Gliding over the Pareto Front with Uniform Designs

Xiaoyuan Zhang^{a}**, Genghui Li** ^{b}**, Xi Lin** ^{a}**, Yichi Zhang**^{c}**, Yifan Chen**^{d}**, Qingfu Zhang**^{a} *
^{a} Department of Computer Science, City University of Hong Kong;
^{b} College of Computer Science and Software Engineering, Shenzhen University
^{c} Department of Statistics, Indiana University Bloomingtom
^{d} Departments of Mathematics and Computer Science, Hong Kong Baptist University

Abstract

Multiobjective optimization (MOO) plays a critical role in various real-world domains. A major challenge therein is generating K uniform Pareto-optimal solutions to represent the entire Pareto front. To address this issue, this paper firstly introduces *fill distance* to evaluate the K design points, which provides a quantitative metric for the representativeness of the design. However, directly specifying the optimal design that minimizes the fill distance is nearly intractable due to the nested $\min - \max - \min$ optimization problem. To address this, we propose a surrogate "max-packing" design for the fill distance design, which is easier to optimize and leads to a rate-optimal design with a fill distance at most $4\times$ the minimum value. Extensive experiments on synthetic and real-world benchmarks demonstrate that our proposed paradigm efficiently produces high-quality, representative solutions and outperforms baseline methods.

1 Introduction

Multiobjective optimization (MOO) is widely used to inform real-world decision-making across various fields, including materials science [20, 46, 5, 30], recommendation systems [29, 61, 31], and industrial design [44, 52, 50, 57]. An MOO problem (MOP) involves optimizing multiple conflicting objectives, which can be (informally) formulated as:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})), \tag{1}$$

where m is the number of objectives. Equation (1) is a vector optimization problem and it does not admit a total ordering, therefore, to compare solutions, the concept of *Pareto optimality* is introduced. A solution is deemed *Pareto optimal* if no other solution $\boldsymbol{x}' \in \mathcal{X}$ *dominates* it. Domination occurs when $f_i(\boldsymbol{x}') \leq f_i(\boldsymbol{x})$ for all $i = 1, \ldots, m$, with at least one strict inequality [37, 19]. The image $\boldsymbol{f}(\boldsymbol{x})$ of a Pareto optimal solution is referred to as a *Pareto objective* in this paper. The set of all Pareto optimal solutions is called the Pareto set (PS), and their corresponding objectives form the Pareto front (PF).

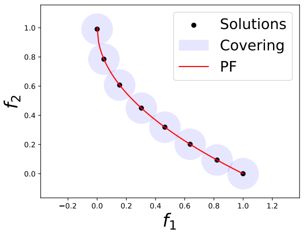


Figure 1: **Covering of a Pareto Front (PF)**. Eight diverse Pareto objectives are used to cover the entire PF with a small covering radius.

Under mild conditions, a PS or PF forms a continuous (m-1)-dim manifold containing infinitely many solutions [25]. For a general MOP, it is intractable to precisely depict the entire PS or PF with a closed-form expression. Researchers thus turns to use a small number K of diverse Pareto optimal objectives to "represent" the entire PF. Several prior works have focused on generating

---

[*] Corresponding to Qingfu Zhang. Contact: {xzhang2523-c@my.,qingfu.zhang@}cityu.edu.hk. The source code is integrated into the LibMOON library, available at `https://github.com/xzhang2523/libmoon`.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

diverse solutions (see Section 2), but they lack formal definitions of representability and uniformity. In this paper, we define representability as the covering radius of a size-K solution set that covers the entire PF. An illustrative example is provided in Figure 1, where eight uniformly distributed solutions cover the entire PF with a small radius.

In this paper, we first introduce the fill distance (FD) as the minimal covering radius of the Pareto objectives, which measures how well a set of discrete solutions represents the true PF. A configuration with a smaller covering radius is considered a better representation of the PF. Next, we demonstrate that optimizing FD is challenging due to its nested $\min - \max - \min$ structure (Equation (4)). To overcome this, we maximize the minimal $(\max - \min)$ pairwise distances between Pareto objectives, which bounds the minimal covering radius up to a constant. Finally, we propose a bi-level optimization framework based on neural network approximation to efficiently solve this $\max - \min$ problem.

Our method, *UMOD* (Uniform Multi-Objective optimization based on Decomposition), extends the decomposition-based MOO paradigm [58]. We conduct comparative evaluations against methods on complex multiobjective problems with numerous local optimas and on fairness classification problems with thousands of decision variables. Empirical results demonstrate the effectiveness of our proposal. The contribution of this paper can be summarized as:

1. We introduce the fill distance as a uniformity metric for Pareto objectives in MOO. We prove that the fill distance serves as an upper bound for the Inverted Generational Distance (IGD), and that the max-packing design is rate-optimal with respect to the fill distance. Therefore, the max-packing design provides an effective surrogate for the fill distance design.
2. We present a practical algorithm for identifying the maximum-packing design of Pareto objectives, and formulate it as a bi-level optimization problem. To expedite the optimization process, we introduce a neural network to replace the frequent solving of the inner-loop optimization problem. Additionally, we analyze the optimization bounds of the neural network-based approach in comparison to solving the original optimization problem directly.
3. Finally, we evaluate our approach against leading MOO methods, including evolutionary algorithms and gradient-based algorithms, on both synthetic and real-world problems. UMOD outperforms these methods in terms of both uniformity and efficiency, based on commonly used metrics in multiobjective optimization.

Notations. In this paper, $\rho(\boldsymbol{y}^{(a)}, \boldsymbol{y}^{(b)})$ represents the Euclidean distance between vectors $\boldsymbol{y}^{(a)}$ and $\boldsymbol{y}^{(b)}$, with bold letters for vectors. Superscripts distinguish vectors, and subscripts (e.g., y_i) indicate vector elements. A PF is denoted by $\mathcal{T}$. The objective space is $\mathcal{Y} = \{\boldsymbol{y} \mid \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\}$. Δ_m is the m-D preference simplex; $\Delta_m = \{\boldsymbol{y} \mid \sum_{i=1}^{m} y_i = 1, y_i \geq 0, i \in [m]\}$, where $[m] = \{1, \ldots, m\}$. Black bold notations (e.g., $\mathbb{Y}$) denote solution sets.

2 Related works

In this section, we review three lines of works to generate uniform or diverse Pareto objectives. We focus our discussions on uniform/diverse *Pareto objectives* rather than *Pareto solutions* because our goal is to produce uniform Pareto solutions in the objective space ($\mathcal{Y}$), not the decision space ($\mathcal{X}$).

2.1 Methods to generate diverse Pareto objectives

Various MOO methods effectively generate diverse Pareto objectives, for both *gradient-based* and *evolution-computation (EC)-based* frameworks. In the line of gradient-based methods, Pareto MultiTask Learning (PMTL) [32] produces Pareto objectives which are constrained in specific regions (sectors); MOO with Stein Variational Gradient Descent (MOO-SVGD) models objective vectors as particles, updating them through repulsive forces with a kernel function to maximize their separation; Exact Pareto Optimization (EPO) [36] aligns solutions with user-specific preference vectors, fostering a diverse distribution of Pareto objectives by specifying diverse preferences. For multiobjective evolutionary algorithms (MOEAs), NSGA2 [14] introduces the crowding distance and Pareto rank to achieve a diverse distribution of Pareto objectives; NSGA3 [13, 27] introduces a new diversity preservation technique based on reference points; MOEA/D [58] and its variants generate diverse Pareto objectives by leveraging the positional relationship between preference vectors and Pareto objectives; Hypervolume-based methods (e.g., SMS-MOEA [6]) maximize the set of solutions with the largest hypervolume to enhance diversity. A key distinction of the proposed UMOD method with the previously mentioned methods is that, for general MOPs, the distribution of the achieved Pareto objectives remains unknown, whereas, important distribution characteristics of its achieved Pareto objectives can be determined by UMOD.

