Machine Learning-Based Decision-Maker For Benchmarking Interactive Multi-Criterion Decision-Making Procedures

COIN Report 2026002

Deepanshu Yadav
Michigan State University
East Lansing, Michigan, USA
deepansh@msu.edu

Kalyanmoy Deb
Michigan State University
East Lansing, Michigan, USA
kdeb@egr.msu.edu

Abstract

Interactive multi-criterion decision-making (iMCDM) procedures rely on a human decision-maker (DM) to iteratively provide preferences, so scalarized subproblems can move toward a preferred Pareto-optimal solution. This human-in-the-loop nature makes systematic benchmarking difficult, as preference information and interaction pattern vary across individuals and problems. To overcome this limitation and to enable computationally-oriented researchers to contribute more profoundly in the MCDM field, we introduce a Machine-based Decision Maker (Machine-DM) that replaces human DMs with pre-trained machine learning models capable of performing the key iMCDM tasks automatically. The Machine-DM predicts objective classifications, such as which objectives should be improved, relaxed, fixed, or allowed to vary and generates corresponding bounding parameters without requiring knowledge of the true target preferred solution. Using this Machine-DM, we develop machine-based versions of four well-known iMCDM procedures: STEM, GUESS, STOM, and NIMBUS. We further propose a set of performance metrics designed to evaluate performance of these iMCDM procedures. Using the Machine-DM concept we also propose a Bench-iMCDM framework for benchmarking iMCDM procedures. Applications to test and engineering problems demonstrate the usefulness of Machine-DM and highlight its potential to serve as a unified framework for comparing a broad class of iMCDM procedures and also to develop new ones.

Keywords

Multi-criterion Decision-making, Machine Learning, Artificial Neural Network, Benchmarking, Evolutionary Algorithms.

ACM Reference Format:
Deepanshu Yadav and Kalyanmoy Deb. . Machine Learning-Based Decision-Maker For Benchmarking Interactive Multi-Criterion Decision-Making Procedures COIN Report 2026002 . In . ACM, New York, NY, USA, 15 pages.

1 Introduction

Real-world applications often involve multiple conflicting objectives [6], which are typically formulated as multi-objective optimization (MOO) and many-objective optimization (MaOO) problems in the literature [8, 32]. Evolutionary multi-objective optimization (EMO) algorithms are designed to handle real-world problems involving multiple, conflicting objectives and numerous constraints [8, 11]. In the EMO literature, both MOO and MaOO algorithms [13, 16, 39] aim to generate a set of non-dominated (ND) or near Pareto-optimal (PO) solutions, each of which is equally good in the absence of explicit *preferences* from a decision maker (DM). Consequently, from a practical standpoint, the outcomes of MOO/MaOO algorithms require an additional decision-making step to identify the most preferred solution.

Multi-criterion decision-making (MCDM) procedures provide a structured way for human DMs to supply and refine their preferences—either once or iteratively—to select a preferred solution from a set of ND solutions [26]. These procedures are typically grouped into three categories depending on when the preference information is introduced [31]: (i) *a priori* methods, where preferences are specified before optimization begins; (ii) *a posteriori* methods, where preferences are expressed after a set of solutions has been produced; and (iii) *interactive* methods, which collect and update preferences throughout the optimization process.

Interactive MCDM (iMCDM) procedures iteratively gather the DM's preferences during optimization to steer the search toward desirable regions of the Pareto front [38]. Such preferences—often expressed through search directions, aspiration levels, or step sizes—are typically utilized within a single-objective achievement scalarizing function (ASF). Several classical iMCDM methods follow this scalarization-driven framework, including NIMBUS [27], Pareto-Race [21, 22], Pareto Navigator [18], GUESS [5], the Step Method (STEM) [1], the Satisficing Trade-Off Method (STOM) [30], Surrogate Worth Trade-Off (SWT) [19], and other iterative ASF-based approaches [34, 35].

Benchmarking iMCDM procedures remains challenging because of several factors: the central involvement of human DMs [26], the diversity in how preference information can be expressed [4], and the dependence on the underlying optimization formulation such as ASF variants [28, 29]. Earlier studies [33] proposed simulating DM behavior through a Machine-based DM (MDM) [25], and a virtual decision-maker library was introduced in [7]. More recently, progressively-interactive EMO (PI-EMO) [17] and Learning-to-Rank (LTR) models [23] attempt to infer preference rankings during the optimization process using value functions or neural networks. However, despite these advances, there is still no comprehensive benchmarking framework for systematically comparing iMCDM algorithms. To overcome this limitation, we introduce a

Machine-based Decision Maker (Machine-DM) that employs a learning model to predict the DM's preference information. It generates both objective classifications and the associated bounding parameters without requiring access to any target solution. By substituting the human DM, the Machine-DM enables a more systematic and repeatable framework for benchmarking iMCDM procedures. The key contributions of this paper are outlined as follows:

(1) Introduction to the concept of Machine Learning (ML)-based Decision-maker (Machine-DM),
(2) Development of a Machine-DM by proposing
   (a) a class predictor ML for predicting the objective class,
   (b) a parameter predictor ML for predicting the bounding parameters of objectives, and
   (c) a preference evaluator for comparing two or more preferred PO-solutions,
(3) Development of four MachDM-iMCDM procedures using STEM, GUESS, STOM, and NIMBUS, and
(4) Bench-iMCDM framework for comparison and performance evaluation of iMCDM procedures.

The remainder of the paper is organized as follows. Section 2 provides background in iMCDM procedures. The proposed Machine-DM approach is detailed in Section 3. Section 4 provides the framework for benchmarking iMCDM procedures. Details in Machine-DM based iMCDM procedures are provided in Section 5. Simulation results are presented in Section 6 followed by the conclusions and future work in Section 7.

2 Background on iMCDM Procedures

Consider an n-variable, M-objective constrained multi-objective optimization (MOO) problem of the form

$$\begin{aligned} \text{Minimize} \quad & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \text{subject to} \quad & g_j(\mathbf{x}) \le 0, \quad j = 1, \ldots, J, \\ & h_k(\mathbf{x}) = 0, \quad k = 1, \ldots, K, \\ & x_i^{(L)} \le x_i \le x_i^{(U)}, \quad i = 1, \ldots, n. \end{aligned} \tag{MOO}$$

Here, J and K denote the numbers of inequality and equality constraints, respectively. The set of feasible solutions satisfying all constraints and variable bounds is denoted as $\mathbf{S}$. Solving (MOO) yields a set of N Pareto-optimal (PO) solutions $\mathbf{x}^{*,(o)}$, $o = 1, \ldots, N$. In practice, MCDM procedures select a single preferred solution from this PO set using a parameterized utility function $s(\mathbf{x}, \mathbf{p})$:

$$\mathbf{x}^{\text{pref}} = \underset{o=1,\ldots,N}{\text{argmin}}\, s(\mathbf{x}^{*,(o)}, \mathbf{p}). \tag{1}$$

A common choice for $s(\cdot)$ is the Achievement Scalarizing Function (ASF) [35], defined as

$$s(\mathbf{x}, \mathbf{p}) = \max_{i=1}^{M} \frac{f_i(\mathbf{x}) - \bar{z}_i}{w_i}, \tag{ASF}$$

where $\mathbf{p} = (\bar{\mathbf{z}}, \mathbf{w})$, with $\bar{\mathbf{z}}$ a reference point and $\mathbf{w}$ a weight vector. Instead of generating a PO set and then identifying a preferred solution, several MCDM approaches directly and repeatedly solve a parameterized and scalarized optimization problem based on the current preference information $\mathbb{P}(\mathbf{p})$ starting with the optimal solution of previous parameterized problem:

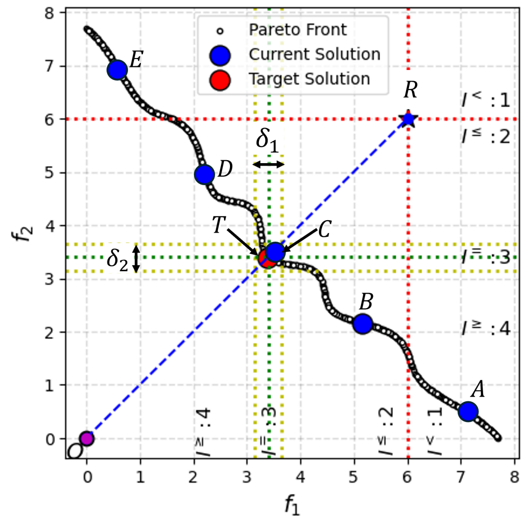


Figure 1: Background on the Machine-DM approach. ***A*-*E* are Pareto-optimal points located in different classes. *R* and *O* indicate a pessimistic point and ideal point, respectively. δ_1 and δ_2 are the width of Class $I^=$ and *T* is the target solution.**

$$\begin{aligned} \text{Minimize} \quad & s(\mathbf{x}, \mathbb{P}), \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{S}. \end{aligned} \tag{2}$$

Preferences may include aspiration levels, reference points or directions, objective classifications $\mathbf{I}$, or other indirect preference forms [36, 37]. ASF-based methods also rely on ideal and nadir point normalization [26]. A variety of iMCDM procedures—such as STEM [1], GUESS [5], STOM [30], NIMBUS [27], and light-beam search [14]— enable stepwise preference articulation. However, a unified framework for benchmarking iMCDM is still lacking. Since these approaches share core components (preference elicitation and ASF formulation), we introduce Machine-DM framework, and demonstrate its integration with STEM, GUESS, STOM, and NIMBUS.

3 Proposed Machine-DM Approach

An overall outline of Machine-DM approach is illustrated in Figure 2. It consists of two artificial neural networks (ANNs) and a preference evaluator that jointly perform objective classification, bounding-parameter prediction, and comparison of preferred PO solutions. Both ANNs take a PO decision vector $\mathbf{x}^{*,c}$ as input.

The first ANN, the class-predictor $\mathcal{F}_{CL}$, predicts the classification vector $\mathbf{I}$ for the M objectives directly from $\mathbf{x}^{*,c}$, without computing its objective values $\mathbf{f}^{*,c}$. Following the structure used in NIMBUS [27], $\mathcal{F}_{CL}$ assigns each objective $f_i^c = f_i(\mathbf{x}^{*,c})$ to one of four classes:

(1) Class $\mathbf{I}^<$: The objective should be improved, i.e., $f_i(\mathbf{x}) < f_i^c$,
(2) Class $\mathbf{I}^\le$: Improvement is desired up to an aspiration level $\bar{z}_i < f_i^c$, which acts as a lower-bound parameter,
(3) Class $\mathbf{I}^=$: The current value is satisfactory, i.e. $f_i(\mathbf{x}) \approx f_i^c$; ideally, all objectives should fall into this class, and
(4) Class $\mathbf{I}^\ge$: The objective may be relaxed up to an allowable upper bound $\varepsilon_i > f_i^c$, serving as an upper-bound parameter.

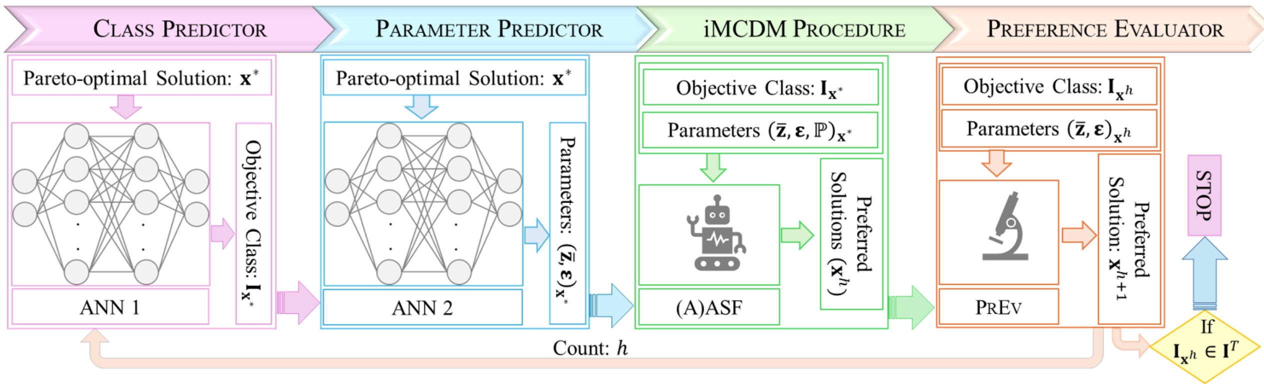


Figure 2: Outline of proposed ML-based Machine DM (Machine-DM) approach

The classification vector $\mathbf{I} \in \{I^<, I^\leq, I^=, I^\geq\}^M$ encodes the *classification status* of a Pareto-optimal (PO) solution. Figure 1 illustrates the objective classification of five solutions (A–E) for a bi-objective problem. The second ANN, referred to as the parameter predictor ML ($\mathcal{F}_{PP}$), predicts the bounding parameters $\bar{z}i$ and ε_i corresponding to Classes $I^\leq$ and $I^\geq$, respectively. Together, these ANNs provide both the classification and the permissible bounds for each objective, effectively emulating the guidance a human decision-maker would offer for the current preferred PO solution $\mathbf{x}^{*,c}$. Additionally, we introduce a preference-evaluator ($\mathcal{M}_{PE}$) to compare and rank solutions using the classification statuses and bounding parameters generated by $\mathcal{F}_{CL}$ and $\mathcal{F}_{PP}$.

