2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

On the Choice of Unique Identifiers for Predicting Pareto-optimal Solutions using Machine Learning

Anirudh Suresh
Department of Mechanical Engineering
Michigan State University
East Lansing, MI 48824, USA
suresha2@msu.edu

Kalyanmoy Deb
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824, USA
kdeb@msu.edu

***Abstract*—Incomplete or sparse non-dominated fronts are unavoidable in multi-objective optimization due to complexity of problems, morphology of Pareto optimal fronts, and stochasticity involved in evolutionary optimization algorithms. It is pragmatic to develop methods that can alleviate some of these issues after the optimization run is complete, without the need for re-optimization or additional solution evaluations. Previously developed methods demonstrated that it is possible to predict Pareto-optimal solutions from *pseudo-weight* vectors using Gaussian Process Regression (GPR) models. We extend the GPR-based method to predict new Pareto-optimal solutions using reference vectors as unique identifiers and demonstrate that like the pseudo-weight vectors, reference vectors can also used instead in learning the association between identifiers and corresponding variable vectors. Results on many test problems indicate that the choice of a suitable identifier makes a large impact on the decision-making process, particularly for visualizing the newly created non-dominated (ND) solutions. In this study, we discuss the advantages and disadvantages of using pseudo-weights and reference vectors as unique identifiers for ND solutions, paving the way to devise further identifiers for predicting new Pareto-optimal solutions.**

***Index Terms*—machine-learning, multi-objective optimization, optimization, evolutionary algorithm, multi-criterion decision making**

I. Introduction

The desired outcome of any multi- or many-objective optimization (M(a)OO) run is a uniformly distributed set of non-dominated (ND) solutions that approximate the Pareto-optimal (PO) front. Though decades of research have led to several robust and reliable algorithms [1]–[6], an incomplete or sparse non-dominated (ND) front is unavoidable in practice during M(a)OO. This could be due to several reasons, such as the inherent stochasticity of evolutionary algorithms, complexity of problems, morphology of PO front, noise in objective evaluation, local attractors, dimensionality of variable space etc. Another common reason is the finite number of solution evaluations allowed during the optimization run. The No-Free-Lunch (NFL) theorem dictates that no single algorithm will be able to solve all these issues even with continual research efforts to address them. The impact of these issues can be felt during decision-making (DM) step as the decision-maker might desire solutions in the gaps or sparse regions. In order to choose a reasonable solution, the decision maker needs to know if the gaps and sparsity are due to the nature of the problem or a failure of the M(a)OO method used. This issue is typically dealt with re-optimization using reference-direction based methods [7], [8] or running the M(a)OO method multiple times and aggregating the results. Some gap-finding algorithms have also been proposed in literature [9]. These approaches are usually tedious and require several more solution evaluations that might not be allowed. Since decision-making is usually not a single-step process, it is pragmatic to have methods that can *generate* PO solutions, as desired by the DM, without re-optimization or new solution evaluations.

Previous works [10] have shown that it is possible to predict PO solutions from *pseudo-weight* vectors (Equation 1). A pseudo-weight vector $\mathbf{w} \in \mathbb{R}^M$ (of dimension to the number of objectives (M) and $\sum_{i=1}^{M} w_i = 1$) is a unique identifier to a single Pareto-optimal solution vector $\mathbf{x}^{\mathbf{w}} \in \ltimes$ of size n. For example, in a two-objective case, the pseudo-weight vector $(0, 1)$ identifies the extreme Pareto-optimal (PO) solution having the best f_2 value and the worst f_1 value. The pseudo-weight vector $(0.5, 0.5)$ represents the unique intermediate Pareto-optimal solution lying in the mid-way in the range between ideal and nadir objective values. In a previous study [10], machine learning (ML) methods were employed to learn the mapping between pseudo-weight vectors ($\mathbf{w}$-vectors) and respective decision variables ($\mathbf{x}$-vectors) of PO solutions. Although the process is somewhat apparently similar to the inverse mapping studies in predicting $\mathbf{x}$ from $\mathbf{f}$, the study indicated a number of advantages of the pseudo-weight based approach. Previous results indicate that given a *new* pseudo-weight vector that is not in the training set of the development of the ML model, the trained ML model can predict the decision variables of the corresponding PO solution with reasonable accuracy. Hence, such a trained ML model developed from evolutionary multi-objective optimization (EMO) algorithms can be used to (i) fill apparent gaps or less dense areas in the obtained non-dominated (ND) front found by an EMO algorithm, (ii) test the extreme PO points, and (iii) provide reasons for gaps/sparse areas of the PO front. The most attractive aspect of the proposed ML-based approach is that it avoids performing another optimization process, such as a reference-point based EMO method (such as R-NSGA-II [7] or R-NSGA-III [8]) involving further post-optimal solution evaluations.