2.2 Subset selection for multiobjective optimization

Another approach to generating a size-K diverse Pareto set is subset selection. This method first generates a large number of Pareto solutions, then selects K solutions to maximize hypervolume or minimize IGD [23, 53, 9, 45]. Subset selection, a discrete optimization problem, is generally inefficient to solve compared with continuous optimization problems. Recently, some approaches employ greedy algorithms [9, 33, 28] to obtain approximate solutions, with *naive greedy methods* typically providing a $(1-1/e)$ guarantee. In contrast, our method addresses a continuous optimization problem on the PF, using gradient-based techniques to solve the established optimization problem to improve both accuracy and efficiency.

2.3 Preference adjustment methods in the decomposition-based MOO paradigm

Since the proposed method can also be classified under the preference adjustment category, we discuss the relationship between the proposed UMOD and preference/weight[2] adjustment methods. The study of preference adjustment methods start from MOEA/D-AWA [40], where its strategy is to remove the preference corresponding to the most crowded objective and add a preference corresponding to the most sparse one. Subsequently, several preference adjustment methods have been introduced [35], including DEA-GNG [34] and MOEA/D-SOM [22], which utilize neural gas networks to guide the selection of preference vectors. W-MOEA/D [21], tw-MOEA/D [38], paλ-MOEA/D [48], and MOEA/D-AWG [55] use mathematical models to shape the non-dominated solutions and adjust preference vectors. The proposed method differs from other preference adjustment methods in two ways: (1) it models the PF with a neural network for better accuracy and efficiency, and (2) it offers a rigorous theoretical analysis for selecting preference vectors yielding uniformity and representativeness.

3 Pareto solutions with uniform designs

3.1 FD as an upper bound of IGD

We first define *fill distance* (FD) [11] of a set $\mathbb{Y}$ ($\mathbb{Y} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(K)}]$) and establish its relationship with the inverted generational distance (IGD) indicator [49] of a set $\mathbb{Y}$, a famous metric in MOO. Formally, FD and IGD are defined as follows:

Definition 1 (FD & IGD)**.**

$$\mathrm{FD}(\mathbb{Y}) = \max_{\boldsymbol{y} \in \mathcal{T}} \min_{\boldsymbol{y}' \in \mathbb{Y}} \rho(\boldsymbol{y}, \boldsymbol{y}') = \max_{\boldsymbol{y} \in \mathcal{T}} \mathrm{dist}(\boldsymbol{y}, \mathbb{Y}), \qquad \mathrm{IGD}(\mathbb{Y}) = \int_{\mathcal{T}} \min_{\boldsymbol{y}' \in \mathbb{Y}} \rho(\boldsymbol{y}, \boldsymbol{y}') d\boldsymbol{y}, \tag{2}$$

where $\rho(\cdot, \cdot)$ represents the Euclidean distance. The term $\min_{\boldsymbol{y}' \in \mathbb{Y}} \rho(\boldsymbol{y}, \boldsymbol{y}')$ represents the nearest distance from a point $\boldsymbol{y}$ on the PF to the reference set $\mathbb{Y}$. Therefore, $\mathrm{FD}(\mathbb{Y}) = \max_{\boldsymbol{y} \in \mathcal{T}} \min_{\boldsymbol{y}' \in \mathbb{Y}} \rho(\boldsymbol{y}, \boldsymbol{y}')$ denotes the *covering radius* of $\mathbb{Y}$, i.e., the largest radius within which at least one solution in $\mathbb{Y}$ covers the entire PF. However, $\mathrm{IGD}(\mathbb{Y})$, which represents the average distance from a point on the PF to the set $\mathbb{Y}$, lacks the clear geometric interpretation that fill distance offers. For MOO, the goal is to minimize a set of Pareto objectives, i.e., $\mathbb{Y} \subset \mathcal{T}$, by optimizing either the FD or IGD indicator: $\min_{\mathbb{Y} \subset \mathcal{T}} \mathrm{FD}(\mathbb{Y})$ or $\min_{\mathbb{Y} \subset \mathcal{T}} \mathrm{IGD}(\mathbb{Y})$ to reach a diverse distribution. Let the optimal sets be $\mathbb{Y}^{\mathrm{FD}}$ and $\mathbb{Y}^{\mathrm{IGD}}$ respectively. The following theorem compares FD and IGD.

Theorem 2 (FD as an upper bound of IGD)**.**

$$\mathrm{IGD}(\mathbb{Y}^{\mathrm{IGD}}) \leq \mathrm{IGD}(\mathbb{Y}^{\mathrm{FD}}) \leq \mathrm{FD}(\mathbb{Y}^{\mathrm{FD}}) \tag{3}$$

The first inequality holds by definition since $\mathbb{Y}^{\mathrm{IGD}}$ minimizes IGD, while the second inequality holds because, for a fixed configuration $\mathbb{Y}^{\mathrm{FD}}$, the average distance ($\mathrm{IGD}(\mathbb{Y}^{\mathrm{FD}})$) is always less than or equal to the maximum distance ($\mathrm{FD}(\mathbb{Y}^{\mathrm{FD}})$). Theorem 2 establishes that the optimal configuration of FD

---

[2] In this paper, “preference vector” and “weight vector” are used interchangeably.

($\mathbb{Y}^{\mathrm{FD}}$) provides an upper bound for the IGD value. To the best of our knowledge, the optimal IGD configuration does not provide a similar upper bound for FD, which is why we focus on FD in this paper.

3.2 Max-packing design as a surrogate of FD design

The minimization of FD involves solving the following nested $\min - \max - \min$ problem:

$$d^{\mathrm{FD}} = \min_{\mathbb{Y} \subset \mathcal{T}} \max_{\boldsymbol{y} \in \mathcal{T}} \min_{\boldsymbol{y}' \in \mathbb{Y}} \rho(\boldsymbol{y}, \boldsymbol{y}'). \tag{4}$$

A small d^{FD} indicates that the *optimal* configuration $\mathbb{Y}^{\mathrm{FD}}$ can well cover the entire PF with a relatively low covering radius. However this triply nested structure is known difficult to optimize [54, 39]. Thus, we plan to seek a max-packing problem which its optimal result serves as a good surrogate result of Equation (4).

$$d^{\mathrm{Pack}} = \max_{\mathbb{Y} \subset \mathcal{T}} \delta = \max_{\mathbb{Y} \subset \mathcal{T}} \left(\min_{1 \le i < j \le K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) \right), \tag{5}$$

where δ represents the *separation distance* between two vectors. The optimal design $\mathbb{Y}^{\mathrm{Pack}}$, which solves the optimization problem (Equation (5)), is known as the max-packing design [8]. In this paper, we show that $\mathbb{Y}^{\mathrm{Pack}}$ effectively optimizes FD *when the decision space is a PF*, as d^{FD} is bounded by $\mathbb{Y}^{\mathrm{Pack}}$ up to a constant factor, independent of size K in Theorem 3.

Theorem 3 (Surrogate for minimal FD)**.** *Consider a connected, compact* [3] *PF $\mathcal{T}$. The minimal fill distance d^{FD} between $\mathcal{T}$ and a size-K design $\mathbb{Y}^{\mathrm{Pack}}$/$\mathbb{Y}^{\mathrm{FD}}$ will then be bounded as:*

$$\frac{1}{4} d^{\mathrm{Pack}} \le d^{\mathrm{FD}} \le \mathrm{FD}(\mathbb{Y}^{\mathrm{Pack}}) \le d^{\mathrm{Pack}}, \tag{6}$$

Furthermore, the fill distance between $\mathcal{T}$ and the optimal design d^{Pack} induced by $\mathbb{Y}^{\mathrm{Pack}}$ is upper bounded by d^{Pack}, which is guaranteed to be upper bounded by $4d^{\mathrm{FD}}$. $\mathbb{Y}^{\mathrm{Pack}}$ is thus considered a quality, rate-optimal representative for the whole PF $\mathcal{T}$.