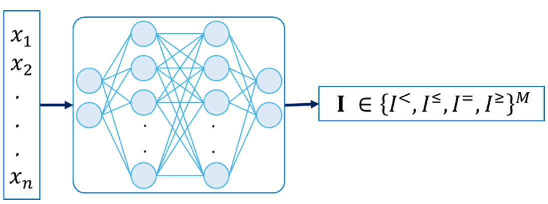


Figure 3: Class predictor ML ($\mathcal{F}_{CL}$): $\mathbf{x}^*(\in \mathbb{R}^n) \longrightarrow \mathbf{I}(\in \mathbb{Z}^M)$. $\mathbb{Z} = \{1, 2, 3, 4\}$ represents four Classes $\{I^<, I^\leq, I^=, I^\geq\}$.

3.1 Class Predictor ML

Ideally, the class-predictor illustrated in Figure 3 is an ANN that takes a PO decision vector ($\mathbf{x}^*$) as input and outputs the corresponding objective classification status vector:

$$\mathcal{F}_{CL} : \mathbf{x}^* \xrightarrow{\mathcal{F}_{CL}} \{I^<, I^\leq, I^=, I^\geq\}^M, \quad \mathbf{x}^* \in \mathbb{R}^n.$$

In essence, $\mathcal{F}_{CL}$ establishes a direct mapping from a decision vector ($\mathbf{x}$) to its objective classification status ($\mathbf{I}$) without explicitly computing the objective values. A key advantage of $\mathcal{F}_{CL}$ is that it requires no prior knowledge of the pessimistic point (R), the target solution (T), or the actual objective values (refer to Figure 1). Consequently, using this ANN for a given $\mathbf{x}$ eliminates the need for costly objective function evaluations. In this sense, ANN1 serves both as a surrogate model and as a classifier, providing the objective classification.

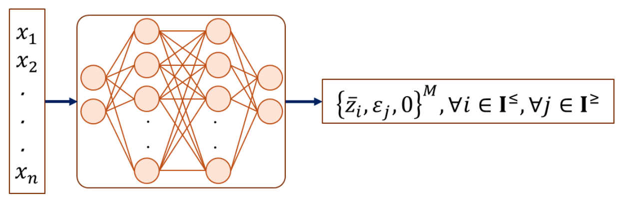


Figure 4: Parameter predictor ML ($\mathcal{F}_{PP}$) : $\mathbf{x} \longrightarrow \{\bar{z}, \varepsilon, 0\}^M$.

3.2 Parameter Predictor ML

The parameter-predictor ANN (Figure 4) estimates the bounding parameters $(\bar{z}i, \varepsilon_j; i \in I^\leq, j \in I^\geq)$, which define the lower and upper bounds of objective values for the subsequent iteration of the iMCDM procedure. Traditionally, these bounds are provided manually by a human decision-maker. The parameter-predictor ANN ($\mathcal{F}_{PP}$) automates this process by directly inferring the bounding parameters from a given PO decision vector $\mathbf{x}^*$:

$$\mathcal{F}_{PP} : \mathbf{x}^* \xrightarrow{\mathcal{F}_{PP}} \{\bar{z}, \varepsilon, 0\}^M.$$

Here, ε_j corresponds to each Class $I^\geq$ objective, $\bar{z}i$ corresponds to each Class $I^\leq$ objective, and zero is assigned for objectives in Classes $I^<$ or $I^=$. Specifically, ε_j represents the upper bound for the j-th Class $I^\geq$ objective (as identified by $\mathcal{F}_{CL}$), ensuring $f_j(\mathbf{x}) \leq \varepsilon_j$, while $\bar{z}_i$ is the lower bound for the i-th Class $I^\leq$ objective, requiring $f_i(\mathbf{x}) \geq \bar{z}i$. For objectives in Classes $I^<$ and $I^=$, no bounds are necessary, so $\mathcal{F}_{PP}$ outputs zero.

For a Class $I^\geq$ objective, the upper bound ε_j is determined based on the current objective value $f_j^c = f_j(\mathbf{x}^{*,c})$ and the perceived target f_j^T. When f_j^c is far from f_j^T, the decision-maker may be uncertain about the desired improvement. This uncertainty is addressed by computing Δ_j as shown in Figure 6 and defined in Equation (3):

$$\Delta_j = \min\left[\alpha_j\left(z_j^{\text{nad}} - z_j^*\right), \left(f_j^T - f_j^c\right) + 0.5\delta_j\right], \tag{3}$$

where α_j is a problem-specific parameter, and δ_j defines the Class $I^=$ range, as illustrated in Figure 6. The upper bound ε_j is then set with a small uniform randomness ($\mathbb{U}$) to reflect human uncertainty:

$$\varepsilon_j = f_j^c + (1 + \alpha_j \mathbb{U}[-1,\ 1])\Delta_j, \tag{4}$$

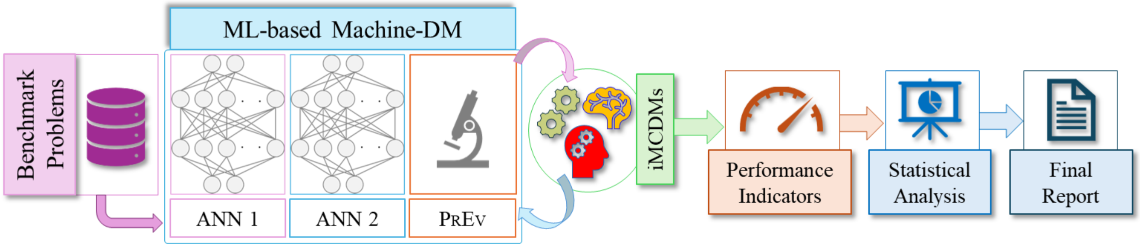


Figure 5: Outline of proposed framework for benchmarking iMCDM procedures

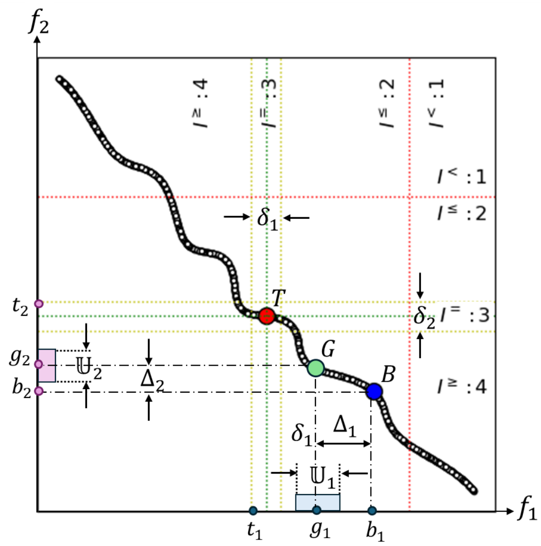


Figure 6: Defining the bounding parameters $(\bar{z}_i, \varepsilon_j)$ for parameter-predictor ML.

ensuring that $\varepsilon_j > f_j^c$ when $\alpha_j < 1$. Similarly, for Class $I^{\leq}$ objectives, the lower bound $\bar{z}_i$ is computed as:

$$\bar{z}_i = f_i^c - (1 + \alpha_i \mathbb{U}[-1,\ 1])\Delta_i, \tag{5}$$

with Δ_i defined analogously to Equation (3) using index i. This ensures $\bar{z}_i < f_i^c$ for $\alpha_i < 1$. The inclusion of $\mathbb{U}$ introduces variability in the bounds, mimicking the inexact responses of a human DM.

3.3 Preference Evaluator

When all objectives belong to Class $I^{=}$ (or $I^{=} \cup I^{\geq}$), the solution is considered preferred, and the iMCDM procedure can terminate. The preference-evaluator metric ($\mathcal{M}_{PE}$) is defined as:

$$\mathcal{M}_{PE} = \max_{m=1}^{M} |3 - I_m| + \{\max\left(\bar{z}_i^n, \varepsilon_j^n\right),\ \forall i \in \mathbf{I}^{\leq}, \forall j \in \mathbf{I}^{\geq}\}, \tag{6}$$

where $I_m \in 1, 2, 3, 4$ corresponds to Classes $I^{<}, I^{\leq}, I^{=}, I^{\geq}$, and

$$\bar{z}_i^n = \frac{\bar{z}_i - f_i^T}{f_i^R - f_i^T}, \quad \varepsilon_j^n = \frac{f_j^T - \varepsilon_j}{f_j^T - f_j^*}. \tag{7}$$

The first term quantifies the deviation from the ideal class, while the second captures the relative proximity for objectives in Classes $I^{\leq}$ and $I^{\geq}$. Smaller values of $\mathcal{M}_{PE}$ indicate solutions closer to the target, facilitating the ranking and comparison of PO solutions.

4 Proposed Bench-iMCDM Framework

This section presents a benchmarking framework for iMCDM procedures, along with preliminary guidelines based on the proposed ML-based Machine-DM approach illustrated in Figure 5, which is currently under development for general use.

4.1 Outline of Bench-iMCDM framework

An outline of Bench-iMCDM framework is provided in Figure 5. Having the proposed Machine-DM at its core, Bench-iMCDM framework consists of the following four key components:

(1) **Benchmark problems:** Bench-iMCDM requires a diverse suite of benchmark problems of varying complexity, along with additional descriptors outlined in Section 4.2.
(2) **ML-based Machine-DM and Preference Evaluators:** For each problem and decision-making instance identified in item (1), a corresponding Machine-DM and other DM-related steps in iMCDM procedures will be provided.
(3) **Performance metrics:** A set of performance indicators for assessing and comparing MachDM-iMCDM procedures, with candidate metrics described in Section 4.3.
(4) **Competing MachDM-iMCDM procedures and results:** A collection of existing MachDM-iMCDM procedures will be supplied as reference benchmarks for evaluating user-developed methods on the problems in item (1).

4.2 Information on benchmark problems

We clearly outline the information that will be provided to an user for developing or executing their iMCDM procedures:

(1) The complete formulation of the optimization problem (objectives, constraints, and variable bounds as specified in Equation MOO) will be provided. The ideal and nadir points of the PF will also be supplied for normalization.
(2) For each benchmark problem and each decision-making instance corresponding to a single preferred PO target solution, two pre-trained ANNs ($\mathcal{F}_{CL}$ and $\mathcal{F}_{PP}$) will be made available. Importantly, information on the target solution or reference point will be disclosed, as well.
(3) A common initial point ($\boldsymbol{x}^{(0)}$) will be provided to ensure all competing iMCDM procedures start from an identical start point, enabling fair comparisons.
(4) One or more standard performance-evaluation metrics will be supplied so that all users can employ consistent criteria to compare their iMCDM procedures.

Users are expected to develop their own iMCDM procedures, apply them to identify the target solution using the above information, and report the performance evaluation. The effectiveness of a proposed iMCDM approach can be assessed by aggregating its performance multiple problems across a test suite. Since no human DM is involved in this benchmarking framework, it is anticipated that EMO/MCDM researchers will be encouraged to design and evaluate new, innovative iMCDM procedures.

4.3 Performance evaluation metrics

When implementing the Bench-iMCDM framework with a specific iMCDM algorithm on a given problem, a single run is inadequate for reliable evaluation and benchmarking. Therefore, statistical testing is performed by varying the start point or target solution across multiple executions of MachDM-iMCDM. Let E denote the number of independent runs conducted for a given decision-making instance by changing the start point in iMCDM procedures.