However, pseudo-weight vectors may not be the only way to uniquely identify each solution in the PO front. Reference vectors (RV) $\mathbf{r}$, lying on the M-dimensional unit simplex ($\sum_{i=1}^{M} r_i = 1$), have been extensively used in MOO literature [2], [5] to characterize solutions during EMO runs. Giagkiozis and Fleming [11] used Radial Basis Neural Networks (RBNN) [12] to learn the mapping between reference vectors and decision variables. Similarly, Takagi et al. [13], [14] used Kriging [15] and RBNNs to map reference vectors to decision variables and objective functions. These reference vectors represent the geometric location of the PO solution on the unit simplex. In this study, we extend the rigorous analysis done in [10] to predict PO solutions from reference vectors and study the prediction capabilities and compare with the pseudo-weights approach. We experimentally demonstrate that the ML based learning of the mapping process is agnostic to the choice of unique identifiers or placeholders for PO solutions and discuss the implications of the choice of these identifier vectors.

Though both pseudo-weights and reference vectors facilitate the learning between PO solution identifier vectors and respective decision variables, the information they convey is very different. Pseudo-weights convey priorities and might be of relevance if the DM does not know the exact region in the PO front to sample new solutions from. On the other hand, pseudo-weights are not intuitive from the perspective of understanding the geometry of the PO front. Reference vectors keep the geometric structure of the PO front more closely than the pseudo-weights. If the DM knows the exact regions where they desire new solutions, the RV-based approach may be more relevant. In this study, we attempt to explain the advantages and disadvantages of the choice of these identifier vectors by applying them on well-known multi-objective test problems [16], [17], as examples.

The remainder of the paper is arranged as follows. In Section II, we describe the different choices we have for uniquely identifying ND solutions and the implications of the choice of these identifiers for learning ML mapping from the perspective of decision-making. In Section III, we demonstrate that this ML-based mapping is agnostic to the choice of identifier and that reference vectors can be equally effective placeholders as pseudo-weights. Finally, in Section IV, we summarize our findings and present several future directions for this work.

II. Identifiers for ND Solutions

Given a set of ND solutions, a convenient method to characterize each solution on the approximated PO front is to use pseudo-weights [18] with $\sum_{i=1}^{M} w_i^{(k)} = 1$ for k-th PO solution:

$$w_i^{(k)} = \frac{\left(f_i^{\max} - f_i^{(k)}\right) \Big/ \left(f_i^{\max} - f_i^{\min}\right)}{\sum_{j=1}^{M} \left(f_j^{\max} - f_j^{(k)}\right) \Big/ \left(f_j^{\max} - f_j^{\min}\right)}. \tag{1}$$

These vectors describe each solution's normalized distance to the worst solution with respect to each objective. Each pseudo-weight vector is unique to a particular solution and can act as a unique identifier for each region on the PO front. As mentioned before, components of this vector sum to one and represent *priorities* (not to be confused with weighted-sum approach) for optimization. This can be convenient for DMs as they might not know the exact objective values they want but might be able to describe their desired solutions in the form of desired priorities of objectives and hence find the corresponding PO solution. This property has been exploited in our previous work [10] to train ML models to predict $\mathbf{x}$-vectors based on their pseudo-weights. It has been demonstrated that this mapping between pseudo-weights and decision variables is learnable and can be an effective tool during decision-making process. These trained models can be used to fill gaps or sparse regions in PO fronts, explore extremes of PO front, identify reasons (dominance or infeasibility) for pseudo-weight to not associate a PO solution, or simply create more new PO solutions for the sake of visualization or other purposes. All these are achieved without any additional solution evaluations.