The second inequality $d^{\mathrm{FD}} \le d^{\mathrm{Pack}}$ is proved by Auffray et al. [3], Pronzato [39], and we provide a tighter lower bound $d^{\mathrm{Pack}}/4$, utilizing the topological property of a PF. The complete proof of Theorem 3 is left in Appendix A.1. Furthermore, under an additional strict inequality condition $d_K^{\mathrm{Pack}} < d_{K+1}^{\mathrm{Pack}}$ on the PF, the max-packing design $\mathbb{Y}_K^{\mathrm{Pack}}$ serves as a d_K^{Pack}-covering of $\mathcal{T}$, which is established by Theorem 4. The subscript "K" specifically denotes the max-packing distance d_K^{Pack} for a size-K design, similarly used for $\mathbb{Y}_K^{\mathrm{Pack}}$ to represent a size-K design.

Theorem 4. *Consider a connected, compact PF ($\mathcal{T}$), with the property $d_K^{\mathrm{Pack}} < d_{K+1}^{\mathrm{Pack}}$, the max-packing design $\mathbb{Y}_K^{\mathrm{Pack}}$ covers $\mathcal{T}$ with radius of at most d_K^{Pack}.*

This theorem suggests that $\mathbb{Y}_K^{\mathrm{Pack}}$ can represent $\mathcal{T}$ well since the maximal distance between any vector $\boldsymbol{y} \in \mathcal{T}$ and $\mathbb{Y}_K^{\mathrm{Pack}}$ is bounded by d_K^{Pack}.

3.3 Characterizations of a max-packing design

This section discusses key properties of the max-packing design on a PF. For bi-objective problems, $\mathbb{Y}^{\mathrm{Pack}}$ shows favorable properties when $\mathbb{Y}^{\mathrm{Pack}} \subset \mathcal{T}$, as formalized in Theorem 5, with the proof in Appendix A.2.

Theorem 5 (Characterization of $\mathbb{Y}^{\mathrm{Pack}}$ for biobjective problems)**.** *Let $\mathbb{Y}^{\mathrm{Pack}} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(K)}]$ be sorted by increasing first component, such that $y_1^{(1)} \le \ldots \le y_1^{(K)}$. For a compact, connected $\mathcal{T}$, $\mathbb{Y}^{\mathrm{Pack}}$ is characterized as follows:*

1. *Equal spacing: $\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}) = \ldots = \rho(\boldsymbol{y}^{(K-1)}, \boldsymbol{y}^{(K)})$, for $K \ge 3$.*
2. *Endpoint alignment: $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(K)}$ are two endpoints ($\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}$) of $\mathcal{T}$, i.e., $\boldsymbol{y}^{(1)} = \boldsymbol{p}^{(1)} = [\inf_{\boldsymbol{y} \in \mathcal{T}} y_1, \sup_{\boldsymbol{y} \in \mathcal{T}} y_2]$ and $\boldsymbol{y}^{(K)} = \boldsymbol{p}^{(2)} = [\sup_{\boldsymbol{y} \in \mathcal{T}} y_1, \inf_{\boldsymbol{y} \in \mathcal{T}} y_2]$, for $K \ge 2$.*

---

[3] A connected, compact set is also called a *rectifiable* set.

Remark. Firstly, $\mathbb{Y}^{\text{Pack}}$, including both the starting and ending points, spans the maximum range among all configurations, which is desirable. Secondly, equal pairwise distances between Pareto objectives yields an intuitive interpretation of uniformity. Maximizing hypervolume ensures the "equal spacing" property only for bi-objective *linear* PFs [4][Theorem 4], while our design only requires the PF to be *compact and connected*.

Besides this non-asymptotical bi-objective results, we examine the asymptotic properties of $\mathbb{Y}^{\text{Pack}}$ by Theorem 6.

Theorem 6 (Asymptotic uniformity [8](Theorem. 2.1))**.** *As the number of set size $K \to \infty$, the set sequence $\{\mathbb{Y}_K^{\text{Pack}}\}$ weakly converge to a uniform distribution over a compact, connected $\mathcal{T}$. Specifically, for any subset $\mathcal{B} \subset \mathcal{T}$ with measure-zero boundary, the proportion of points in $\mathbb{Y}_K^{\text{Pack}}$ lying within $\mathcal{B}$ converges to the proportion of $\mathcal{T}$ occupied by $\mathcal{B}$:*

$$\lim_{K \to \infty} \frac{\#(\mathbb{Y}_K^{\text{Pack}} \cap \mathcal{B})}{\#(\mathbb{Y}_K^{\text{Pack}})} = \frac{\text{Vol}(\mathcal{B})}{\text{Vol}(\mathcal{T})} = \mathbb{P}(\boldsymbol{y} \in \mathcal{B} \mid \boldsymbol{y} \in \mathcal{T}), \tag{7}$$

where $\#$ denotes the number of points in a set, and "Vol" denotes the volume of a set.

Theorem 6 shows that as the solution set size K grows, $\mathbb{Y}_K^{\text{Pack}}$ approaches a uniform distribution. Specifically, random variable $\boldsymbol{Y}_K^{\text{Pack}} \xrightarrow{d} \text{Unif}(\mathcal{T})$, meaning $\boldsymbol{Y}_K^{\text{Pack}}$, the categorical distribution where each $\boldsymbol{y} \in \mathbb{Y}_K^{\text{Pack}}$ has probability $1/K$, converges in distribution to $\text{Unif}(\mathcal{T})$.

4 Efficient optimization of a size-K uniform set

The original max-packing problem (Equation (5)) maximizes the minimal pairwise distance among Pareto objectives and can be reformulated as the following constrained optimization problem on the PF:

$$\max \left(\min_{1 \le i < j \le K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) \right) \qquad \text{s.t. } \boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)} \in \mathcal{T}. \tag{8}$$

To constrain $\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}$ pairs as Pareto objectives, we solve decision variables of $\boldsymbol{y}^{(i)}$'s as the optimal solution of the following modified Tchebycheff (mTche) aggregation function (Equation (9)),

$$\boldsymbol{y} = \tilde{\boldsymbol{h}}(\boldsymbol{\lambda}) = \arg \min_{\boldsymbol{y}' \in \mathcal{Y}} \left\{ \frac{y_i - z_i}{\lambda_i} \right\} : \qquad \left[0, \frac{\pi}{2}\right]^{m-1} \mapsto \mathbb{R}^m, \tag{9}$$

where $\boldsymbol{z}$ is a reference point ($z_i \le y_i, \forall \boldsymbol{y} \in \mathcal{Y}, \forall i \in [m]$). This substitution is equivalent *when the optimal solution of Equation* (9) *is unique*, as for any Pareto objective $\boldsymbol{y}$, there exists a preference $\boldsymbol{\lambda} \in \Delta_m$ such that the optimal value of Equation (9) matches $\boldsymbol{y}$. This argument is explained in Appendix A.4. Substituting Equation (9) into Equation (8) yields the following bi-level optimization problems:

$$\begin{cases} d^{\text{Pack}} = \max\limits_{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}} \min\limits_{1 \le i < j \le K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) \\ \boldsymbol{y}^{(k)} = \arg \min\limits_{\boldsymbol{y}^{(k)} \in \mathcal{Y}} \left\{ \dfrac{y_i^{(k)} - z_i}{\lambda_i(\boldsymbol{\vartheta}^{(k)})} \right\}, \ i \in [m]. \end{cases} \Rightarrow \begin{cases} d^{\text{Pack}} = \max\limits_{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}} \min\limits_{1 \le i < j \le K} \rho(\boldsymbol{h}(\boldsymbol{\vartheta}^{(i)}), \boldsymbol{h}(\boldsymbol{\vartheta}^{(j)})) \\ \text{s.t.} \quad \boldsymbol{\vartheta}^{(k)} \in \left[0, \dfrac{\pi}{2}\right]^{m-1}. \end{cases} \tag{10}$$