4.3.1 Number of DM Calls: Consider that the number of iterations in MachDM-iMCDM (DM calls) to get a converged solution close to target point is $\mathrm{h_e}$ during the experiment e. If the total number of experiments conducted is E, the metric number of DM calls (h_E) can be formulated as follows:

$$h_E = \text{median}\,\{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_E\} \tag{8}$$

4.3.2 Success Rate: Success Rate ($\eta_E \in [0, 1]$) is the proportion of the experiments/runs in which the target classification is achieved using MachDM-iMCDM procedure and can be expressed as:

$$\eta_E = \underset{e=1,\ldots,E}{\text{median}} \left[1\!\left(\mathbf{I}_e^T \in \mathbf{I}^{=} \cup \mathbf{I}^{\geq} \right) \right]. \tag{9}$$

4.3.3 Deviation from Target Class: In each independent run e, the deviation from the target class (**I**) is computed by taking the absolute difference between the each objective class and target class and normalized it between 0 and 1. The objective class deviation (ϕ_e) from the target class in a given experiment e is computed as:

$$\phi_e = \frac{1}{2M} \sum_{m=1}^{M} \left| 3 - I_m^{T,e} \right|. \tag{10}$$

Finally, the deviation metric across experiment ($\Phi_E \in [0, 1]$) is computed as:

$$\Phi_E = \text{median}\,\{\phi_1, \phi_2, \ldots, \phi_E\}. \tag{11}$$

4.3.4 Number of Solution Evaluations: Number of function evaluation across E experiments is computed, as follows:

$$N_{evl}^E = Gen \times pop \times (h_E + 1), \tag{12}$$

where Gen and pop are the number of generations and population size of R-GA. Including the computation of feasible point in Subproblem (SP-0), total (h_e + 1) times solution **x** is evaluated.

5 Machine-DM based iMCDM Procedure

This section provides details on STEM, GUESS, STOM and NIMBUS procedure following Machine-DM approach.

5.1 MachDM-STEM procedure

The steps involved in MachDM-STEM are provided in the following:

(1) Compute a feasible start point $\mathbf{x}^{(0)}$ and corresponding objective vector $\mathbf{z}^{(0)}$. Calculate ideal objective vector ($\mathbf{z}^*$) and nadir objective vector ($\mathbf{z}^{nad}$). Set $h = 1$. Solve the following AASF subproblem to compute start point ($\mathbf{z}^{(1)}$):

$$\begin{aligned} &\text{Minimize} \quad \max_{i=1}^{M} \left[\frac{f_i(\mathbf{x}) - z_i^{(0)}}{z_i^{nad} - z_i^{**}} + \rho \sum_{i=1}^{M} \frac{f_i(\mathbf{x})}{z_i^{nad} - z_i^{**}} \right] \\ &\text{subject to} \quad \mathbf{x} \in \mathrm{S}. \end{aligned} \tag{SP-0}$$

(2) Denote the solution by $\mathbf{x}^h \in S$ and the corresponding objective vector by $\mathbf{z}^h \in Z$, where S and Z are decision space and criterion space, respectively.
(3) Use ANN-1 to classify the objective functions at $\mathbf{x}^h$ into satisfactory $\{I^{\geq}, I^{=}\}$ and unsatisfactory ones $\{I^{<}, I^{\leq}\}$. If all the objective functions belong to class $\{I^{=}, I^{\geq}\}$, go to step (6). Otherwise, use ANN-2 to specify relaxed upper bounds $\varepsilon_i (i \in I^{\geq})$ and lower bound $\bar{z}_i (i \in I^{\leq})$ for the satisfactory objective functions.
(4) Compute weights $w_i; i = 1, \ldots, M$; as provided in Section 5.1.1. Set $w_i = 0$, for $i \in I^{\geq} \cup I^{=}$. Solve the following subproblem, where the upper bounds are taken into account:

$$\begin{aligned} &\text{Minimize} \quad \max_{i=1}^{M} \left[w_i | f_i(\mathbf{x}) - z_i^* | \right] \\ &\text{subject to} \quad f_i(\mathbf{x}) \leq \varepsilon_i, \quad \forall\, i \in \mathbf{I}^{\geq}, \\ &\qquad\qquad\quad f_j(\mathbf{x}) \leq f_j(\mathbf{x}^h), \quad \forall j \in \mathbf{I}^{<} \cup \mathbf{I}^{\leq} \cup \mathbf{I}^{=}, \\ &\qquad\qquad\quad \mathbf{x} \in \mathrm{S}. \end{aligned} \tag{SP-I}$$

(5) Compute preference-evaluator metric ($\mathcal{M}_{PE}$) for solution $\mathbf{x}^{h+1}$, and compute corresponding objective vector $\mathbf{z}^{h+1}$. *If* $\mathbf{x}^h = \mathbf{x}^{h+1}$, *go to step (6)*. Otherwise, set $h = h + 1$, and go to step (2).
(6) Stop. The final solution is $\mathbf{x}^h$.

5.1.1 Computation of weight vector for MachDM-STEM in Step (4). The weight w_i can be computed as follows:

$$\begin{aligned} &w_i = \frac{e_i}{\sum_{j=1}^{M} e_j}, \quad i = 1, \ldots, M; \\ &\text{where}, \forall i \quad e_i = \frac{1}{z_i^*} \frac{z_i^{\text{nad}} - z_i^*}{z_i^{\text{nad}}}, \quad z_i^*, z_i^{\text{nad}} \neq 0. \end{aligned} \tag{\mathbf{w}\text{-vec.}}$$

5.2 MachDM-STOM procedure

The MachDM-STOM procedure has the following steps:

(1) Calculate the utopian objective vector $\mathbf{z}^{**}$. Set $h = 0$.
(2) Compute a feasible start point $\mathbf{x}^{(0)}$ and corresponding objective vector $\mathbf{z}^{(0)}$. Solve the AASF subproblem (SP-I) to compute start point ($\mathbf{z}^{(1)}$). Set $h = 1$.
(3) Use the Class Predictor (ANN1: $\mathcal{F}_{CL}$) to classify the objective vector $\mathbf{z}^h$ into the classes $\{I^{<}, I^{\leq}, I^{=}, I^{\geq}\}$. If all the objectives belongs to class $\{I^{=}, I^{\geq}\}$, go to step (6).
(4) Use Parameter Predictor (ANN 2: $\mathcal{F}_{PP}$) to compute new aspiration levels $\hat{z}_i^h = \bar{z}_i^h; \forall i \in I^{\leq}$. Compute a new aspiration level $\hat{z}_i^h; \forall i \in I^{<}$ by selecting a random point between z_i^* and z_i^h. Set $\hat{z}_i^h = z_i^h \quad \forall i \in I^{=}$. Use automatic trade-off to obtain $\hat{z}_i^h; \forall i \in I^{\geq}$, if available [30]. Otherwise, use Parameter Predictor ANN to compute the term $\hat{z}_i^h = \varepsilon_i^h$.

(5) Using the reference point $\hat{\mathbf{z}}^h$, minimize the following scalarizing function:

$$\text{Minimize} \quad \max_{i=1}^{M}\left[\frac{f_i(\mathbf{x}) - z_i^{**}}{\hat{z}_i^h - z_i^{**}}\right] + \rho \sum_{i=1}^{M} \frac{f_i(\mathbf{x})}{\hat{z}_i^h - z_i^{**}}, \qquad \text{(SP-II)}$$
$$\text{subject to} \quad \mathbf{x} \in \mathrm{S}.$$

to compute a PO-solution $\mathbf{x}^{h+1}$. Let the corresponding objective vector be $\mathbf{z}^{h+1}$. If $\mathbf{x}^{h+1} = \mathbf{x}^h$, for $h > 1$, go to Step (6). Otherwise, set $h = h + 1$ and go to step (3).

(6) Stop with the final solution $\mathbf{x}^h$.

5.3 MachDM-GUESS procedure

The MachDM-GUESS procedure has the following steps:

(1) Calculate ideal objective vector ($\mathbf{z}^*$) and nadir objective vector ($\mathbf{z}^{nad}$). Set $h = 0$.

(2) Compute a feasible start point $\mathbf{x}^{(0)}$ and corresponding objective vector $\mathbf{z}^{(0)}$. Set $h = 1$. Solve the AASF subproblem (SP-I) to compute start point ($\mathbf{z}^{(1)}$).

(3) Predict the objective classification $\left(\mathbf{I} \in \{I^<, I^\leq, I^=, I^\geq\}\right)$ and parameters ($\varepsilon_i^h, \forall i \in I^\leq; \bar{z}_j^h, \forall j \in I^\geq$) using Class Predictor (ANN1: $\mathcal{F}_{CL}$) and Parameter Predictor (ANN2: $\mathcal{F}_{PP}$), respectively. If all the objective functions belong to class $\{I^=, I^\geq\}$, go to Step (7).

(4) Use the predicted parameters ($\varepsilon_i^h, \forall i \in I^\leq; \bar{z}_j^h, \forall j \in I^\geq$) to specify upper or lower bounds to the objective functions and solve the following AASF problem by computing a reference point $\hat{\mathbf{z}}^h$ as provided in Subsection 5.3.1.

$$\text{Minimize} \quad \max_{i=1}^{M}\left[\frac{f_i(\mathbf{x}) - z_i^{\text{nad}}}{z_i^{\text{nad}} - \hat{z}_i^h}\right] + \rho \sum_{i=1}^{M} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - \hat{z}_i^h}, \qquad \text{(SP-III)}$$
$$\text{subject to} \quad \mathbf{x} \in \mathrm{S}.$$

(5) Use Preference Evaluator to compute $\mathcal{M}_{PE}$ values for current solution ($\mathbf{x}^h$) and compare it with previous solution's ($\mathbf{x}^1, \ldots, \mathbf{x}^{h-1}$) $\mathcal{M}_{PE}$ values to rank the solutions according to their ascending order of $\mathcal{M}_{PE}$ values.

(6) For current solution ($\mathbf{x}^h$), *if* $\mathbf{x}^h = \mathbf{x}^{h+1}$*, go to step (7).* Otherwise, set $h = h + 1$ and go to step (3).

(7) Stop. The final solution is $\mathbf{x}^h$.

5.3.1 Computation of reference point for MachDM-GUESS in Step (4). Consider that $\bar{z}_i^h$ and ε_i^h are the bounding parameters of solution $\mathbf{z}^h$, in an iteration h. The reference point ($\hat{\mathbf{z}}^h$) in Step (4) of MachDM-GUESS procedure is computed as follows:

(a) for objectives (f_i) in class $I^<$, $\hat{z}_i^h = \mathbb{U}[z_i^*, z_i^h]$,
(b) for $f_i, \forall i \in I^\leq, \hat{z}_i^h = \mathbb{U}[z_i^*, \bar{z}_i^h]$,
(c) for $f_i, \forall i \in I^=, \hat{z}_i^h = z_i^h$, and
(d) finally, for objectives f_i in class $I^\geq$, $\hat{z}_i^h = \mathbb{U}[z_i^h, \varepsilon_i]$.

5.4 MachDM-NIMBUS procedure

The detailed description on MachDM-NIMBUS procedure is provided in Appendix A3.

6 Simulation Results

In the previous section, we have presented a four Mach-iMCDM approaches, in which every human decision-making event is replaced with the proposed Machine-DM models. It uses two neural networks – ANN1 and ANN2 – along with a preference-evaluator to execute decision-making steps as closely as possible to the original iMCDM procedures. We demonstrate the comparison of four MachDM-iMCDM procedures on four test and four real-world constrained and unconstrained multi- and many-objective optimization problems. For each problem, four decision-making instance is chosen based on the start point. The description on four start points for each example is provided in Appendix A.4. The description on Machine-DM parameters α_j and δ_j presented in Equations 3 and 4 are expressed as $\alpha_j = \hat{\alpha}$ and $\delta_j = \hat{\delta} \times (z_j^{\text{nad}} - z_j^*)$ for all j. Description on problems and respective parameters $\hat{\alpha}$ and $\hat{\delta}$ are provided in Table 1.

Table 1: Problem description and parameter setting for GA and Machine-DM. Parameters M, n, and J are number of objectives, variables, and constraints, respectively.

