from two to five objectives, a total of 63 test instances. The repair shows significant improvements in earlier generations, while the performance is either better or at par towards the end, as compared with the base EMO/EMaO algorithm. Later, this paper has discussed certain characteristics of problems where the proposed IR2 operator yields good results, such as multi-modality, bias, and inter-variable linkages. A restricted use of the learning approach has been found to reduce the computational expense and to improve convergence behavior in certain problems.

This paper has also discussed the effect of boundary points (and interior points) in dealing with many-objective optimization problems. With a recently proposed energy method [32], we have shown that decreasing the proportion of boundary

points or increasing the proportion of interior points while keeping the population size similar can help improve the learning process, and hence, accelerate convergence. In future studies, this work can be extended to single-objective multimodal problems and to discrete- and mixed-variable problems. The effect of distribution of reference points on quality of learning can be explored. It would also be interesting to divide the variables into different “buckets” based on their dependencies, since the machine learning algorithms would then work better with fewer variables. Nevertheless, this extensive study has revealed a new concept using an AI/ML method to improve the performance of EMO/EMaO algorithms for solving complex optimization problems.

ACKNOWLEDGMENT

Authors would like to acknowledge the support provided by the Government of India under SPARC project No. P66. The authors would also thank Blank and Deb [34] for developing the python-based pymoo platform which has been utilized in this research.

REFERENCES

- [1] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.
- [2] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [4] 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.
- [5] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [6] 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.
- [7] M. Pelikan, D. E. Goldberg, and E. Cant-Paz, “The bayesian optimization algorithm,” in *GECCO*. Morgan Kaufmann Publishers Inc., 1999, pp. 525–532.
- [8] M. Luis, G. Jesus, B. Antonio, and M. J. Manuel, “Introducing moneda: scalable multiobjective optimization with a neural estimation of distribution algorithm,” in *Proceedings of Genetic and Evolutionary Computation conference (GECCO-2008)*, 2008, pp. 689–696.
- [9] S. Mittal, D. Saxena, K. Deb, and E. Goodman, “An ANN-assisted repair operator for evolutionary multiobjective optimization,” Department of Electrical and Computer Engineering, Michigan State University, East Lansing, USA, Tech. Rep. COIN Report No. 2020005, 2020. [Online]. Available: <https://www.egr.msu.edu/kdeb/papers/c2020005.pdf>
- [10] A. Gaur and K. Deb, “Effect of size and order of variables in rules for multi-objective repair-based innovization procedure,” in *Proceedings of Congress on Evolutionary Computation (CEC-2017) Conference*. Piscataway, NJ: IEEE Press, 2017.
- [11] K. Deb and R. Datta, “Hybrid evolutionary multi-objective optimization and analysis of machining operations,” *Engineering Optimization*, vol. 44, no. 6, pp. 685–706, 2012.
- [12] A. Ghosh, E. Goodman, K. Deb, R. Averill, and A. Diaz, “A large-scale bi-objective optimization of solid rocket motors using innovization,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.
- [13] S. Bandaru and K. Deb, “Towards automating the discovery of certain innovative design principles through a clustering based optimization technique,” *Engineering optimization*, vol. 43, no. 9, pp. 911–941, 2011.
- [14] C. Dudas, A. H. Ng, L. Pehrsson, and H. Bostrom, “Integration of data mining and multi-objective optimisation for decision support in production systems development,” *International Journal of Computer Integrated Manufacturing*, no. ahead-of-print, pp. 1–16, 2013.
- [15] S. Bandaru and K. Deb, “A dimensionally-aware genetic programming architecture for automated innovization,” in *Proceedings of the Seventh International Conference on Evolutionary Multi-Criterion Optimization (EMO-13), LNCS 7811*. Heidelberg: Springer, 2013, pp. 513–527.
- [16] S. Mittal, D. K. Saxena, and K. Deb, “Learning-based multi-objective optimization through ANN-assisted online innovization,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO 20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 171–172. [Online]. Available: <https://doi.org/10.1145/3377929.3389925>
- [17] M. Pelikan, D. Goldberg, and F. Lobo, “A survey of optimization by building and using probabilistic models,” *Computational Optimization and Applications*, vol. 21, no. 1, pp. 5–20, 2002.
- [18] L. Li, H. Chen, and C. L. et al., “A robust hybrid approach based on estimation of distribution algorithm and support vector machine for hunting candidate disease genes,” *The Scientific World Journal*, vol. 2013, no. 393570, p. 7, 2013.
- [19] S. Salcedo-Sanz, “A survey of repair methods used as constraint handling techniques in evolutionary algorithms,” *Computer Science Review*, vol. 3, no. 3, pp. 175 – 192, 2009.
- [20] C. Zhang, J. Chen, and B. Xin, “Distributed memetic differential evolution with the synergy of lamarckian and baldwinian learning,” *Applied Soft Computing*, vol. 13, no. 5, pp. 2947 – 2959, 2013.
- [21] Y. Qi, F. Liu, M. Liu, M. Gong, and L. Jiao, “Multi-objective immune algorithm with baldwinian learning,”

- Applied Soft Computing*, vol. 12, no. 8, pp. 2654 – 2674, 2012.
- [22] K. Deb, “Unveiling innovative design principles by means of multiple conflicting objectives,” *Engineering Optimization*, vol. 35, no. 5, pp. 445–470, 2003.
- [23] K. Deb and A. Srinivasan, “Innovization: Innovating design principles through optimization.” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, New York: ACM, 2006, pp. 1629–1636.
- [24] I. Das and J. Dennis, “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM Journal of Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [25] K. Deb, S. Bandaru, and H. Seada, “Generating uniformly distributed points on a unit simplex for evolutionary many-objective optimization,” in *Proceedings of the Tenth International Conference on Evolutionary Multi-Criterion Optimization (EMO-19)*, LNCS 11411. Heidelberg: Springer, 2019, pp. 179–190.
- [26] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple Criteria Decision Making Theory and Applications*, G. Fandel and T. Gal, Eds. Berlin: Springer-Verlag, 1980, pp. 468–486.
- [27] N. Padhye, K. Deb, and P. Mittal, “Boundary handling approaches in particle swarm optimization,” in *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*. *Advances in Intelligent Systems and Computing*, J. Bansal, P. Singh, K. Deep, M. Pant, and A. Nagar, Eds., vol. 201. Springer, India, 2013, pp. 287–298.
- [28] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000. [Online]. Available: <https://doi.org/10.1162/106365600568202>
- [29] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London: Springer London, 2005, pp. 105–145. [Online]. Available: https://doi.org/10.1007/1-84628-137-7_6
- [31] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, “Performance comparison of nsga-ii and nsga-iii on various many-objective test problems,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 3045–3052.
- [32] 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 Transactions on Evolutionary Computation*, pp. 1–1, 2020.
- [33] J. Lin, H. Liu, and C. Peng, “The effect of feasible region on imbalanced problem in constrained multi-objective optimization,” in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 2017, pp. 82–86.
- [34] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.