In addition to solving the mTche aggregation function, it can also use the recently developed PMGDA [59] to achieve the exact Pareto optimal solution. The advantage is that by selecting an appropriate control parameter σ, PMGDA converges faster to the exact Pareto solution. The function $\boldsymbol{\lambda}(\boldsymbol{\vartheta})$ converts a "preference angle" from an angle space $\left[0, \frac{\pi}{2}\right]^{m-1}$ into a preference vector. The conversion relationship is detailed in Appendix C.3. We use $\boldsymbol{\lambda}(\boldsymbol{\vartheta})$ as decision variables for easier optimization since $\boldsymbol{\lambda}(\boldsymbol{\vartheta})$ is constrained in a box. In the right equation, the Pareto objective is denoted as $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{\vartheta}) = \tilde{\boldsymbol{h}}(\boldsymbol{\lambda}(\boldsymbol{\vartheta}))$. Various bi-level optimization methods [47, 60] can be used to solve problem (Equation (10)). For efficiency consideration, we use a gradient-based approach. Define $\delta = \min_{(i,j)} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$, where (i^*, j^*) is the optimal pair from $\arg\min_{(i,j)} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$. After some basic algebraic calculations, $\frac{\partial \delta}{\partial \boldsymbol{\vartheta}}$ can be calculated by the following two equations:

$$\frac{\partial \delta}{\partial \boldsymbol{\vartheta}^{(i^*)}} = \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{\vartheta}^{(i^*)})}{\partial \boldsymbol{\vartheta}^{(i^*)}}}_{\boldsymbol{B}(n\times m)} \underbrace{\frac{\boldsymbol{h}(\boldsymbol{\vartheta}^{(i^*)}) - \boldsymbol{h}(\boldsymbol{\vartheta}^{(j^*)})}{\rho(\boldsymbol{h}(\boldsymbol{\vartheta}^{(i^*)}), \boldsymbol{h}(\boldsymbol{\vartheta}^{(j^*)}))}}_{\boldsymbol{A}(m\times 1)}^{\top}, \quad \frac{\partial \delta}{\partial \boldsymbol{\vartheta}^{(j^*)}} = -\underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{\vartheta}^{(j^*)})}{\partial \boldsymbol{\vartheta}^{(j^*)}}}_{\boldsymbol{C}(n\times m)} \underbrace{\frac{\boldsymbol{h}(\boldsymbol{\vartheta}^{(i^*)})\boldsymbol{h}(\boldsymbol{\vartheta}^{(j^*)})}{\rho(\boldsymbol{h}(\boldsymbol{\vartheta}^{(i^*)}), \boldsymbol{h}(\boldsymbol{\vartheta}^{(j^*)}))}}_{\boldsymbol{A}(m\times 1)}^{\top},$$

The remaining terms $\frac{\partial \delta}{\partial \boldsymbol{\vartheta}^{(k)}}, k \neq i^*, j^*$ are zero vectors. Calculating the gradient vector $\boldsymbol{A}$ is straightforward by using some basic calculations. However, computing the gradient matrices $\boldsymbol{B}$ or $\boldsymbol{C}$ is difficult because $\boldsymbol{h}(\boldsymbol{\vartheta})$ is a black-box function, representing the optimal values of an optimization problem. Estimating matrices $\boldsymbol{B}$ or $\boldsymbol{C}$ using finite difference is impractical, as it requires solving Equation (9) for at least $n \times m$ times, which is highly time-consuming. To efficiently estimate $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\vartheta}}$, we propose using a neural model $\boldsymbol{h}_\phi$ to approximate $\boldsymbol{h}$ based on historical data $(\boldsymbol{\vartheta}, \boldsymbol{h}(\boldsymbol{\vartheta}))$, allowing us to estimate $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\vartheta}}$ via $\frac{\partial \boldsymbol{h}_\phi}{\partial \boldsymbol{\vartheta}}$. Since a neural network is introduced, we analyze its induced error in Theorem 7, with the full proof in Appendix A.3.

Theorem 7 (Optimization error ϵ_{nn} introduced by using a network)**.** *Let $\boldsymbol{h}_\phi(\boldsymbol{\vartheta})$ be an approximator of $\boldsymbol{h}(\boldsymbol{\vartheta})$ such that $||\boldsymbol{h}_\phi(\boldsymbol{\vartheta}) - \boldsymbol{h}(\boldsymbol{\vartheta})|| \leq \epsilon$, for every $\boldsymbol{\vartheta} \in [0, \frac{\pi}{2}]^{m-1}$, as commonly assumed in bi-level optimization, e.g., [18], Eq. (10). ϵ_{nn} is the difference between the maximum of the minimal distances calculated using the approximate function $\boldsymbol{h}_\phi$ and the true function $\boldsymbol{h}$. Then, ϵ_{nn} is bounded by 2ϵ:*

$$\epsilon_{\text{nn}} = \left| \max_{\boldsymbol{\vartheta}^{(1)},\ldots,\boldsymbol{\vartheta}^{(K)}} \min_{1\leq i<j\leq K} \rho(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}), \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)})) - d^{\text{Pack}} \right| \leq 2\epsilon.$$

Remark. The error ϵ, defined as $||\boldsymbol{h}_\phi(\boldsymbol{\vartheta}) - \boldsymbol{h}(\boldsymbol{\vartheta})|| \leq \epsilon$, is both influenced by the covering radius R of the estimated solutions $\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(1)}), \ldots, \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(K)})$ and the fitting error ϵ_{fit}, where $\epsilon_{\text{fit}} = \max_{k\in[K]} ||\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(k)}) - \boldsymbol{h}(\boldsymbol{\vartheta}^{(k)})||$. For any $\boldsymbol{\vartheta}$, the error satisfies $||\boldsymbol{h}_\phi(\boldsymbol{\vartheta}) - \boldsymbol{h}(\boldsymbol{\vartheta})|| \leq L||\boldsymbol{\vartheta} - \boldsymbol{\vartheta}^{(i')}|| + \epsilon_{\text{fit}} \leq L \cdot R + \epsilon_{\text{fit}}$, where L is the Lipschitz constant of function $(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}) - \boldsymbol{h}(\boldsymbol{\vartheta}))$, $\boldsymbol{\vartheta}^{(i')}$ is the nearest solution to $\boldsymbol{\vartheta}$ among the estimated solutions, and R is the covering radius, which can be further reduced by adding more training pairs. For overparameterized networks, ϵ_{fit} can be considered as very small.

Practical algorithms. Due to the space limit, the pseudo-codes for UMOD are provided in Algorithm 1 and Algorithm 2 in Appendix C.1. Initially, K uniform preferences are generated by the Das-Dennis algorithm [12]. Then either a decomposition-based multiobjective evolutionary algorithm or a gradient-based MOO with mTche aggregation function is employed for producing preference angle and Pareto objective pairs $(\boldsymbol{\vartheta}, \boldsymbol{y})$. Evolutionary algorithms are preferred for problems with multiple local optimas, while gradient-based MOO methods are preferred for multiobjective machine learning (MOML) problems. Given $(\boldsymbol{\vartheta}, \boldsymbol{y})$ pairs, a PF model $\boldsymbol{h}_\phi(\boldsymbol{\vartheta})$ is fitted by optimizing the mean square estimation error. Finally, preference angles are updated to maximize the minimal pairwise distances of Pareto objectives. These steps are repeated for N iterations until convergence.