Examples	Problem Details			NSGA-II/III Parameters		Machine-DM Parameters		ANN1/ANN2 Parameters		
	M	n	J	pop	Gen	$\hat{\delta}$	$\hat{\alpha}$	lr_1	lr_2	l_{reg}
Knee	2	5	0	1000	100	0.100	0.075	0.0100	0.010	0.010
Knee	3	5	0	1350	100	0.050	0.100	0.0010	0.001	0.010
Crash.	3	5	0	1350	100	0.100	0.050	0.0010	0.002	0.020
Car Side	3	7	10	1350	100	0.100	0.075	0.0020	0.001	0.012
River	4	2	0	2000	200	0.015	0.025	0.0025	0.001	0.010
WATER	5	3	7	2000	200	0.025	0.035	0.0010	0.002	0.020
NIMBUS	6	2	0	2000	200	0.030	0.050	0.0020	0.001	0.010
DTLZ2	8	10	0	2000	500	0.050	0.200	0.0010	0.001	0.010

To train ANN1 and ANN2, a representative PO solution set is first generated using NSGA-II or NSGA-III in *pymoo* [2]. The augmentation parameter (ρ) in AASF formulation presented in Subproblems is set to 0.001. The Subproblems (SP-0), (SP-I), (SP-II), and (SP-III) are solved using real-coded GA (RGA) [12]. The RGA parameters – population size and number of generations – are set to 100 and 200, respectively, for each problem. Other RGA parameters are kept as their standard values [16]. Maximum DM call is set as $h_e^{max} = 11$.

ANN1 and ANN2 with hidden layers [128, 64, 32, 16] is trained using the Adam optimizer [20] with learning rates lr_1 and lr_2, respectively. For ANN2, L2-regularization with parameter l_{reg} is used. For both the ANNs, train-test split is kept as 80% and 20%. The batch-size and maximum epoch for both the ANNs are set to 32 and 500, respectively. For training ANN1, error metrics – Accuracy, Precision, Recall, and F-1 Score – are checked for their closeness to one. For training ANN2, R^2 and MSE is used as error metric. Details of the error metrics for training ANNs are presented in Appendix A.2.

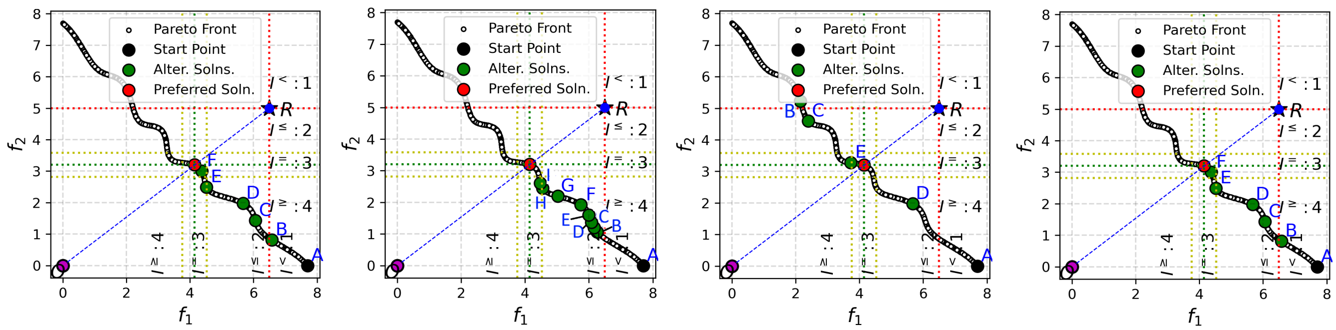
6.1 Bi-objective knee example

Starting with a feasible start point $\mathbf{x}^{(0)}$ = [1.50, 0.01, 0.01, 0.01, 0.01], the implementation of MachDM-iMCDM procedures on bi-objective knee example [3] is provided in Figures 7a, 7b, 7c, and 7d, respectively. The iteration-wise solutions for MachDM-STEM, MachDM-STOM, MachDM-GUESS, and MachDM-NIMBUS procedures is provided in Appendix A.3. It can be observed from figure and table that for this decision-making instance the four MachDM-iMCDM procedures obtained six, eleven, five, and seven number of solution (z), respectively. For statistical comparison, four different

Table 2: Statistical comparison of MachDM-iMCDM procedures demonstrated on benchmark and real-world examples. Values of the performance metrics for four independent runs along with median values are provided for each MachDM-iMCDM procedure and for each examples, using four different start points. Information on start points is provided in Appendix A.4. Metrics h_E, η_E, Φ_E, and N_{evl} denotes the number of DM calls, success rate, and deviation from the target class, respectively.

Example	MachDM-STEM				MachDM-STOM				MachDM-GUESS				MachDM-NIMBUS			
	h_E	η_E	Φ_E	N^E_{evl}	h_E	η_E	Φ_E	N^E_{evl}	h_E	η_E	Φ_E	N^E_{evl}	h_E	η_E	Φ_E	N^E_{evl}
Knee example (2-objective)	5.0	1.0	0.250	1.20E+05	5.0	1.0	0.250	1.20E+05	2.0	1.0	0.250	6.00E+04	6.0	1.0	0.250	1.40E+05
	6.0	1.0	0.000	1.40E+05	9.0	1.0	0.250	2.00E+05	5.0	1.0	0.250	1.20E+05	6.0	1.0	0.000	1.40E+05
	6.0	1.0	0.000	1.40E+05	9.0	1.0	0.250	2.00E+06	3.0	1.0	0.250	8.00E+04	7.0	1.0	0.000	1.60E+05
	6.0	1.0	0.250	1.40E+05	5.0	1.0	0.250	1.20E+05	4.0	1.0	0.250	1.00E+05	6.0	1.0	0.250	1.40E+05
Median	6.0	**1.0**	**0.125**	1.40E+05	7.0	**1.0**	0.250	1.60E+05	**3.5**	**1.0**	0.250	**9.00E+04**	6.0	**1.0**	**0.125**	1.40E+05
Knee example (3-objective)	9.0	0.0	0.333	2.00E+05	8.0	1.0	0.000	1.80E+05	6.0	1.0	0.167	1.40E+05	5.0	0.0	0.500	1.20E+05
	6.0	1.0	0.167	1.40E+05	6.0	1.0	0.333	1.40E+05	3.0	1.0	0.333	8.00E+04	6.0	1.0	0.167	1.40E+05
	6.0	0.0	0.333	1.40E+05	5.0	1.0	0.000	1.20E+05	6.0	1.0	0.167	1.40E+05	6.0	1.0	0.167	1.40E+05
	6.0	1.0	0.167	1.40E+05	4.0	1.0	0.167	8.20E+04	4.0	1.0	0.167	8.20E+04	6.0	1.0	0.167	1.40E+05
Median	6.0	0.5	0.250	1.40E+05	5.5	**1.0**	**0.083**	1.30E+05	**5.0**	**1.0**	0.167	**1.20E+05**	6.0	**1.0**	0.167	1.40E+05
Vehicle crash-worthiness	5.0	1.0	0.000	1.20E+05	3.0	1.0	0.000	8.00E+04	2.0	1.0	0.000	6.00E+04	6.0	0.0	0.167	1.40E+05
	5.0	1.0	0.000	1.20E+05	8.0	1.0	0.000	1.80E+05	2.0	1.0	0.000	6.00E+04	6.0	0.0	0.167	1.40E+05
	6.0	0.0	0.167	1.40E+05	5.0	1.0	0.000	1.20E+05	2.0	1.0	0.000	6.00E+04	6.0	0.0	0.167	1.40E+05
	5.0	0.0	0.333	1.20E+05	2.0	1.0	0.000	6.00E+04	2.0	1.0	0.000	6.00E+04	5.0	1.0	0.000	1.20E+05
Median	5.0	0.5	0.083	1.20E+05	4.0	**1.0**	**0.000**	1.00E+05	**2.0**	**1.0**	**0.000**	**6.00E+04**	6.0	0.0	0.167	1.40E+05
Car side impact	8.0	1.0	0.000	1.80E+05	10.0	1.0	0.000	2.20E+05	3.0	1.0	0.000	6.20E+04	7.0	1.0	0.000	1.60E+05
	8.0	1.0	0.000	1.80E+05	8.0	1.0	0.167	1.80E+05	5.0	1.0	0.000	1.20E+05	9.0	0.0	0.167	2.00E+06
	7.0	0.0	0.167	1.60E+05	11.0	0.0	0.333	2.40E+05	4.0	1.0	0.000	1.00E+05	6.0	1.0	0.000	1.40E+05
	8.0	0.0	0.167	1.80E+05	4.0	1.0	0.000	1.00E+05	3.0	1.0	0.167	8.00E+04	8.0	1.0	0.000	1.80E+05
Median	8.0	0.5	0.083	1.80E+05	9.0	**1.0**	0.083	2.00E+05	**3.5**	**1.0**	**0.000**	**9.00E+04**	7.5	**1.0**	**0.000**	1.70E+05
River pollution	9.0	0.0	0.250	2.00E+05	11.0	0.0	0.500	2.40E+05	7.0	1.0	0.000	1.60E+05	9.0	0.0	0.250	2.00E+05
	4.0	0.0	0.250	1.00E+05	11.0	0.0	0.500	2.40E+05	4.0	1.0	0.125	1.00E+05	4.0	0.0	0.250	1.00E+05
	8.0	0.0	0.125	1.80E+05	11.0	0.0	0.500	2.40E+05	7.0	1.0	0.000	1.60E+05	9.0	0.0	0.250	2.00E+05
	3.0	0.0	0.250	8.00E+04	11.0	0.0	0.500	2.40E+05	5.0	1.0	0.000	1.20E+05	3.0	0.0	0.250	8.00E+04
Median	**6.0**	0.0	0.250	1.40E+05	11.0	0.0	0.500	2.40E+05	**6.0**	**1.0**	**0.000**	**1.40E+05**	6.5	0.0	0.250	1.50E+05
WATER example	3.0	0.0	0.400	8.00E+04	11.0	0.0	0.200	2.40E+05	6.0	1.0	0.000	1.40E+05	3.0	0.0	0.200	8.00E+04
	5.0	0.0	0.400	1.20E+05	11.0	0.0	0.400	2.40E+05	4.0	1.0	0.000	1.00E+05	5.0	0.0	0.400	1.20E+05
	3.0	0.0	0.200	8.00E+04	2.0	1.0	0.000	6.00E+04	3.0	1.0	0.000	8.00E+04	2.0	1.0	0.000	6.00E+04
	6.0	1.0	0.000	1.40E+05	11.0	0.0	0.200	2.40E+05	5.0	1.0	0.000	1.20E+05	5.0	0.0	0.200	1.20E+05
Median	**4.0**	0.0	0.300	**1.00E+05**	11.0	0.0	0.200	2.40E+05	4.5	**1.0**	**0.000**	1.10E+05	**4.0**	0.0	0.200	**1.00E+05**
NIMBUS benchmark example	7.0	0.0	0.333	1.60E+05	11.0	0.0	0.250	2.40E+05	11.0	0.0	0.333	2.40E+05	8.0	0.0	0.250	1.80E+05
	7.0	0.0	0.417	1.60E+05	11.0	0.0	0.250	2.40E+05	11.0	0.0	0.417	2.40E+05	10.0	0.0	0.417	2.20E+05
	5.0	0.0	0.417	1.20E+05	11.0	0.0	0.167	2.40E+05	11.0	0.0	0.333	2.40E+05	5.0	0.0	0.333	1.20E+05
	7.0	0.0	0.500	1.60E+05	11.0	0.0	0.250	2.40E+05	11.0	0.0	0.167	2.40E+05	9.0	0.0	0.417	2.00E+06
Median	**7.0**	**0.0**	0.417	**1.60E+05**	11.0	**0.0**	**0.250**	2.40E+05	11.0	**0.0**	0.333	2.40E+05	8.5	**0.0**	0.375	1.90E+05
DTLZ2 (8-objective)	11.0	0.0	0.313	2.40E+05	11.0	0.0	0.500	2.40E+05	10.0	1.0	0.063	2.20E+05	11.0	0.0	0.563	2.40E+05
	11.0	0.0	0.438	2.40E+05	11.0	0.0	0.563	2.40E+05	11.0	0.0	0.563	2.40E+05	11.0	0.0	0.563	2.40E+05
	10.0	1.0	0.125	2.20E+05	11.0	0.0	0.500	2.40E+05	7.0	1.0	0.125	1.60E+05	11.0	0.0	0.500	2.40E+05
	11.0	0.0	0.375	2.40E+05	11.0	0.0	0.563	2.40E+05	11.0	0.0	0.500	2.40E+05	11.0	0.0	0.563	2.40E+05
Median	11.0	0.0	0.344	2.40E+05	11.0	0.0	0.531	2.40E+05	**10.5**	**0.5**	**0.313**	**2.30E+05**	11.0	0.0	0.563	2.40E+05

start points $\mathbf{x}^{(0)}$ (provided in Appendix A.4) are used to execute these four MachDM-iMCDM procedures and performance metrics are evaluated. The median values of performance metrics are provided in Table 2. An efficient iMCDM procedure has smaller values of h_E, Φ_E, and N^E_{evl}. The bold text are used to highlight the best metric for a given problem. For two-objective knee example, these metrics are found to be the best for the GUESS procedure except for deviation from target class Φ_E, which is found best for NIMBUS



(a) MachDM-STEM solutions (b) MachDM-STOM solutions (c) MachDM-GUESS solutions (d) MachDM-NIMBUS solutions

Figure 7: MachDM-iMCDM solutions obtained for two-objective knee example with start point $\mathbf{x}^{(0)}$=[1.50, 0.01, 0.01, 0.01, 0.01].

procedure. All the procedures has success rate η_E=1, which indicates that all the four procedures are able to achieve the target class $\mathbf{I} \in I^{=} \cup I^{\geq}$.