Though every PO solution can be assigned a pseudo-weight vector based on their $\mathbf{f}$-values, there is no guarantee that every pseudo-weight vector has a PO solution. During DM, common solutions to this issue are to either choose a solution from available pseudo-weight vectors or take the closest available solution if the pseudo-weight does not exactly correspond to a PO solution. This process becomes tricky and less intuitive at more than two objectives as certain desired *priorities* might not exist in the PO front and the meaning of *closest available pseudo-weight* is not readily understandable. Though these vectors sum to one they do not always span the whole unit simplex and can conceal critical information regarding the geometry of the PO front. The range of available pseudo-weight values is dependent on the range of objective values of the PO front and hence cannot be known prior to optimization.

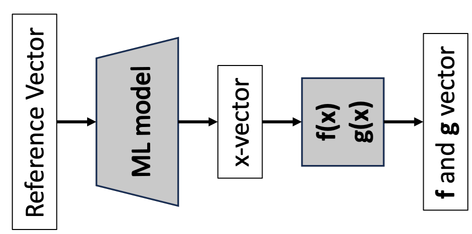


Fig. 1: ML-assisted method to predict ND solutions from reference vectors.

An alternative identifier for each PO solution can be their respective reference vectors or the reference points on the unit simplex. These vectors capture the geometry of the location of a solution on the PO front and have been regularly used in MOO algorithms [2], [3], [5] with $\sum_{i=1}^{M} r_i^{(k)} = 1$ for every k:

$$r_i^{(k)} = \frac{\left(f_i^{(k)} - f_i^{\min}\right) / \left(f_i^{\max} - f_i^{\min}\right)}{\sum_{j=1}^{M} \left(f_i^{(k)} - f_i^{\min}\right) / \left(f_i^{\max} - f_i^{\min}\right)}. \tag{2}$$

Unlike pseudo-weights, a solution corresponding to $\mathbf{r} = (1, 0)$

would represent the worst PO solution for f_1 and the best PO solution for f_2. Conveniently, the intuition behind this relationship also holds at higher dimensions. Despite their inverse characteristics, the mathematical relationship among $w_i^{(k)}$ and $r_i^{(k)}$ is not independent of objective values, making it less intuitive to relate one to the other exactly:

$$w_i^{(k)} = r_i^{(k)} \frac{1 - \widetilde{f}_i^{(k)}}{M r_i^{(k)} - \widetilde{f}_i^{(k)}}, \text{ and } r_i^{(k)} = w_i^{(k)} \frac{\widetilde{f}_i^{(k)}}{M w_i^{(k)} + \widetilde{f}_i^{(k)} - 1},$$

where $\widetilde{f}_i^{(k)}$ is the normalized value of $f_i^{(k)}$. We explain the difference between pseudo-weights and RVs using four cases by plotting 600 solutions on both the identifier and the objective space.

- In Figure 2a, the orange points represent the reference vectors corresponding to PO front of 3-objective DTLZ2 problem. As expected, the RVs cover the whole of the unit simplex as DTLZ2 has a continuous PO front. These RVs not only demonstrate the availability of solutions at different regions of the objective space, but also provide a geometrically intuitive platform for navigating the PO front. On the other hand, the pseudo-weight vectors, represented by the blue points, are aggregated in the center as a smoothed inverted triangle. According to the pseudo-weights computed, there exists no PO solution outside the blue points even though the PO front (Figure 2d) covers the whole first quadrant. This might be a cause of confusion from the perspective of a decision maker as *regions* desired by the DM might not exist in the pseudo-weight space.
- Figure 2b shows the RVs and pseudo-weight vectors of the PO front of 3-objective DTLZ7. As expected, similar to the actual PO front (Figure 2e), RVs and pseudo-weights are also in the form of four islands. However, the islands formed by the pseudo-weights (blue points) in Figure 2b are of different sizes, unlike the corresponding RVs. This can be a cause of confusion for the DMs as the pseudo-weights misrepresents the geometry of the PO front. On the other hand, the RVs described by the orange points capture the geometry information. From the perspective of *priorities*, the difference in the size of the islands indicates the sensitivity of navigating the PO front. While this does not correlate with the size of islands in the objective space, this sensitivity information can be vital during decision-making.
- Figure 2c shows RVs and pseudo-weights of PO front of 3-objective car-side impact problem. This PO front is continuous, as shown in Figure 2f and this fact is reflected in both RVs and pseudo-weights. An interesting observation in Figure 2f is the fact that the points are dense close to the minimum of f_2 and f_3. However, this leads to uniformly distributed RVs that misrepresent the availability of solutions on the PO front, unlike pseudo-weights where the solutions are denser on one side. This shows that in some cases, pseudo-weights might be better identifiers for PO solutions.
- Figure 3b shows PO front of 3-objective C2DTLZ2, a constrained problem where the PO front is disconnected by infeasible regions. Similar to DTLZ2 case, corresponding RVs span the entire unit simplex while pseudo-weights form a cluster in the middle of the simplex as shown in Figure 3a. The confusion caused by this disparity is further exacerbated by the infeasible regions (or gaps) in pseudo-weight space that are present on the opposite side compared to the PO solutions or RVs. For a problem like C2DTLZ2, clearly, RVs can be better identifiers.