5 Experiments

The experiments compare UMOD with other methods on two types of MOPs: (1) those with many local optimas which can be solved by MOEAs efficiently, and (2) multiobjective fairness classification neural networks as decision variables. MOEAs and multiobjective fairness problems are executed with 31/5 random seeds, separately. We employ seven indicators to assess performance of different algorithms: **(1) Hypervolume** (↑) [24], **(2) IGD** (↓) [26], **(3) Sparsity** (↓) [56], **(4) Spacing** (↓) [43], (5) **Uniformity** (↑), (6) **Smooth Uniformity** (↑), and (7) **Fill distance** (↓). Indicators 1-2 focus on convergence and diversity for multi-objective optimization (MOO), while indicators 3-7 evaluate solution uniformity. See Appendix B.1 for more detailed expression for these indicators.

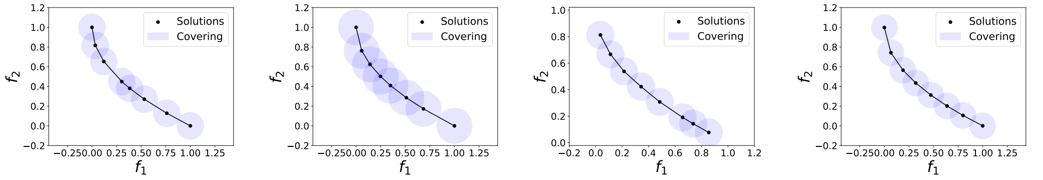
5.1 Comparison with MOEAs

We demonstrate the effectiveness of our proposed method across a diverse set of MOEA benchmark problems, including the ZDT[4] [16], DTLZ [15][5], and real-world testing problems [50]. Real-world

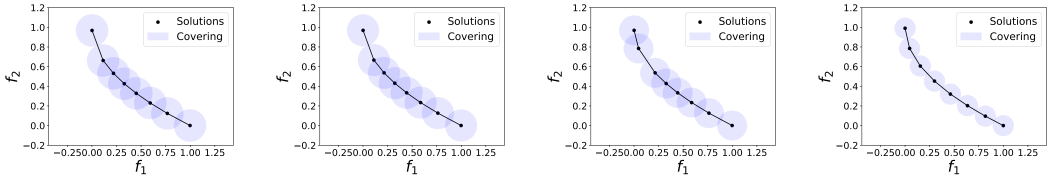
---

[4] ZDT5 is a discrete optimization problem and is commonly excluded in MOEA studies.

[5] DTLZ 7 is excluded because our theoretical results only apply to compact and connected PFs, which DTLZ 7 does not satisfy. Results on DTLZ 5-6 are demonstrated in Appendix B.6.

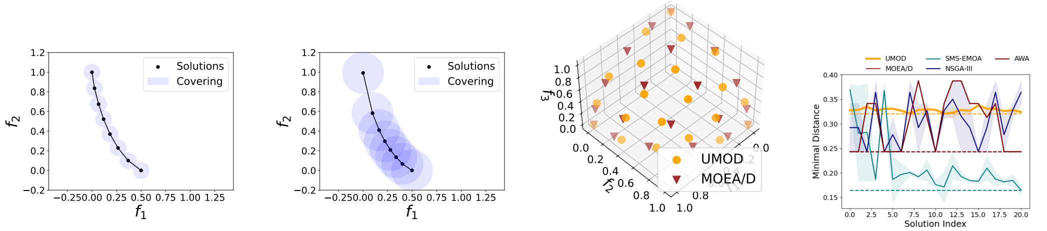


(a) LMPFE (b) DEAGNG (c) Subset selection (d) SMS-EMOA



(e) NSGA3 (f) MOEA/D (g) MOEA/D-AWA (h) UMOD

Figure 3: Result comparison by different methods on ZDT1.



(a) UMOD on RE21 (b) MOEA/D on RE21 (c) UMOD and MOEA/D (d) Minimal distances to other objectives

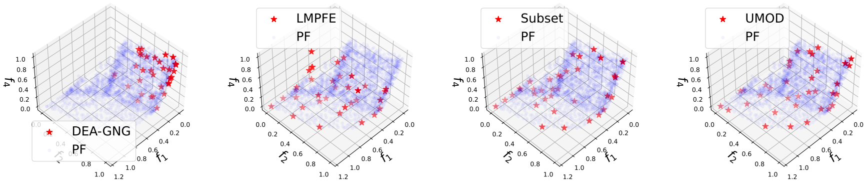
Figure 4: Results on RE21 and DTLZ2.

testing problems include: four-bar truss design (RE21), reinforced concrete beam design (RE22), disc brake design (RE33), rocket injector design (RE37), car side impact design (RE41), and conceptual marine design (RE42). Notably, RE41 and RE42 are complex four-objective problems that involve extensive four-dimensional objective spaces. The prefix "RE" denotes real-world scenarios.

Table 1: Partial results for biobjective problems (full results are in Table 7).