6.2 Three-objective examples

Three-objective examples include three-objective knee, vehicle crashworthiness, and car side impact problem. The median values of performance metrics for each MachDM-iMCDM procedures executed for four different start point ($\mathbf{x}^{(0)}$) are provided in Table 2. For three-objective knee example, GUESS procedure has least number of DM calls (h_E) and solution evaluations (N^E_{elv}). Except for STEM, each procedure has success rate of 1. However, the deviation from target class (Φ_E) is best for STOM. For vehicle crashworthiness and car side impact problems, GUESS procedure has better or equally good values of performance metrics.

6.3 Many-objective examples

MaOO problems include river pollution, WATER, NIMBUS benchmark, and DTLZ2 examples. For river pollution example, GUESS procedure has better or equally good values of performance metrics. For WATER example, STEM and NIMBUS have least values for h_E and N^E_{evl}. But their success rate (η_E) is zero and have significant deviation (Φ^E) from the target class. However, for GUESS procedure, η_E=1.0 and Φ^E=0. In this case, STOM is the worst performing procedure. For NIMBUS benchmark example, all the procedures have η_E=0 and $\Phi^E \neq$0, which indicates none of the procedures are able to achieve the target class. However, GUESS procedure has the minimum deviation from the target class (Φ_E=0.333). For DTLZ2 example, GUESS procedure is the best performing procedure with lower h_E, Φ_E, N^E_{evl} values and higher Φ^E value.

The simulation results conducted following Bench-iMCDM framework suggests that GUESS procedure has the best performance. Interestingly, several past studies [9, 10] conducted involving human DMs also suggested that GUESS procedure is the most efficient and easy-to-implement procedure. The interesting aspect of our proposed Bench-iMCDM framework is that we obtained the same finding using our proposed Machine-DM approach, without involving any human DMs.

6.4 Availability of Bench-iMCDM

To promote transparency, reproducibility [24], and future research in iMCDM, the Bench-iMCDM framework—comprising benchmark problem formulations, ANN1 and ANN2 models, and the preference-evaluator metric—will be released on GitHub. The repository will function as a general-purpose resource for researchers and practitioners to perform decision-making tasks for comparative analysis and methodological development within the EMO–MCDM domain.

6.5 Challenges and limitations

In several cases, premature convergence occurs because the MachDM-iMCDM procedure terminates when consecutive iterations produce identical solutions, typically when an objective is classified as $I^{=}$. Under this classification, the constraints in Subproblems (SP-III) make it over-constraint problem, restricting progress toward the target classes $\mathbf{I}^T \in I^{=} \cup I^{\geq}$ for the remaining objectives.

7 Conclusions and Future Work

The benchmarking of existing iMCDM procedures is a challenging task due to involvement of a human decision-maker, variation in preference incorporation, and the AASF formulation typically employed in them. In this paper, we have proposed a bench-iMCDM framework using a Machine-DM which uses two ANNs for predicting objective classification and bounding parameters, and a preference evaluator without involving any human DM. The generalization of the proposed Machine-DM has allowed us to integrate the Machine-DM approach with four iMCDM procedures: STEM, STOM, GUESS, and NIMBUS. Based on the objective classification and number of DM calls, four performance metrics have been proposed to compare Machine-DM based iMCDM procedures. The implementation of Machine-DM and Bench-iMCDM on four benchmark and four real-world examples suggests that GUESS is the best-performing iMCDM procedure among four iMCDMs. Further description of the Bench-iMCDM framework and the proposed Machine-DM procedure, followed with more results, are presented in the Appendix A.4. This study should motivate EMO researchers to propose novel and more efficient iMCDM procedures.

The success-rate metric (Φ^E) in Table 2 indicates that, in a few cases, the proposed Machine-DM approach terminates prematurely without reaching the target class. Future work will address this limitation by introducing a stochastic objective classifier $\mathcal{F}_{CL}$. In addition, we plan to improve the parameter predictor to handle cases in which the STEM and NIMBUS subproblem constraints (SP-I) leads to infeasible search, leading to premature convergence.

References

[1] R Benayoun, J De Montgolfier, Jo Tergny, and O Laritchev. 1971. Linear programming with multiple objective functions: Step method (STEM). *Mathematical programming* 1, 1 (1971), 366–375.

[2] Julian Blank and Kalyanmoy Deb. 2020. Pymoo: Multi-objective optimization in python. *IEEE Access* 8 (2020), 89497–89509.

[3] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. 2004. Finding Knees in Multi-objective Optimization. In *Parallel Problem Solving from Nature - PPSN VIII*. Springer Berlin Heidelberg, Berlin, Heidelberg, 722–731.

[4] Jürgen Branke, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz. 2009. Interactive Evolutionary Multiobjective Optimization Using Robust Ordinal Regression. In *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 554–568.

[5] John Telfer Buchanan. 1997. A naive approach for solving MCDM problems: The GUESS method. *Journal of the Operational Research Society* 48, 2 (1997), 202–206.

[6] Vira Chankong and Yacov Y Haimes. 2008. *Multiobjective decision making: Theory and methodology*. Courier Dover Publications.

[7] Lu Chen, Bin Xin, Jie Chen, and Juan Li. 2017. A virtual-decision-maker library considering personalities and dynamically changing preference structures for interactive multiobjective optimization. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. Spain, 636–641. https://doi.org/10.1109/CEC.2017.7969370

[8] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, NY. https://doi.org/10.1007/978-0-387-36797-2

[9] James L Corner and John T Buchanan. 1995. Experimental consideration of preference in decision making under certainty. *Journal of Multi-Criteria Decision Analysis* 4, 2 (1995), 107–121.

[10] James L Corner and John T Buchanan. 1997. Capturing decision maker preference: Experimental comparison of decision analysis and MCDM techniques. *European Journal of Operational Research* 98, 1 (1997), 85–97.

[11] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK.

[12] Kalyanmoy Deb. 2005. A population-based algorithm-generator for real-parameter optimization. *Soft Computing* 9, 4 (2005), 236–253.

[13] Kalyanmoy Deb and Himanshu Jain. 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE transactions on evolutionary computation* 18, 4 (2013), 577–601.

[14] Kalyanmoy Deb and Abhay Kumar. 2007. Light beam search based multi-objective optimization using evolutionary algorithms. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2125–2132.

[15] Kalyanmoy Deb and Kaisa Miettinen. 2010. Nadir point estimation using evolutionary approaches: better accuracy and computational speed through focused search. In *Multiple criteria decision making for sustainable energy and transportation systems*. Springer, 339–354.

[16] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[17] Kalyanmoy Deb, Ankur Sinha, Pekka J Korhonen, and Jyrki Wallenius. 2010. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation* 14, 5 (2010), 723–739.

[18] Petri Eskelinen, Kaisa Miettinen, Kathrin Klamroth, and Jussi Hakanen. 2010. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum* 32, 1 (2010), 211–227.

[19] Warren A Hall and Yacov Y Haimes. 1976. The surrogate worth trade-off method with multiple decision-makers. In *Multiple Criteria Decision Making Kyoto 1975*. Springer, 207–233.

[20] Imran Khan Mohd Jais and Amelia Ritahani Ismail. 2019. Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science* 2, 1 (2019), 10.

[21] Pekka Korhonen and Jyrki Wallenius. 1990. A multiple objective linear programming decision support system. *Decision Support Systems* 6, 3 (1990), 243–251.

[22] Pekka Korhonen and Guang Yuan Yu. 2000. Quadratic Pareto race. In *New Frontiers of Decision Making for the Information Technology Era*. World Scientific, 123–142.

[23] Ke Li, Guiyu Lai, and Xin Yao. 2023. Interactive evolutionary multiobjective optimization via learning to rank. *IEEE Transactions on Evolutionary Computation* 27, 4 (2023), 749–763.

[24] Manuel López-Ibáñez, Jürgen Branke, and Luís Paquete. 2021. Reproducibility in evolutionary computation. *ACM Transactions on Evolutionary Learning and Optimization* 1, 4 (2021), 1–21.

[25] Manuel López-Ibáñez and Joshua Knowles. 2015. Machine decision makers as a laboratory for interactive EMO. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 295–309.

[26] Kaisa Miettinen. 2012. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media.

[27] Kaisa Miettinen and Marko M Mäkelä. 1996. NIMBUS —- Interactive Method for Nondifferentiable Multiobjective Optimization Problems. In *Multi-Objective Programming and Goal Programming*. Springer Berlin Heidelberg, 50–57.

[28] Kaisa Miettinen and Marko M Mäkelä. 1999. Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. *Journal of the Operational Research Society* 50, 9 (1999), 949–959.

[29] Kaisa Miettinen and Marko M Mäkelä. 2002. On scalarizing functions in multiobjective optimization. *OR spectrum* 24, 2 (2002), 193–213.

[30] Hirotaka Nakayama and Yoshikazu Sawaragi. 1984. Satisficing trade-off method for multiobjective programming. In *Interactive Decision Analysis*. Springer, 113–122.

[31] Robin C Purshouse, Kalyanmoy Deb, Maszatul M Mansor, Sanaz Mostaghim, and Rui Wang. 2014. A review of hybrid evolutionary multiple criteria decision making methods. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1147–1154.

[32] Ralph E. Steuer. 1986. *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley.

[33] Theodor J Stewart. 1996. Robustness of additive value function methods in MCDM. *Journal of Multi-Criteria Decision Analysis* 5, 4 (1996), 301–309.

[34] Andrzej Wierzbicki. 1999. Reference point approaches. In *Multicriteria Decision Making*. Springer, 237–275.

[35] Andrzej P Wierzbicki. 1980. The use of reference objectives in multiobjective optimization. In *Multiple Criteria Decision Making Theory and Application*. Springer, 468–486.

[36] Deepanshu Yadav, Deepak Nagar, Palaniappan Ramu, and Kalyanmoy Deb. 2023. Visualization-aided multi-criteria decision-making using interpretable self-organizing maps. *European Journal of Operational Research* 309, 3 (2023), 1183–1200.

[37] Deepanshu Yadav, Palaniappan Ramu, and Kalyanmoy Deb. 2022. Visualization-aided multi-criterion decision-making using reference direction based Pareto race. In *2022 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 125–132.

[38] Deepanshu Yadav, Palaniappan Ramu, and Kalyanmoy Deb. 2023. Interpretable self-organizing map assisted interactive multi-criteria decision-making following Pareto-Race. *Applied Soft Computing* 149 (2023), 111032.

[39] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

Appendix

A.1 Data Generation for Training ANNs

In this section, we provide a comprehensive computational framework for generating data for training ANN1 and ANN2 in detail. For a given M-objective and n-variable problem, generate a well distributed, well-converged, and a diverse set of near Pareto optimal solutions $\left(\left[\mathbf{x}^*, \mathbf{f}^*\right]_{i=1}^{N}\right)$. Consider that $R = \mathbf{f}^R$ is an unknown M-dimensional pessimistic point and $T = \mathbf{f}^T$ is the M-dimensional target point. Machine-DM parameters described in main manuscript are δ and α. Also, $\mathbf{z}^*$ and $\mathbf{z}^{nad}$ represent ideal and nadir point, respectively.