To demonstrate the mirroring aspect of pseudo-weights and RVs, a random solution is highlighted with a star-outline (with blue and orange shading) in Figure 2 and Figure 3. We can observe that the corresponding pseudo-weight is on the opposite side compared to RV and $\mathbf{f}$-vector.

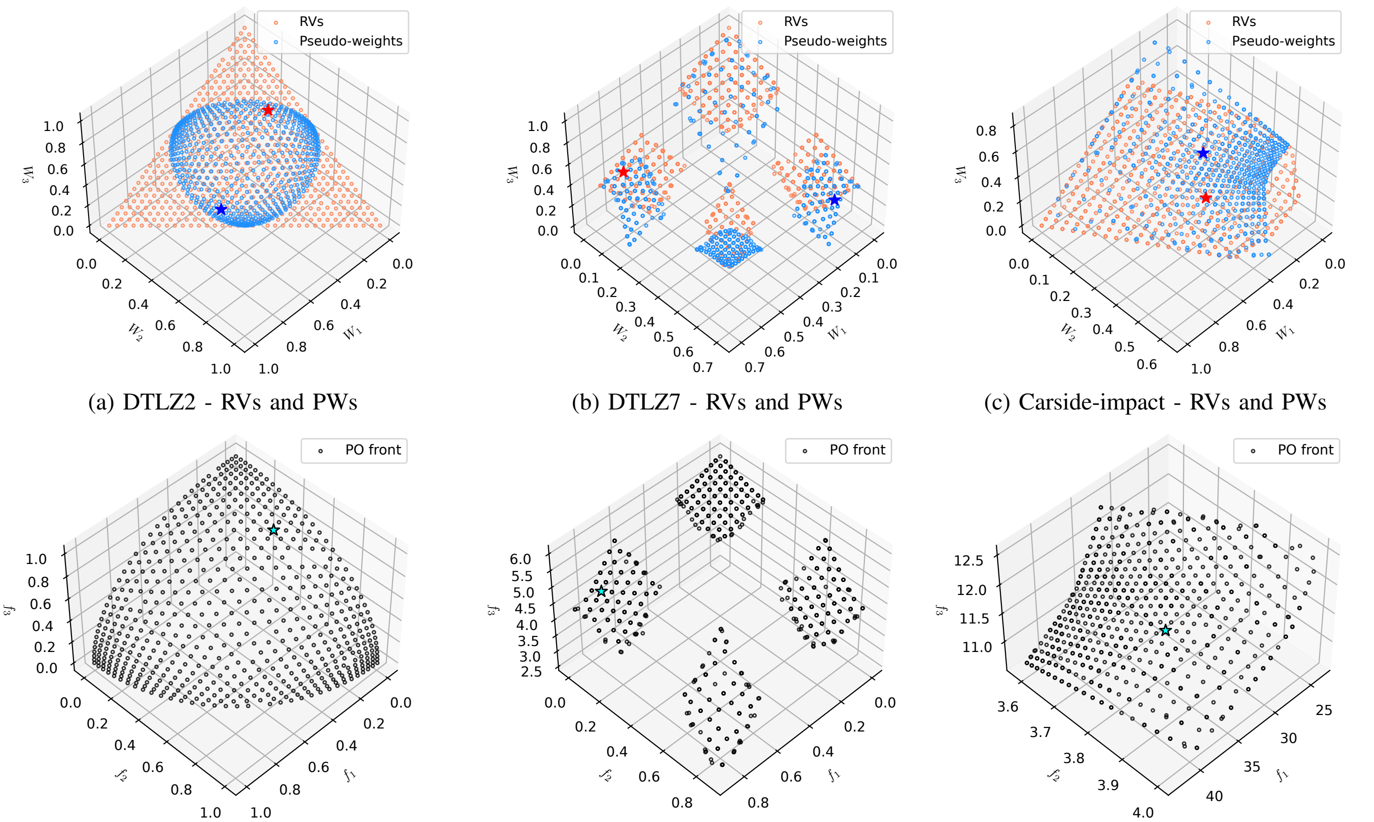
Clearly, reference vectors provide some advantages compared to pseudo-weights with respect to the interpretation of the geometry of the PO front. In this study, we show that pseudo-weights can be replaced by reference vectors for learning the mapping between identifiers and decision variables. While curating the dataset for ML training, we compute corresponding reference vectors by normalizing the objective values and then dividing each objective vector by their sum, as shown in Equation 2. These points represent the intersection of the line joining normalized $\mathbf{f}$-vector and origin, and the unit simplex hyperplane.