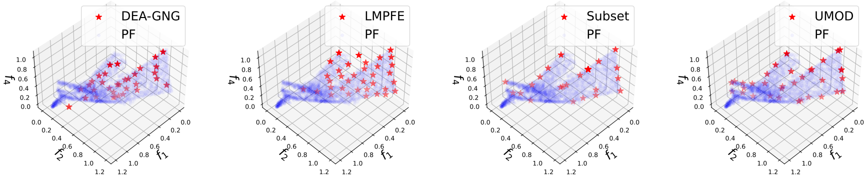
	Indicator	DEA-GNG [34]	LMPFE [51]	Subset [9, 45]	NSGA3 [13, 27]	SMS-EMOA [6]	MOEA/D [58]	MOEA/D-AWA [40]	UMOD
ZDT1	HV	1.03 (0.00) (6)	1.03 (0.00) (2)	1.02 (0.00) (7)	1.03 (0.00) (5)	1.04 (0.00) (1)	1.03 (0.00) (4)	1.03 (0.00) (3)	**1.04 (0.00) (0)**
	IGD	5.68 (0.29) (7)	5.42 (0.21) (6)	5.21 (0.40) (1)	5.28 (0.00) (3)	5.22 (0.07) (2)	5.30 (0.00) (4)	5.42 (0.02) (5)	**5.19 (0.00) (0)**
	Spacing	6.35 (1.74) (7)	2.44 (1.82) (2)	3.51 (0.47) (3)	5.90 (0.00) (5)	1.91 (0.46) (1)	5.92 (0.01) (6)	3.69 (0.04) (4)	**0.12 (0.06) (0)**
	Sparsity	4.80 (0.19) (7)	4.55 (0.12) (4)	**2.59 (0.18) (0)**	4.60 (0.00) (5)	4.48 (0.01) (2)	4.61 (0.00) (6)	4.51 (0.01) (3)	4.39 (0.00) (1)
	Uniform	1.33 (0.11) (6)	1.67 (0.32) (2)	0.89 (0.10) (7)	1.53 (0.00) (3)	1.85 (0.03) (1)	1.51 (0.00) (4)	1.51 (0.00) (5)	**2.07 (0.01) (0)**
	SUniform	0.47 (0.12) (6)	0.68 (0.13) (2)	0.09 (0.09) (7)	0.49 (0.00) (4)	0.74 (0.01) (1)	0.48 (0.00) (5)	0.53 (0.00) (3)	**0.77 (0.00) (0)**
	Fill Distance	1.60 (0.23) (6)	1.23 (0.09) (2)	2.00 (0.18) (7)	1.55 (0.00) (5)	1.16 (0.05) (1)	1.50 (0.00) (4)	1.43 (0.03) (3)	**1.04 (0.01) (0)**
RE21	HV	1.24 (0.00) (5)	1.23 (0.01) (7)	1.24 (0.00) (4)	1.24 (0.00) (2)	1.24 (0.00) (1)	1.24 (0.00) (6)	1.24 (0.00) (3)	**1.24 (0.00) (0)**
	IGD	4.63 (0.28) (5)	4.61 (0.13) (4)	5.15 (0.01) (6)	4.44 (0.00) (3)	4.23 (0.02) (1)	5.40 (0.02) (7)	4.33 (0.04) (2)	**4.12 (0.00) (0)**
	Spacing	6.63 (1.16) (6)	3.62 (0.93) (4)	1.38 (0.00) (1)	5.71 (0.01) (5)	3.19 (0.37) (2)	10.49 (0.04) (7)	3.47 (0.41) (3)	**0.12 (0.05) (0)**
	Sparsity	3.10 (0.21) (6)	2.90 (0.10) (4)	**1.74 (0.00) (0)**	3.02 (0.00) (5)	2.82 (0.02) (2)	3.87 (0.02) (7)	2.84 (0.01) (3)	2.70 (0.00) (1)
	Uniform	0.81 (0.12) (7)	0.95 (0.16) (5)	0.95 (0.00) (4)	1.16 (0.00) (2)	1.26 (0.05) (1)	0.93 (0.00) (6)	1.11 (0.03) (3)	**1.62 (0.01) (0)**
	SUniform	-0.03 (0.07) (5)	0.12 (0.08) (3)	-0.17 (0.00) (7)	0.08 (0.00) (4)	0.20 (0.01) (1)	-0.17 (0.00) (6)	0.13 (0.01) (2)	**0.31 (0.00) (0)**
	Fill Distance	1.45 (0.20) (4)	1.21 (0.12) (3)	2.62 (0.01) (7)	1.47 (0.00) (5)	1.09 (0.04) (1)	2.11 (0.01) (6)	1.15 (0.02) (2)	**0.83 (0.00) (0)**
Rank	HV	5.14	4.14	5.57	3.86	**0.57**	3.86	3.57	1.29
	IGD	5	4.86	4	2.43	2.29	4.43	4	**1**
	Spacing	5.71	2.71	3.86	3.86	3	5.14	3.57	**0.14**
	Sparsity	5.29	5.71	1.43	3.71	2.71	4.86	3.43	**0.86**
	Uniform	5	3.86	6.43	3	1.71	4.29	3.57	**0.14**
	SUniform	5.14	3.57	6.86	3.14	2	4	3.14	**0.14**
	Fill Distance	5.43	3.57	5.57	3.43	2.29	3.57	3.29	**0.86**

Multiobjective testing problems often have many local Pareto optimal solutions, leading to comparisons primarily with MOEAs. Baseline MOEAs include: *MOEA/D* [58], a decomposition-based



(a) DEAGNG (b) LMPFE (c) Subset selection (d) UMOD

Figure 5: Results on RE41 by different methods (full results are in Figure 8).



(a) DEAGNG (b) LMPFE on RE41 (c) Subset selection (d) UMOD

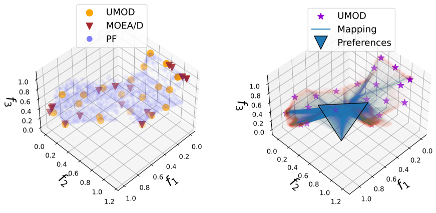
Figure 6: Results on RE42 by different methods (full results are in Figure 9).

approach; (2) *MOEA/D-AWA* [40], integrating adaptive weight adjustment; (3) *NSGA3* [13, 27], generating diverse Pareto objectives through crowding distance; (4) *SMS-EMOA* [6], maximizing hypervolume for diverse solutions; (5) *LMPFE* [51], estimating the PF using multiple local models; (6) *Subset selection* [9, 45], choosing a solution set by hypervolume maximization[6]; and (7) *DEA-GNG* [34], a preference adjustment method based on growing neural gas network. Methods (1)-(4) are classical MOEA methods implemented by Pymoo [7], while methods (5)-(7) are more recent methods. Full name of these methods are provided in Table 4.

Table 2: Partial results on three-objective problems (full results are in Table 8).

	Indicator	DEA-GNG [34]	LMPFE [51]	Subset [45]	NSGA3 [13]	SMS-EMOA [6]	MOEA/D [58]	MOEA/D-AWA [40]	UMOD
DTLZ2	HV	1.01 (0.01) (7)	1.05 (0.00) (6)	1.08 (0.00) (1)	1.06 (0.00) (5)	**1.08 (0.00) (0)**	1.06 (0.00) (3)	1.06 (0.00) (4)	1.07 (0.00) (2)
	IGD	12.52 (0.64) (2)	12.46 (0.09) (1)	15.14 (0.00) (6)	12.54 (0.00) (3)	15.45 (0.25) (7)	12.55 (0.00) (5)	12.55 (0.00) (4)	**12.19 (0.04) (0)**
	Spacing	5.29 (1.72) (2)	2.64 (0.39) (1)	8.97 (0.00) (7)	5.45 (0.00) (3)	7.15 (0.51) (6)	5.45 (0.01) (4)	5.45 (0.01) (5)	**0.52 (0.21) (0)**
	Sparsity	**1.50 (0.13) (0)**	2.18 (0.13) (2)	2.42 (0.00) (3)	2.43 (0.00) (4)	2.75 (0.14) (7)	2.43 (0.00) (6)	2.43 (0.00) (5)	1.61 (0.13) (1)
	Uniform	1.58 (0.19) (5)	2.45 (0.13) (1)	0.96 (0.00) (7)	2.43 (0.00) (4)	1.51 (0.15) (6)	2.43 (0.00) (3)	2.43 (0.00) (2)	**3.17 (0.09) (0)**
	SUniform	0.36 (0.23) (5)	0.97 (0.04) (1)	-0.04 (0.00) (7)	0.95 (0.00) (4)	0.18 (0.07) (6)	0.95 (0.00) (2)	0.95 (0.00) (3)	**1.18 (0.02) (0)**
	Fill Distance	3.20 (0.60) (5)	2.69 (0.10) (4)	3.54 (0.00) (7)	2.59 (0.00) (1)	3.42 (0.16) (6)	2.59 (0.00) (3)	2.59 (0.00) (2)	**2.37 (0.08) (0)**
Rank	HV	6.8	4.6	1.8	4.4	**0.2**	3.6	4.2	2.4
	IGD	6	3	4.6	1.8	4.6	3.4	4	**0.6**
	Spacing	4.4	2.8	5.6	3.6	4	3.4	3.6	**0.6**
	Sparsity	**1.4**	3.6	1.6	4.6	4.8	5.6	4.6	1.8
	Uniform	5.8	2	5.8	2.8	4	3.4	3.4	**0.8**
	SUniform	6	2	5.8	3.4	4.4	3	3.2	**0.2**
	Fill Distance	6.6	4.2	5.2	1	4.2	3.2	3.2	**0.4**

We present the results for *biobjective* problems in Figure 3 and Table 1[7]. By directly minimizing maximal pairwise distances, the uniform indicator (which corresponds to maximal pairwise distances, see Appendix B.1, metric 5) is optimized effectively and ranks best among all methods. The fill distance, a surrogate for maximal pairwise distance up to constant, also performs best among all methods. This indicates that solutions found by UMOD cover the true PF with the minimal covering radius among all methods. Figure 3 further confirms that the covering radius of UMOD is significantly smaller than other methods. We also observe



(a) UMOD and MOEA/D on RE37 (b) Visualization on RE37

Figure 2: (a): UMOD solutions are more uniform. (b): A PF model can be trained with a small number of solutions.