A.1.a Data Generation for ANN1

For a M-dimensional PO point $\mathbf{f}^*$, the M-dimensional objective class $\mathbf{I}$ is computed as follows:

$$I_m = \begin{cases} 1 : \text{Class } I^{<}, & \text{if} \quad f_m^* > f_m^R \\ 2 : \text{Class } I^{\leq}, & \text{if} \quad f_m^R \geq f_m^* > f_m^T + 0.5 \times \hat{\delta}_m \\ 3 : \text{Class } I^{=}, & \text{if} \quad f_m^T - 0.5 \times \hat{\delta}_m \geq f_m^* \geq f_m^T \\ & \qquad +0.5 \times \hat{\delta}_m \\ 4 : \text{Class } I^{\geq}, & \text{if} \quad f_m^* < f_m^T - 0.5 \times \hat{\delta}_m \end{cases} ; \quad m = 1, \ldots, M. \tag{13}$$

Here, $\hat{\delta}_m = \delta_m \times (z_m^{nad} - z_m^*)$ defines the width of target objective class ($I^=$) in m-th objective. Next, for all the PO solutions

Table 3: Error metric for ANN models across benchmark MCDM problems and instances.

S.I.	Example	Objective function	ANN 1 ($\hat{\mathcal{F}}_{CL}$) Error Metric				ANN 2 ($\hat{\mathcal{F}}_{PP}$) Error Metric	
			Accuracy	Precision	Recall	F1-Score	R^2	MSE
1	Knee (two-objective)	f_1	1.0000	1.0000	1.0000	1.0000	0.9956	0.0134
		f_2	1.0000	1.0000	1.0000	1.0000	0.9965	0.0108
2	Knee (three-objective)	f_1	0.9554	0.9482	0.9296	0.9372	0.9930	0.0198
		f_2	0.9732	0.9516	0.9609	0.9550	0.9953	0.0133
		f_3	0.9286	0.8947	0.9023	0.8983	0.9780	0.0321
3	Vehicle Crashworthiness (three-objective)	f_1	0.9610	0.9495	0.9518	0.9494	0.9962	0.0026
		f_2	0.9610	0.9651	0.9453	0.9522	0.9915	0.0066
		f_3	1.0000	1.0000	1.0000	1.0000	0.9975	0.0023
4	Car side Impact (three-objective)	f_1	0.9505	0.9259	0.9489	0.9364	0.9952	0.0049
		f_2	0.9286	0.9059	0.8633	0.8752	0.9973	0.0028
		f_3	0.9670	0.9468	0.8982	0.9201	0.9950	0.0042
5	River Pollution (four-objective)	f_1	0.9950	0.9947	0.9976	0.9961	0.9987	0.0003
		f_2	0.9750	0.9421	0.9647	0.9514	0.9939	0.0001
		f_3	0.9600	0.9681	0.9059	0.9447	0.9954	0.0196
		f_4	0.9650	0.9748	0.9359	0.9527	0.9931	0.0535
6	WATER (five-objective)	f_1	0.9600	0.8344	0.9132	0.8605	0.9992	0.0008
		f_2	0.9475	0.8056	0.8211	0.8130	0.9996	0.0004
		f_3	0.9800	0.7417	0.7367	0.7391	0.9997	0.0003
		f_4	0.9775	0.7606	0.8051	0.7163	0.9994	0.0004
		f_5	0.9425	0.7073	0.7137	0.7013	0.9988	0.0012
7	NIMBUS Benchmark (six-objective)	f_1	0.9850	0.9437	0.9138	0.9275	0.9700	0.0625
		f_2	0.9600	0.8784	0.9290	0.9002	0.9750	0.0532
		f_3	0.9750	0.8998	0.9193	0.9087	0.9500	0.1232
		f_4	0.9550	0.8562	0.9743	0.9009	0.9350	0.1340
		f_5	0.9600	0.8063	0.8040	0.8048	0.9200	0.1628
		f_6	0.9800	0.6845	0.7067	0.6947	0.9700	0.0769
8	DTLZ2 (eight-objective)	f_1	0.9503	0.9057	0.9121	0.9080	0.9830	0.0478
		f_2	0.9526	0.9184	0.9159	0.9161	0.9756	0.0625
		f_3	0.9464	0.9364	0.8994	0.9165	0.9786	0.0601
		f_4	0.9619	0.9382	0.9199	0.9278	0.9786	0.0677
		f_5	0.9611	0.9513	0.9603	0.9553	0.9882	0.0383
		f_6	0.9580	0.9358	0.9244	0.9299	0.9860	0.0438
		f_7	0.9565	0.9430	0.9249	0.9319	0.9845	0.0581
		f_8	0.9775	0.9727	0.9743	0.9734	0.9926	0.0222

$\left(\left[\mathbf{x}^*, \mathbf{f}^*\right]_{i=1}^{N}\right)$ obtain the class according to Equation (13). Get the training data ($\mathcal{D}_{ANN1}$) for ANN1 as follows:

$$\mathcal{D}_{ANN1} = \left[\mathbf{x}^{*,(i)}, \mathbf{I}^{(i)}\right]_{i=1}^{N} \tag{14}$$

Next, the data $\mathcal{D}_{ANN1}$ is used for train-test-validation split and for training ANN1 ($\mathcal{F}_{CL}$).

A.1.b Data Generation for ANN2

In this section, we provide the computational framework for generating training data for ANN2 ($\mathcal{F}_{PP}$). Following the objective classification definition provided in Equation (13), compute the M-dimensional bounding parameter ($\mathbf{\Lambda}$) as follows:

$$\Lambda_m = \begin{cases} 0 & \text{if } I_m \in \mathbf{I}^{<} \\ \bar{z}_m, & \text{if } I_m \in \mathbf{I}^{\leq} \\ 0 & \text{if } I_m \in \mathbf{I}^{=} \\ \varepsilon_m, & \text{if } I_m \in \mathbf{I}^{>} \end{cases} ; \; m = 1, \ldots, M. \tag{15}$$

The detailed computation for $\bar{z}$ and ε is provided in Appendix A of the main document. Next, for all the PO solutions $\left(\left[\mathbf{x}^*, \mathbf{f}^*\right]_{i=1}^{N}\right)$, compute the bounding parameter vector ($\mathbf{\Lambda}$). Get the training data ($\mathcal{D}_{ANN2}$) as follows:

$$\mathcal{D}_{ANN2} = \left[\mathbf{x}^{*,(i)}, \mathbf{\Lambda}^{(i)}\right]_{i=1}^{N}. \tag{16}$$

Next, the data $\mathcal{D}_{ANN2}$ is used for train-test-validation split and for training ANN2 ($\mathcal{F}_{PP}$).

Table 4: A generic architecture of ANN1–Class Predictor ANN ($\mathcal{F}_{CL}$) for an n-variable, M-objective MOO problem, trained with fixed hidden (dense) layer [128,64,32,16].

Layer Type	Output Shape	Number of Parameters	Connected to
Input Layer	n	0	–
Dense Layer 1	128	$128(n+1)$	Input Layer
Dense Layer 2	64	8,256	Dense Layer 1
Dense Layer 3	32	2,080	Dense Layer 2
Dense Layer 4	16	528	Dense Layer 3
Output Layer 1	1	68	Dense Layer 4
⋮	⋮	⋮	⋮
Output Layer M	1	68	Dense Layer 4
Total Trainable Parameters:	$128n + 10,992 + 68M$		

Table 5: A generic architecture of ANN2–Parameter Predictor ANN ($\mathcal{F}_{PP}$) for an n-variable, M-objective MOO problem, trained with fixed hidden (dense) layer [128,64,32,16].

Layer Type	Output Shape	Number of Parameters	Connected to
Input Layer	n	0	–
Dense Layer 1	128	$128(n+1)$	Input Layer
Dense Layer 2	64	8,256	Dense Layer 1
Dense Layer 3	32	2,080	Dense Layer 2
Dense Layer 4	16	528	Dense Layer 3
Output Layer 1	1	17	Dense Layer 4
⋮	⋮	⋮	⋮
Output Layer M	1	17	Dense Layer 4
Total Trainable Parameters:	$128n + 10,992 + 17M$		

A.2 Architecture and Error Metric for ANNs

The architecture of ANN1 and ANN2 is provided in Table 4 and Table 5, respectively. For ANN1 and ANN2, four-layered feed-forward neural networks are used with a reduced number of hidden layers [128, 64, 32, 16] and A*dam optimizer* [20] with a learning rate set to lr_1 and lr_2, respectively. ANN1 uses *Softmax* and ANN2 uses *Sigmoid* activation functions. For ANN2, L2-regularization with parameter l_{reg} is used to train the respective ANN. For both the ANNs, 80% of data is used in training and 20% data is used for testing. In training phase, 20% of training data is used for cross-validation purpose. The batch-size and maximum epoch for both the ANNs are set to 32 and 500, respectively. For training ANN1, error metrics – Accuracy, Precision, Recall, and F-1 Score – are checked for their closeness to one. For training ANN2, R^2 and MSE is used as error metric.

The error metric of class-predictor ML ($\hat{\mathcal{F}}_{CL}$) is provided in terms of Accuracy, Precision, Recall, and F-1 Score. For parameter-predictor ML ($\hat{\mathcal{F}}_{PP}$), coefficient of regression (R^2) and mean squared error (MSE) is used indicate the ANN performance and error metric. The error metrics for class-predictor ML ($\hat{\mathcal{F}}_{CL}$) and parameter-predictor ML ($\hat{\mathcal{F}}_{PP}$) are provided in Table 3.

A.3 NIMBUS procedure

This section illustrates the use of the above Machine-DM concept within a popular iMCDM procedure – NIMBUS [27]. To suit our Machine-DM concept within NIMBUS, we modify the original NIMBUS procedure slightly. The modified procedure is named as MachDM-NIMBUS here.

Consider the following n-variable, M-objective constrained multi-objective optimization (MOO) problem:

$$\begin{aligned} \text{Minimize} \quad & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \text{subject to} \quad & g_j(\mathbf{x}) \le 0, \quad j = 1, \ldots, J, \\ & h_k(\mathbf{x}) = 0, \quad k = 1, \ldots, K, \\ & x_i^{(L)} \le x_i \le x_i^{(U)}, \quad i = 1, \ldots, n. \end{aligned} \tag{17}$$

First, we present the original NIMBUS procedure. Consider the MOO problem represented in Equation MOO. NIMBUS performs objective (f_i) classification into the following five classes [28], based on the objective vector $\mathbf{f}(\mathbf{x}^h)$ at the current solution $\mathbf{x}^h$, as follows:

(1) whose values should be minimized further requiring $f_i(\mathbf{x}) < f_i(\mathbf{x}^h)$, for $i \in \mathbf{I}^<$,
(2) whose values should be minimized till some aspiration level $\bar{z}_i \left(< f_i(\mathbf{x}^h)\right)$, such that $f_i(\mathbf{x}) \ge \bar{z}_i$ for $i \in \mathbf{I}^\le$,
(3) whose current values are satisfactory at the moment ($i \in \mathbf{I}^=$),
(4) whose values are allowed to increase until or up to a specific upper bound $\varepsilon_i \left(\ge f_i(\mathbf{x}^h)\right)$, such that $f_i(\mathbf{x}) \le \varepsilon_i$ for $i \in \mathbf{I}^\ge$, and
(5) whose values can change freely ($i \in \mathbf{I}^\diamond$).

Let $f(\mathbf{x}^h)$ be the feasible start point at an iteration, at which the classification is performed. Further to the classification, the following augmented achievement scalarization function (AASF) *subproblem* is formulated and solved:

$$\begin{aligned} \underset{(\mathbf{x})}{\text{Minimize}} \quad & \max_{i \in I^<, j = I^\le} \left[\frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^{**}}, \frac{f_j(\mathbf{x}) - \bar{z}_j}{z_j^{\text{nad}} - z_j^{**}} \right] + \rho \sum_{i=1}^{M} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{**}}, \\ \text{subject to} \quad & f_i(\mathbf{x}) \le f_i(\mathbf{x}^h), \quad i \in \mathbf{I}^< \cup \mathbf{I}^\le \cup \mathbf{I}^=, \\ & f_i(\mathbf{x}) \le \varepsilon_i, \quad i \in \mathbf{I}^\ge, \\ & \mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_J(\mathbf{x}))^T \le 0, \end{aligned} \tag{18}$$

where z_i^{nad} and z_i^* are nadir and ideal objective values, respectively, of the i-th objective, and z_i^{**} (slightly smaller than z_i^*) is the utopian objective value. These values are computed at the start of the NIMBUS procedure. While it is possible to obtain the ideal objective vector by minimizing each constrained objective problem individually, the nadir objective vector can only be estimated [15]. The parameter ρ (>0) is usually chosen to be a small scalar value and the term including ρ ensures that the resulting optimal solution of the above problem is a Pareto-optimal point, and not a weak Pareto-optimal point [26].