III. Pareto Estimation

Here, we emulate a DM setting at the end of optimization, as shown in Figure 1, by using PO solutions as training set for the ML method and removing specific points and using them as test set in order to understand the prediction capabilities of the method. We consider three such scenarios:

- **random** gap where the test set is uniformly distributed on the PO front, as if the DM desires a new solution from anywhere on the PO front. Here, we assume the existing solutions are well spread out and the DM desires a solution at a point where there is no existing solution.
- **continuous** gap where the test set contains a continuous set of points from the middle of the PO front. This simulates a scenario where the PO front contains a gap and the DM desires solutions in the gap region.
- **edge** case where the continuous test set is sampled from one of the edges of PO front. This solves the task of extending the PO front on any side of the existing solutions based on curiosity from the DM.

For all problems, the Gaussian Process Regression (GPR) based modeling is used with $110M$ points used for training and $10M$ points used as test set, where M is the number of objectives in the problem. For the modeling purpose, a grid search of mean and regression functions are evaluated and the best model is chosen. We present results from multi- (ZDT [19], crashworthiness [20]) and many-objective (DTLZ [16]) problems as well as constrained (BNH, OSY, C2DTLZ2,



(a) DTLZ2 - RVs and PWs
(b) DTLZ7 - RVs and PWs
(c) Carside-impact - RVs and PWs
(d) DTLZ2 - $\mathbf{f}$-vectors
(e) DTLZ7 - $\mathbf{f}$-vectors
(f) Carside-impact - $\mathbf{f}$-vectors

Fig. 2: Identifiers and their corresponding objective values. One random objective vector is marked in both spaces.

carside-impact [17]) problems. We perform all experiments 31 times and present Mean Absolute Errors (MAE) of predicted $\mathbf{x}$-vectors, scaled by their variable ranges, and MAE of corresponding $\mathbf{f}$-vectors (on evaluating the predicted decision variables), scaled based on theoretical PO values.

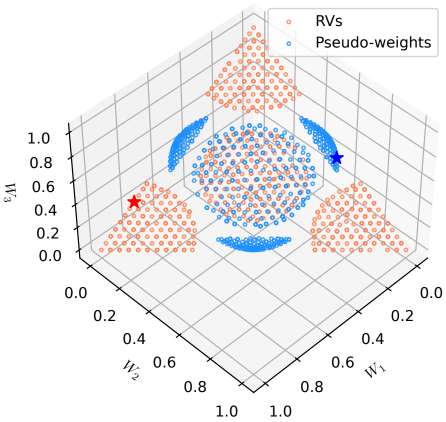
Table I shows MAE of predicted $\mathbf{x}$-vectors and their corresponding (evaluated) $\mathbf{f}$-vectors. We can see that the error values are extremely low for 2 objective ZDT problems. As expected, the *edge* case errors are higher than *random* and *continuous* case as extrapolation is a harder ML task than interpolation. This pattern is also seen in BNH and OSY problems where the error values are higher but low enough to be considered a successful mapping between reference vectors and decision variables.