---

[6] Code: `https://github.com/HisaoLabSUSTC/GAHSS`.

[7] For a better illustration, the unit of IGD, Spacing, Sparsity, Uniform, SUniform, and Fill distance are scaled by 0.01,0.01,0.01,0.1,0.1, and 0.1.

Table 3: Statistical results on fairness classification problems. Results are averaged on five random seeds. Results are presented in the the format of “(mean)/(std)”.

	Indicator	UMOD	Agg-mTche [37]	Agg-LS [37]	Agg-PBI [58]	EPO [36]	PMGDA [59]	HVGrad [17]
Adult	$\delta_{\mathrm{Unif}}(\uparrow)$	**0.1049** (0.0000)	0.0151 (0.0000)	0.0006 (0.0000)	0.0232 (0.0000)	0.0161 (0.0000)	0.0201 (0.0000)	0.0421 (0.0000)
	$\delta_{\mathrm{SUnif}}(\uparrow)$	**0.0088** (0.0000)	-0.0849 (0.0000)	-0.1482 (0.0000)	-0.0764 (0.0000)	-0.0838 (0.0000)	-0.0738 (0.0000)	-0.0510 (0.0000)
	HV(↑)	0.6800 (0.0000)	0.6769 (0.0000)	0.6422 (0.0000)	0.6679 (0.0000)	0.6755 (0.0000)	0.6764 (0.0000)	**0.6865** (0.0000)
	Span(↑)	**0.0773** (0.0000)	0.0615 (0.0000)	0.0012 (0.0000)	0.0735 (0.0000)	0.0594 (0.0000)	0.0688 (0.0000)	0.0766 (0.0000)
	Spacing(↓)	0.8677 (0.0979)	2.9454 (0.0194)	**0.0199** (0.0000)	2.3149 (0.0105)	2.9316 (0.0055)	5.6512 (0.0058)	1.9228 (0.0146)
	Sparsity(↓)	1.3914 (0.0057)	0.3067 (0.0004)	**0.0002** (0.0000)	0.3013 (0.0001)	0.3124 (0.0002)	0.8369 (0.0003)	0.4888 (0.0002)
Compass	$\delta_{\mathrm{Unif}}(\uparrow)$	**0.3801** (0.0027)	0.0131 (0.0000)	0.0014 (0.0000)	0.0203 (0.0000)	0.0191 (0.0000)	0.0176 (0.0000)	0.2199 (0.0000)
	$\delta_{\mathrm{SUnif}}(\uparrow)$	**0.3148** (0.0014)	-0.0429 (0.0000)	-0.1126 (0.0000)	-0.0332 (0.0000)	-0.0358 (0.0000)	-0.0360 (0.0000)	0.1635 (0.0000)
	HV(↑)	0.7262 (0.0012)	0.7394 (0.0000)	0.7256 (0.0000)	0.7359 (0.0000)	0.7448 (0.0000)	0.7428 (0.0000)	**0.7625** (0.0000)
	Span(↑)	**0.2156** (0.0006)	0.1950 (0.0000)	0.1783 (0.0000)	0.1954 (0.0000)	0.1979 (0.0000)	0.1986 (0.0000)	0.2089 (0.0000)
	Spacing(↓)	**4.4137** (6.1916)	25.4158 (0.1208)	27.1222 (0.0185)	22.8444 (0.2152)	25.7459 (0.2039)	26.4816 (0.1548)	7.6151 (0.0198)
	Sparsity(↓)	19.7485 (1.5139)	12.5492 (0.0864)	12.4902 (2.1463)	10.8758 (0.0985)	13.1004 (0.1647)	13.8356 (0.0960)	**10.0604** (0.0233)

that the IGD indicator (Appendix B.1, metric 2), representing the mean Euclidean distance between the true PF and the found size-K solution set, is significantly improved. The significant improvement over IGD, a well-established indicator of uniformity and convergence of Pareto solutions, suggests that our method finds high-quality Pareto solutions and also implies an inherent theoretical connection between IGD and minimal pairwise distance maximization, warranting further investigation.

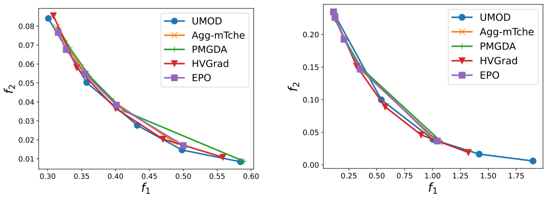
We would like to mention another advantage of UMOD is it can handle PFs of different scales, as demonstrated in Figure 4(a)/(b). Unlike using fixed preference vectors, which can result in non-uniform Pareto objectives, UMOD ensures uniformity in the objective space, remaining the uniformity of the achieved distribution unaffected by the scale of a PF.

For *three-objective* problems, DTLZ1 owns a simplex-like PF, making DTLZ1 the only problem where uniform preferences result in uniform Pareto objectives. For DTLZ2 problem with a sphere-like PF, using uniform preferences on the simplex fails to produce uniform Pareto objectives by MOEA/D. As shown in Figure 4(c), objectives solved by MOEA/D around the center of the PF are sparse, while those around the margin are dense. In contrast, UMOD produces uniform Pareto objectives on the PF. The minimal distances from one Pareto objective to the rest of the solutions, sorted by index, are shown in Figure 4(d), indicating that only UMOD achieves the maximal minimal pairwise distance. The results for the real-world RE37 problem are shown in Figure 2(a), indicating MOEA/D produces inefficient duplicated solutions on the boundary of the PF. In contrast, UMOD effectively reduces duplicated solutions by maximizing the minimal pairwise distance on the PF, as duplicated solutions have a minimal pairwise distance of zero. A visualization of the learned PF is shown in Figure 2(b), indicating that a PF model can be learned efficiently by using only a small number of solutions. Most indicators related with uniformity, such as IGD, uniform distance, and FD outperforms other methods significantly, indicating UMOD finds much more uniform and representative solutions. The HV indicator of UMOD is the 3-rd best and comparable to SMS-EMOA, a method directly optimizes the HV indicator.

Finally, we discuss the results for four-objective problems, shown in Figures 8 and 9 and Table 9. UMOD outperforms DEAGNG and LMPFE by discovering a broader PF (Figure 5), as it directly maximizes the minimal pairwise distance on the PF. Although subset selection performs similarly to UMOD on the RE41 problem, it relies on an inefficient two-phase optimization to select a subset from a larger solution set, making it less efficient for four-objective problems. On RE41, UMOD significantly outperforms other methods in IGD, finding more representative solutions. On RE42, UMOD ranks highly (best or second best) in uniformity indicators like spacing, sparsity, and smooth uniformity. Due to a challenging PF region, IGD indicators for UMOD, NSGA3, and SMS-EMOA are similar, with these three methods outperforming the others.

5.2 Results on fairness classification

In this subsection, we compare our method with other gradient-based MOO methods on multiobjective fairness tasks including Adult [2] and Compass [1]. The decision variable is neural network parameters for both accuracy and fairness. The first objective is data classification accuracy, measured by binary cross-entropy loss, while the second uses a hyperbolic tangent relaxation of the difference of equality of opportunity (DEO) loss [41][Eq. 6]. Details of the data and network architecture are in Table 6 in Appendix B.3.



(a) Adult

(b) Compass

Figure 7: Result on fairness classification.