A.3.a MachDM-NIMBUS Procedure

We propose a Machine-based decision-making procedure using the NIMBUS method (MachDM-NIMBUS), but without considering the objective class $I^{\diamond}$. This is because such an objective class is not used in the NIMBUS sub-problem (Equation 18) and may not occur in most iterations of the decision-making process. For a large number of objectives, a few objectives may fall under this class in some iterations, but we ignore this class for this study. We only consider the objectives that can be either improved (in $\mathbf{I}^{<} \cup \mathbf{I}^{\leq}$) or relaxed (in $\mathbf{I}^{\geq}$) or acceptable (in $\mathbf{I}^{=}$) at its current level. The steps used in NIMBUS method, combined with proposed class-predictor ML ($\mathcal{F}_{CL}$), parameter-predictor ML ($\mathcal{F}_{PP}$), and preference-evaluator ($\mathcal{M}_{PE}$) based Machine-DM steps (in italics), are provided in the following:

(1) Find a starting feasible point $\mathbf{x}^0$ by solving the following auxiliary problem:

$$\begin{aligned} &\text{Minimize} && \max\left[0, g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})\right], \\ &\text{subject to} && \mathbf{x} \in \mathbb{R}^n. \end{aligned} \tag{19}$$

Then, find a weakly PO solution $\mathbf{x}^1$ by setting $i \in \mathbf{I}^{<}$ for all $i = 1, \ldots, M$ and solving the selected subproblem presented in Equation (18). Set the iteration counter $h = 1$.

(2) *Use ANN1 ($\hat{\mathcal{F}}_{CL}$) to predict the objective classes* $\mathbf{I}$ at the point $\mathbf{z}^h = \mathbf{f}(\mathbf{x}^h)$ such that $\mathbf{I}^{\geq} \cup \mathbf{I}^{=} \neq \emptyset$ and $\mathbf{I}^{<} \cup \mathbf{I}^{\leq} \neq \emptyset$, meaning that there is at least one objective which can be relaxed to improve at least one other objective. If either of these unions is empty (meaning that no improvement or relaxation is possible), go to Step (9). Otherwise, *use ANN2 ($\hat{\mathcal{F}}_{PP}$) to predict the aspiration levels $\bar{z}_i^h$* for $i \in \mathbf{I}^{\leq}$ and the upper bounds ε_i^h for $i \in \mathbf{I}^{>}$. *Compute the preference value of* $\mathbf{x}^h$ *using the preference-evaluator* $\mathcal{M}_{PE}$.

(3) Calculate $\hat{\mathbf{x}}^h$ by solving the subproblem represented in Equation 18. If $\hat{\mathbf{x}}^h = \mathbf{x}^h$, *check if class-predictor ML (ANN1) can predict an alternate classification (discussed as an extension of our proposed Machine-DM in Section* **??***).* If yes, set $\mathbf{x}^{h+1} = \mathbf{x}^h, h = h + 1$, and go to Step (2); if not, go to Step (9) to conclude the final step of the procedure. If $\hat{\mathbf{x}}^h \neq \mathbf{x}^h$, go to Step (4).

(4) At this step, $\hat{\mathbf{x}}^h$ is a different solution than $\mathbf{x}^h$. Denote $\hat{\mathbf{z}}^h = \mathbf{f}(\hat{\mathbf{x}}^h)$. Predict the objective classification and bounding parameters of $\hat{\mathbf{x}}^h$ using two ANN models. Then, compute the preference metric $\mathcal{M}_{PE}$ value at $\hat{\mathbf{x}}^h$. *If objective classes for* $\mathbf{z}^h$ *and* $\hat{\mathbf{z}}^h$ *are significantly different (based on cosine similarity discussed in Section 7)*, compute P intermediate solutions between $\mathbf{x}^h$ and $\hat{\mathbf{x}}^h$ by setting a direction $\mathbf{d}^h = \hat{\mathbf{x}}^h - \mathbf{x}^h$ and moving to Step (6). In the original iMCDM procedure, a human DM is expected to make the decision on whether the difference is significant and determine the size of P. In the event $\mathbf{z}^h$ and $\hat{\mathbf{z}}^h$ are not significantly different, go to Step (5).

(5) At this step, another subjective evaluation was called for in the original NIMBUS procedure, but *we achieve here using our $\mathcal{M}_{PE}$ metric.* If $\mathbf{z}^h$ is preferred to $\hat{\mathbf{z}}^h$ based in the metric value, set $\mathbf{x}^{h+1} = \mathbf{x}^h$, $h = h + 1$, and go to Step (2). Otherwise, the Machine-DM continues from $\hat{\mathbf{z}}^h$. Set $\mathbf{x}^{h+1} = \hat{\mathbf{x}}^h$, $h = h + 1$, and go to Step (2) [1] .

(6) Machine-DM specifies the desired number of P alternatives and calculate P vectors $\mathbf{f}(\mathbf{x}^h + t_j\mathbf{d}^h)$, $j = 2, \ldots, P - 1$, where $t_j = (j - 1)/(P - 1)$.

(7) Find a weakly PO solution by solving the following auxiliary problem for each t_j:

$$\begin{aligned} &\text{Minimize} && \max_{i=1,\ldots,M} \left[f_i(\mathbf{x}) - f_i(\mathbf{x}^h + t_j\mathbf{d}^h)\right], \\ &\text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{20}$$

where S is the set of feasible solutions.

(8) *Present P alternative solutions found in Step (7) to Machine-DM to rank the solutions using the preference-evaluator $\mathcal{M}_{PE}$ to obtain the most preferred solution.* Denote the corresponding decision vector by $\mathbf{x}^{h+1}$ and set $h = h + 1$. If the classification of each objective $i \notin (\mathbf{I}^{=} \cup \mathbf{I}^{\geq})$ (meaning there is still some objectives which must be improved), go to Step (2), else go to Step (9).

(9) Finally, check the Pareto-optimality of $\mathbf{x}^h$ by solving the following auxiliary problem and compute the solution $(\tilde{\mathbf{x}}, \tilde{\gamma})$:

$$\begin{aligned} &\underset{(\mathbf{x},\boldsymbol{\gamma})}{\text{Maximize}} && \sum_{i=1}^{M} \gamma_i, \\ &\text{subject to} && f_i(\mathbf{x}) + \gamma_i \leq f_i(\mathbf{x}^h), \quad i \in \mathbf{I}^{<}, \\ & && \gamma_i \geq 0, \quad i \in \mathbf{I}^{<}, \\ & && \mathbf{g(x)} = (g_1(\mathbf{x}), \ldots, g_J(\mathbf{x}))^T \leq 0. \end{aligned} \tag{21}$$

(10) If $\tilde{\mathbf{x}} = \mathbf{x}^h$, stop with the final solution $(\mathbf{x}^h)$, else, go to Step (2) with $\mathbf{x}$.

Note that the above procedure requires the Machine-DM to be called in the following steps:

- Step (2) for obtaining classification status and bounding parameter values using ANN models and for using the predictor-evaluator metric for comparison purpose,
- Step (3) for deciding whether to go for an alternate classification task if two consecutive solutions are identical, for which we propose later in Subsection **??** stochastic ANN models for handling this case,
- Step (4) for deciding whether to choose intermediate points or to continue, which we handle using the cosine similarity condition, given in Subsection 7,
- Step (5) for choosing the better of two solutions, which we achieve using the preference-evaluator metric, and
- Step (8) for selecting the most preferred intermediate solution, which we also achieve using the preference-evaluator metric.

We increment the number of DM calls when either Step (2) or Step (3) is called, due to a relatively more-involved decision-making task. Other DM events are achieved using preference-evaluator metric or cosine similarity condition, which use mathematical equations.

[1] Since, subproblem solution problem is always weakly Pareto optimal, the remaining part of original step is ignored.

Table 6: Iteration-wise solutions obtained in MachDM-iMCDM procedures for two-objective knee example with start point $\mathbf{x}^{(0)}$=[1.50, 0.01, 0.01, 0.01, 0.01].

MachDM-STEM procedure								
Points	f_1	f_2	I_1	I_2	$\bar{z}_1/\varepsilon_1$	$\bar{z}_2/\varepsilon_2$	$\mathcal{M}_{PE}$	Rank
A: $\mathbf{z}^{(1)}$	7.7000	0.0000	1	4	–	0.8118	2.7361	6
B: $\mathbf{z}^{(2)}$	6.5741	0.8118	1	4	–	1.4289	2.5457	5
C: $\mathbf{z}^{(3)}$	6.0608	1.4288	2	4	5.4986	1.9746	1.3773	4
D: $\mathbf{z}^{(4)}$	5.6709	1.9744	2	4	4.9221	2.4945	1.2169	3
E: $\mathbf{z}^{(5)}$	4.5198	2.4943	2	4	4.3556	3.0054	1.0592	2
F: $\mathbf{z}^{(6)}$	4.3769	3.0048	3	3	–	–	0.0167	1

MachDM-STOM procedure								
Points	f_1	f_2	I_1	I_2	$\bar{z}_1/\varepsilon_1$	$\bar{z}_2/\varepsilon_2$	$\mathcal{M}_{PE}$	Rank
A: $\mathbf{z}^{(1)}$	7.7000	0.0000	1	4	–	0.8118	2.7361	9
B: $\mathbf{z}^{(2)}$	6.2548	1.0676	2	4	5.8465	1.6609	1.4741	8
C: $\mathbf{z}^{(3)}$	6.1577	1.1987	2	4	5.7172	1.7774	1.4381	7
D: $\mathbf{z}^{(4)}$	6.0815	1.3671	2	4	5.5565	1.9224	1.3934	6
E: $\mathbf{z}^{(5)}$	6.0006	1.6047	2	4	5.3328	2.1241	1.3311	5
F: $\mathbf{z}^{(6)}$	5.7510	1.9238	2	4	4.9901	2.4331	1.2358	4
G: $\mathbf{z}^{(7)}$	5.0330	2.1976	2	4	4.5053	2.8703	1.1009	3
H: $\mathbf{z}^{(8)}$	4.5460	2.4502	2	4	4.3711	2.9914	1.0635	2
I: $\mathbf{z}^{(9)}$	4.4791	2.6078	3	4	–	3.0379	0.0492	1

MachDM-GUESS procedure								
Points	f_1	f_2	I_1	I_2	$\bar{z}_1/\varepsilon_1$	$\bar{z}_2/\varepsilon_2$	$\mathcal{M}_{PE}$	Rank
A: $\mathbf{z}^{(1)}$	7.7000	0.0000	1	4	–	0.8118	2.7361	5
B: $\mathbf{z}^{(2)}$	2.1433	5.2300	4	1	2.5981	0.0000	2.4282	4
C: $\mathbf{z}^{(3)}$	2.3928	4.5953	4	2	3.0282	4.2024	1.3084	3
D: $\mathbf{z}^{(4)}$	5.6684	1.9759	2	4	4.9201	2.4963	1.2163	2
E: $\mathbf{z}^{(5)}$	3.7243	3.2752	4	3	4.0152	–	0.0338	1

MachDM-NIMBUS procedure								
Points	f_1	f_2	I_1	I_2	$\bar{z}_1/\varepsilon_1$	$\bar{z}_2/\varepsilon_2$	$\mathcal{M}_{PE}$	Rank
A: $\mathbf{z}^{(1)}$	7.7000	0.0000	1	4	–	0.8118	2.7361	6
B: $\mathbf{z}^{(2)}$	6.5746	0.8115	1	4	–	1.4286	2.5458	5
C: $\mathbf{z}^{(3)}$	6.0610	1.4283	2	4	5.4991	1.9741	1.3774	4
D: $\mathbf{z}^{(4)}$	5.6720	1.9738	2	4	4.9230	2.4937	1.2171	3
E: $\mathbf{z}^{(5)}$	4.5230	2.4881	2	4	4.3577	3.0035	1.0598	2
F: $\mathbf{z}^{(6)}$	4.3776	3.0035	3	3	–	–	0.0168	1

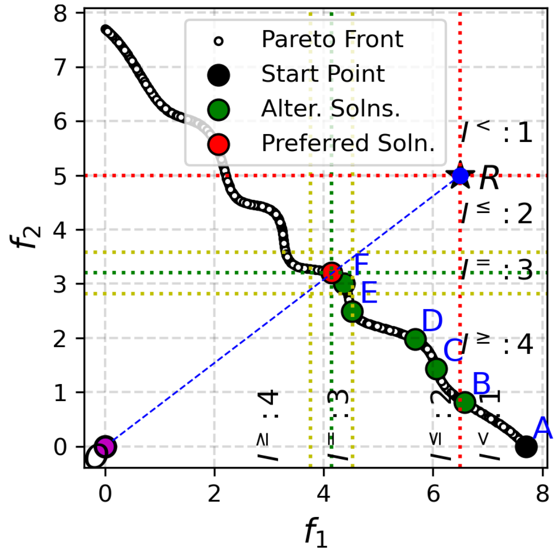


Figure 8: MachDM-STEM solutions obtained for knee example with start point $\mathbf{x}^{(0)}$=[1.50, 0.01, 0.01, 0.01, 0.01].