Similarly, we observe from Table I that many-objective problems also have low prediction errors, competitive with pseudo-weights approach [10]. Here, since the output is the decision variables, the complexity of the mapping problem does not increase while scaling with respect to objectives. Figure 6a shows 3-objective DTLZ2 with a continuous gap in the middle of the PO front and Figure 6b shows 3-objective DTLZ7 with random solutions used as test set. In both these figures, we can observe that the $\mathbf{f}$-vector evaluated from predicted $\mathbf{x}$-vector is sufficiently close to the target boxes, reflective of the error values in Table I.

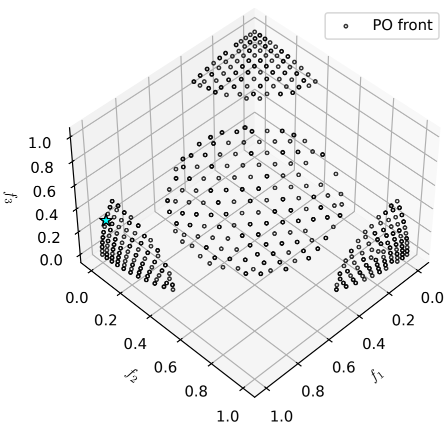
Extending this to 5- and 10-objective DTLZ2 and C2DTLZ2 problems, we observe low errors in both $\mathbf{x}$-vector prediction as well as evaluated $\mathbf{f}$-vector prediction. Figure 7 shows Parallel Coordinate Plot (PCP) for 10-objective DTLZ2 problem with the training and generated $\mathbf{f}$-vectors for *random* gap scenario. We observe that the generated objective vectors have a low deviation from the actual PO front.

IV. Conclusions and Future Directions

In this study, we have demonstrated that reference vectors (RVs), commonly used in EMaO algorithms, can be used to predict PO solutions, just as well as pseudo-weights, using GPR-based ML models. RVs and pseudo-weights provide different advantages during decision-making and need to be chosen carefully based on the requirements of the decision-maker. RVs convey information regarding the geometry of the PO front and can be an effective tool during decision-making if the DM already knows what part of the PO front they desire solutions from. On the other hand, pseudo-weights provide a *priorities* perspective on choosing solutions and can be useful when the DM does not know the exact region. In these ML-based DM methods, RVs or pseudo-weights are merely placeholders and the mapping between them and decision variables is tractable owing to regularity properties of PO frontiers. Similar to RVs, other unique identifiers can also be used here and would be a fascinating future direction as the

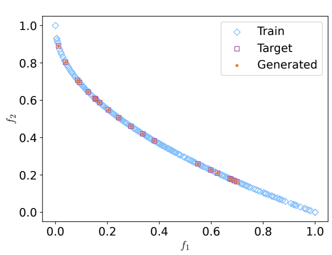


(a) RV and PW vectors for C2DTLZ2

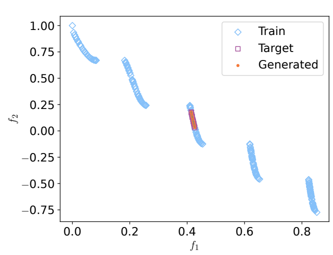


(b) Corresponding $\mathbf{f}$-vectors

Fig. 3: Reference vectors, pseudo weights and corresponding $\mathbf{f}$-vectors of PO front of C2DTLZ2.