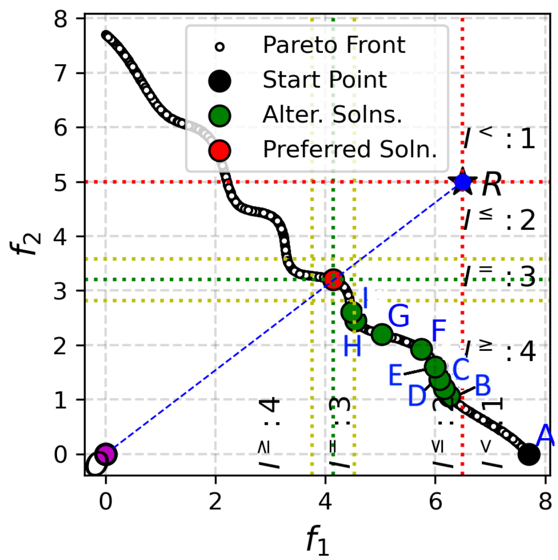


Figure 9: MachDM-STOM solutions obtained for knee example with start point $\mathbf{x}^{(0)}$=[1.50, 0.01, 0.01, 0.01, 0.01].

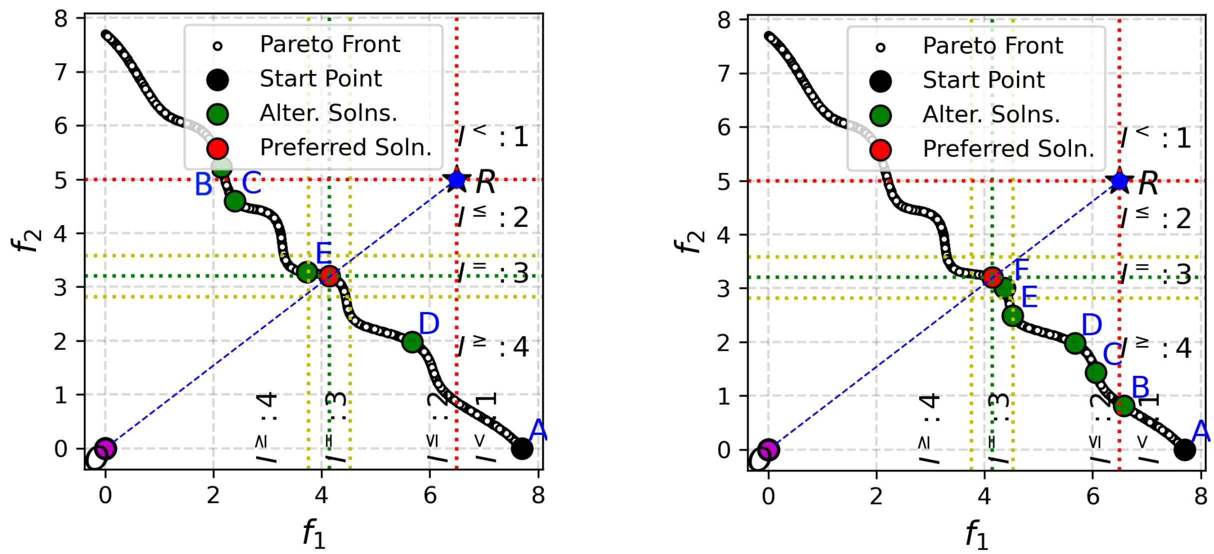
A.3.b Decision for Choosing Intermediate Solutions in MachDM-NIMBUS in Step (4)

This section describes the procedure for identifying the steps responsible for recognizing and initiating the computation of intermediate solutions in the MachDM-NIMBUS procedure. We propose assessing the similarity between two objective classes (predicted using ANN1) obtained in consecutive iterations. When the *cosine similarity* between these consecutive classes becomes negative, the computation of intermediate solutions in NIMBUS (Step (4)) is initiated.

Let $\mathbf{z}^h$ and $\hat{\mathbf{z}}^h$ denote the two NIMBUS solutions evaluated in Step (4), with their corresponding predicted objective classes with $\mathbb{Z}$ represented as $\mathbf{I}^h$ and $\hat{\mathbf{I}}^h$, respectively. The cosine similarity between them is calculated using the following equation:

$$\text{Sim}(\mathbf{z}^h, \hat{\mathbf{z}}^h) = (\mathbf{I}^h - \mathbf{I}^T) \cdot (\hat{\mathbf{I}}^h - \mathbf{I}^T), \tag{22}$$

where $\mathbf{I}^T = [3, \ldots, 3]$. If Sim < 0, meaning that two points are on opposite side of T (trade-off information are dramatically different between $\mathbf{z}^h$ to $\hat{\mathbf{z}}^h$), the decision is to compute some intermediate solutions to proceed with the procedure. Another decision-making



(a) MachDM-GUESS solutions **(b) MachDM-NIMBUS solutions**

Figure 10: MachDM-iMCDM solutions obtained for the knee example with start point $\mathbf{x}^{(0)} = [1.50, 0.01, 0.01, 0.01, 0.01]$.

Table 7: Statistical comparison of MachDM-iMCDM procedures. The median values of the performance metrics are provided for four independent runs for each MachDM-iMCDM procedure per examples, using four start points. Metrics h_E, η_E, Φ_E, and N_{evl} denotes the median values of number of DM calls, success rate, and deviation from the target class, respectively.

Example	MachDM-STEM				MachDM-STOM				MachDM-GUESS				MachDM-NIMBUS			
	h_E	η_E	Φ_E	N_{evl}^E	h_E	η_E	Φ_E	N_{evl}^E	h_E	η_E	Φ_E	N_{evl}^E	h_E	η_E	Φ_E	N_{evl}^E
Knee (2-obj.)	6.0	**1.0**	**0.125**	1.40E+05	7.0	**1.0**	0.250	1.60E+05	**3.5**	**1.0**	0.250	**9.00E+04**	6.0	**1.0**	**0.125**	1.40E+05
Knee (3-obj.)	6.0	0.5	0.250	1.40E+05	5.5	**1.0**	**0.083**	1.30E+05	**5.0**	**1.0**	0.167	**1.20E+05**	6.0	**1.0**	0.167	1.40E+05
Crashworthiness	5.0	0.5	0.083	1.20E+05	4.0	**1.0**	**0.000**	1.00E+05	**2.0**	**1.0**	**0.000**	**6.00E+04**	6.0	0.0	0.167	1.40E+05
Car Side Impact	8.0	0.5	0.083	1.80E+05	9.0	**1.0**	0.083	2.00E+05	**3.5**	**1.0**	**0.000**	**9.00E+04**	7.5	**1.0**	**0.000**	1.70E+05
River Pollution	**6.0**	0.0	0.250	1.40E+05	11.0	0.0	0.500	2.40E+05	**6.0**	**1.0**	**0.000**	**1.40E+05**	6.5	0.0	0.250	1.50E+05
WATER	**4.0**	0.0	0.300	**1.00E+05**	11.0	0.0	0.200	2.40E+05	4.5	**1.0**	**0.000**	1.10E+05	**4.0**	0.0	0.200	**1.00E+05**
NIMBUS bench.	**7.0**	**0.0**	0.417	**1.60E+05**	11.0	**0.0**	**0.250**	2.40E+05	11.0	**0.0**	**0.333**	2.40E+05	8.5	**0.0**	0.375	1.90E+05
DTLZ2 (8-obj.)	11.0	0.0	0.344	2.40E+05	11.0	0.0	0.531	2.40E+05	**10.5**	**0.5**	**0.313**	**2.30E+05**	11.0	0.0	0.563	2.40E+05

***Note: A detailed table with simulation with four different start points implementing four MachDM-iMCDM procedures is provided in Appendix.**

input from a human DM is to provide the number of intermediate points (P). In this study, we choose $P = 1$, but a suitable size of P based on the difference $|\mathbf{z}^h - \hat{\mathbf{z}}^h|$ can be set as an extension of our current approach.

A.4 Additional Results

This section presents the summarized Table of comparison for MachDM-iMCDM procedures using the four metrics– h^E, η^E, ϕ^E, and N_{evl}^E. Table 7 present the metrics for each problem considering four different start point ($\mathbf{x}^{(0)}$). The list of start point for each examples are provided in Table 8.

8 Conclusions

Furthermore, the main manuscript outlines a promising direction for future work: incorporating a stochastic ANN-based class-prediction mechanism to address premature termination issues observed in certain cases. A stochastic class predictor, by allowing probabilistic transitions between classes, is expected to improve adaptability and ensure smoother convergence in boundary or ambiguous regions of the objective space. We plan to enhance the parameter predictor to address situations where the STEM and NIMBUS subproblem constraints provided in Equation (18) make the search infeasible and cause premature convergence.

Table 8: Initial start points ($\mathbf{x}^{(0)}$) used for different benchmark and real-world examples.

Examples	$\mathbf{x}_1^{(0)}$	$\mathbf{x}_2^{(0)}$	$\mathbf{x}_3^{(0)}$	$\mathbf{x}_4^{(0)}$
Knee (2-obj.)	[0.10, 0.01, 0.01, 0.01, 0.01]	[1.50, 0.01, 0.01, 0.01, 0.01]	[5.50, 0.01, 0.01, 0.01, 0.01]	[7.50, 0.01, 0.01, 0.01, 0.01]
Knee (3-obj.)	[0.0223, 0.1075, 0.6478, 0.9607, 0.7703]	[0.7608, 0.0464, 0.0033, 0.7605, 0.2769]	[0.2524, 0.3185, 0.7829, 0.7197, 0.7687]	[0.8005, 0.3790, 0.4249, 0.3394, 0.9044]
Crashworth.	[1.0446, 1.2149, 2.2957, 2.9214, 2.5406]	[2.5216, 1.0928, 1.0065, 2.5210, 1.5539]	[1.5048, 1.6371, 2.5658, 2.4394, 2.5374]	[2.6010, 1.7581, 1.8499, 1.6788, 2.8089]
Car Side Impact	[0.5223, 0.5467, 1.1478, 1.4607, 2.2230, 1.1266, 0.5824]	[1.2608, 0.4918, 0.5033, 1.2605, 1.3597, 0.8756, 1.0429]	[0.7524, 0.7367, 1.2829, 1.2197, 2.2202, 0.5706, 1.0727]	[1.3005, 0.7911, 0.9249, 0.8394, 2.4578, 0.5271, 0.4899]
River Pollution	[0.3156, 0.3752]	[0.8326, 0.3325]	[0.4767, 0.5230]	[0.8604, 0.5653]
WATER	[0.0198, 0.0197, 0.0683]	[0.3448, 0.0142, 0.0103]	[0.1210, 0.0387, 0.0805]	[0.3622, 0.0441, 0.0482]
NIMBUS bench.	[1.0446, 1.2150]	[2.5217, 1.0928]	[1.5048, 1.6371]	[2.6010, 1.7581]
DTLZ2 (8-obj.)	[0.0223, 0.1075, 0.6478, 0.9607, 0.7703, 0.9083, 0.2279, 0.2538, 0.3495, 0.4569]	[0.7608, 0.0464, 0.0033, 0.7605, 0.2769, 0.5946, 0.8036, 0.7403, 0.1239, 0.2439]	[0.2524, 0.3186, 0.7829, 0.7197, 0.7687, 0.2132, 0.8409, 0.9689, 0.1297, 0.1722]	[0.8005, 0.3790, 0.4249, 0.3394, 0.9044, 0.1588, 0.1123, 0.7237, 0.0841, 0.0734]