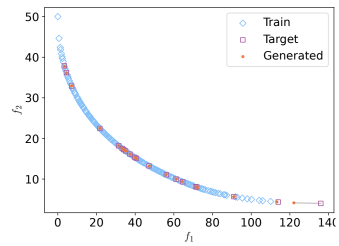
(a) ZDT1 with random test points



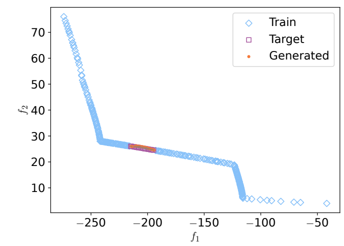
(b) ZDT3 with cont. test points

Fig. 4: Predicting PO solutions from RV for two-objective ZDT1 and ZDT3 problems.

choice of these identifiers can provide a range of advantages from the perspective of MCDM. The choice of a unique identifier is crucial to a successful decision-making process and this study has shown that different identifiers can be learned for the purpose. An interesting future study can be to identify other unique identifiers that can be effective at much

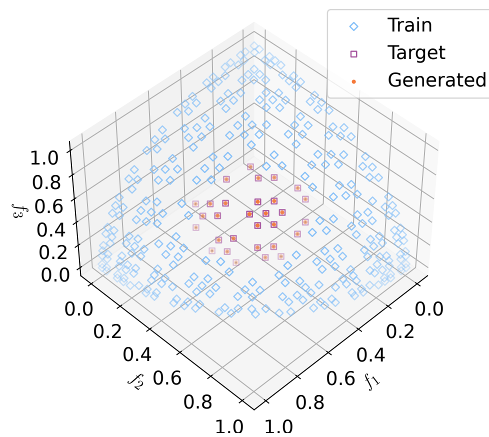


(a) BNH with random test points

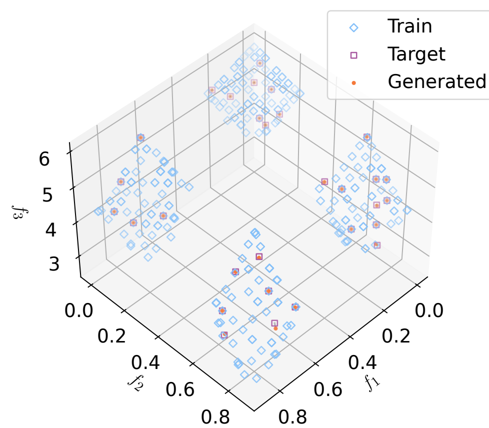


(b) OSY with cont. test points

Fig. 5: Predicting PO solutions from RV for two-objective ZDT1 and ZDT3 problems.



(a) DTLZ2 with cont. test points



(b) DTLZ7 with random test points

Fig. 6: PO solution prediction for 3-objective DTLZ problems.

higher dimensions when coupled with advanced visualization methods like PalletteViz [21]. One important aspect of the ML-learnt models is that optimization-based methods are not needed for filling gaps or providing more PO solutions in sparse regions causing additional solution evaluations.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

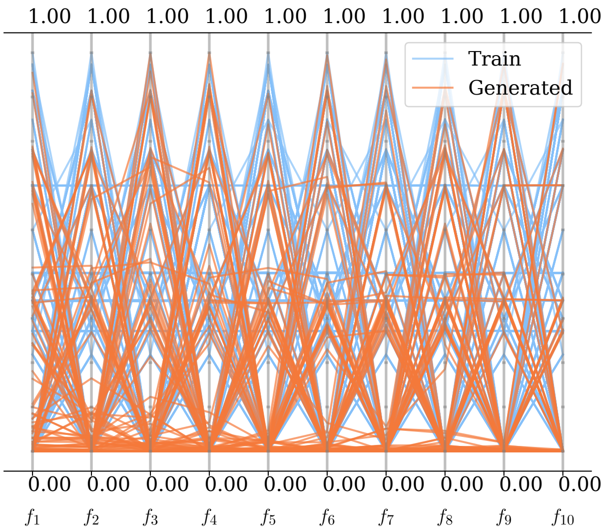


Fig. 7: Training and generated $\mathbf{f}$-vectors for 10-objective DTLZ2 for random test points

TABLE I: Scaled mean absolute errors in $\mathbf{x}$ and $\mathbf{f}$.

problem	M	Gap Type	MAE in x	MAE in f
ZDT1	2	random	8.417E-08	3.796E-05
		continuous	1.099E-07	2.700E-06
		edge	2.603E-04	5.909E-03
ZDT2	2	random	1.702E-11	5.583E-10
		continuous	3.512E-11	1.123E-09
		edge	1.337E-07	5.971E-06
ZDT3	2	random	4.642E-06	6.091E-04
		continuous	2.257E-04	9.155E-03
		edge	1.702E-04	2.242E-02
BNH	2	random	2.289E-03	1.083E-03
		continuous	1.999E-03	4.776E-04
		edge	1.062E-01	1.317E-01
OSY	2	random	2.386E-03	5.382E-03
		continuous	5.859E-03	6.229E-02
		edge	4.335E-02	7.072E-02
DTLZ2	3	continuous	6.469E-04	6.547E-04
		random	6.160E-04	1.849E-04
DTLZ5	3	continuous	3.134E-02	3.527E-04
		random	3.493E-02	1.848E-04
DTLZ7	3	continuous	1.332E-02	6.984E-02
		random	3.015E-04	1.901E-03
Carside	3	continuous	2.419E-02	1.959E-02
		random	1.417E-02	7.899E-03
Crashworthiness	3	continuous	1.001E-01	2.913E-02
		random	1.118E-02	6.492E-03
C2DTLZ2	3	continuous	6.948E-04	4.243E-04
		random	7.931E-04	1.841E-04
DTLZ2	5	continuous	4.766E-03	7.850E-03
		random	7.561E-03	5.722E-03
C2DTLZ2	5	continuous	1.113E-02	2.570E-02
		random	1.428E-02	1.719E-02
DTLZ2	10	continuous	3.369E-02	1.861E-02
		random	5.353E-02	1.375E-02
C2DTLZ2	10	continuous	7.312E-02	9.146E-02
		random	7.451E-02	5.430E-02

[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] 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.

[4] H. Seada and K. Deb, "U-NSGA-III: A unified evolutionary optimization pro- cedure for single, multiple, and many objectives – proof-of-principle results," in *Proceedings of Eighth Conference on Evolutionary Multi-Criterion Optimization (EMO-2015), Part II*. Heidelberg: Springer, 2015, pp. 34–49.

[5] 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.

[6] K. Deb, C. L. do Val Lopes, F. V. C. Martins, and E. F. Wanner, "Identifying pareto fronts reliably using a multi-stage reference-vector-based framework," *IEEE Transactions on Evolutionary Computation*, in press.

[7] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 635–642.

[8] Y. Vesikar, K. Deb, and J. Blank, "Reference point based NSGA-III for preferred solutions," in *IEEE Symposium Series on Computational Intelligence (SSCI-2018)*, 2018.

[9] 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*, 2020, pp. 578–586.

[10] K. Deb, A. Gondkar, and A. Suresh, "Learning to predict Pareto-optimal solutions from pseudo-weights," in *Proceedings of the Evolutionary Multi-Criterion Optimization (EMO-2023) Conference*. Springer, in press.

[11] I. Giagkiozis and P. J. Fleming, "Pareto front estimation for decision making," *Evolutionary computation*, vol. 22, no. 4, pp. 651–678, 2014.

[12] J. Liu, *Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation*. Springer Science & Business Media, 2013.

[13] T. Takagi, K. Takadama, and H. Sato, "Supervised multi-objective optimization algorithm using estimation," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[14] ——, "Pareto front upconvert by iterative estimation modeling and solution sampling," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2023, pp. 218–230.

[15] M. von Stein, "Interpolation of spatial data: Some theory for kriging," 2012.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.

[17] L. Gu, R. Yang, C.-H. Tho, M. Makowskit, O. Faruquet, and Y. L. Y. Li, "Optimisation and robustness for crashworthiness of side impact," *International journal of vehicle design*, vol. 26, no. 4, pp. 348–360, 2001.

[18] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[19] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 06 2000.

[20] X. Liao, Q. Li, W. Zhang, and X. Yang, "Multiobjective optimization for crash safety design of vehicle using stepwise regression model," *Structural and Multidisciplinary Optimization*, vol. 35, pp. 561–569, 2008.

[21] A. K. A. Talukder and K. Deb, "PaletteViz: A visualization method for functional understanding of high-dimensional Pareto-Optimal data-sets to aid multi-criteria decision making," *IEEE Computational Intelligence Magazine*, vol. 15, no. 2, pp. 36–48, 2020.