Results with five solutions are illustrated in Figure 7 compared with PMGDA [59], Hypervolume-gradient method (HVGrad), EPO [36], modified Tchebycheff aggregation function method (Agg-mTche) [58]. For the Adult dataset, all methods run for 30 epochs, and for the Compass dataset, 20 epochs. A detailed description of these methods is in Appendix C.4. The true PF range is often unknown in real-world problems, making it challenging to determine uniform preference vectors for preference-based methods. If these vectors don't include the PF endpoints, important parts of the PF may be missed. In contrast, HV-Grad and the proposed UMOD method do not rely on preferences and can automatically identify a large PF. Figure 7 validates Theorem 5, showing that UMOD identifies both endpoints of the true PF, while HVGrad's endpoints cannot be determined.

Numerical results are presented in Table 3. In real-world problems, objectives often vary in scale, making uniform preference vectors non-equivalent to uniform objective vectors. UMOD, however, is designed to generate uniform objective vectors independent of scale, as evidenced by its superior performance in uniformity and soft uniformity indicators. Additionally, UMOD's spacing indicator, measuring neighborhood distance deviation, is significantly lower than other methods (except Agg-LS, which produces duplicate solutions in the Adult dataset, resulting in the lowest spacing). UMOD's Span indicators outperform other methods a lot, demonstrating its ability to recover a more complete PF, a desirable feature. Many methods show similar HV indicators, but their solution configurations vary, suggesting HV optimization is a coarse measure of uniformity. UMOD's significantly better uniformity performance indicates that directly optimizing uniformity, as in UMOD, is a promising approach for generating diverse Pareto solutions.

6 Conclusions and further works

Conclusions. In this paper, we have proposed a new understanding of a longstanding problem in MOO, generating K uniform and representative Pareto objectives, through searching for a max-packing design on the PF. We provide rigorous analysis of the resulting objective design, and in particular, we show this design will asymptotically converge to the uniform measure over Pareto front. With this new paradigm, we also empirically demonstrate how the space-filling design we obtain can benefit downstream performance with both synthetic and real-world MOO tasks. Overall, we believe that we pave a new way for (rate-)optimally configuring the Pareto objectives.

Future works include applying UMOD to large-scale real-world multiobjective problems, such as material design, vaccine design, and recommendation systems. The broader impacts of this work is discussed in Appendix D.1.

Acknowledge

GH Li offers guidance on evolutionary computation, while YF Chen proves Theorem 3. The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. CityU 11215622].

References

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of data and analytics*, pages 254–264. Auerbach Publications, 2022.

[2] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[3] Yves Auffray, Pierre Barbillon, and Jean-Michel Marin. Maximin design on non hypercube domains and kernel interpolation. *Statistics and Computing*, 22:703–712, 2012.

[4] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102, 2009.

[5] Sterling G Baird, Ramsey Issa, and Taylor D Sparks. Materials science optimization benchmark dataset for multi-objective, multi-fidelity optimization of hard-sphere packing simulations. *Data in Brief*, 50:109487, 2023.

[6] Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[7] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[8] S Borodachov, D Hardin, and E Saff. Asymptotics of best-packing on rectifiable sets. *Proceedings of the American Mathematical Society*, 135(8):2369–2380, 2007.

[9] Weiyu Chen, Hisao Ishibuchi, and Ke Shang. Fast greedy subset selection from large candidate solution sets in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 26(4):750–764, 2021.

[10] Eng Ung Choo and Derek R Atkins. Proper efficiency in nonconvex multicriteria programming. *Mathematics of Operations Research*, 8(3):467–470, 1983.

[11] Paolo Climaco and Jochen Garcke. On minimizing the training set fill distance in machine learning regression, 2024. URL `https://arxiv.org/abs/2307.10988`.

[12] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[13] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014. doi: 10.1109/TEVC.2013.2281535.

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

[15] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE, 2002.

[16] Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1141–1148, 2006.

[17] Timo M Deist, Monika Grewal, Frank JWM Dankers, Tanja Alderliesten, and Peter AN Bosman. Multi-objective learning to predict pareto fronts using hypervolume maximization. *arXiv preprint arXiv:2102.04523*, 2021.

[18] Justin Dumouchelle, Esther Julien, Jannis Kurtz, and Elias B Khalil. Neur2bilo: Neural bilevel optimization. *arXiv preprint arXiv:2402.02552*, 2024.

[19] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

[20] Abhijith M Gopakumar, Prasanna V Balachandran, Dezhen Xue, James E Gubernatis, and Turab Lookman. Multi-objective optimization for materials discovery via adaptive design. *Scientific reports*, 8(1):3738, 2018.

[21] Fang-Qing Gu and Hai-Lin Liu. A novel weight design in multi-objective evolutionary algorithm. In *2010 International Conference on Computational Intelligence and Security*, pages 137–141. IEEE, 2010.

[22] Fangqing Gu and Yiu-Ming Cheung. Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 22(2):211–225, 2017.

[23] Yu-Ran Gu, Chao Bian, Miqing Li, and Chao Qian. Subset selection for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2023.

[24] Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. The hypervolume indicator: Problems and algorithms. *arXiv preprint arXiv:2005.00515*, 2020.

[25] Claus Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.

[26] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29–April 1, 2015. Proceedings, Part II 8*, pages 110–125. Springer, 2015.

[27] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4): 602–622, 2014. doi: 10.1109/TEVC.2013.2281534.

[28] Vishal Kaushal, Ganesh Ramakrishnan, and Rishabh Iyer. Submodlib: A submodular optimization library. *arXiv preprint arXiv:2202.10680*, 2022.

[29] Naime Ranjbar Kermany. *Towards Fairness-aware Multi-Objective Recommendation Systems*. PhD thesis, Macquarie University, 2024.

[30] Beichen Li, Bolei Deng, Wan Shou, Tae-Hyun Oh, Yuanming Hu, Yiyue Luo, Liang Shi, and Wojciech Matusik. Computational discovery of microstructured composites with optimal stiffness-toughness trade-offs, 2024.

[31] Qiuzhen Lin, Xiaozhou Wang, Bishan Hu, Lijia Ma, Fei Chen, Jianqiang Li, and Carlos A Coello Coello. Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation. *Complexity*, 2018:1–18, 2018.

[32] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019.

[33] Yajing Liu, Edwin KP Chong, Ali Pezeshki, and Zhenliang Zhang. Submodular optimization problems and greedy strategies: A survey. *Discrete Event Dynamic Systems*, 30(3):381–412, 2020.

[34] Yiping Liu, Hisao Ishibuchi, Naoki Masuyama, and Yusuke Nojima. Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts. *IEEE Transactions on Evolutionary Computation*, 24(3):439–453, 2019.

[35] Xiaoliang Ma, Yanan Yu, Xiaodong Li, Yutao Qi, and Zexuan Zhu. A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 24(4):634–649, 2020.

[36] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020.

[37] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[38] Martin Pilát and Roman Neruda. General tuning of weights in moea/d. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 965–972. IEEE, 2016.

[39] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Société Française de Statistique*, 158(1):7–36, 2017.

[40] Yutao Qi, Xiaoliang Ma, Fang Liu, Licheng Jiao, Jianyong Sun, and Jianshe Wu. Moea/d with adaptive weight adjustment. *Evolutionary computation*, 22(2):231–264, 2014.

[41] Michael Ruchte and Josif Grabocka. Scalable pareto front approximation for deep multi-objective learning. In *2021 IEEE international conference on data mining (ICDM)*, pages 1306–1311. IEEE, 2021.

[42] Serpil Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87:543–560, 2000.

[43] Jason R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. 1995.

[44] Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. Interactive design space exploration and optimization for cad models. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

[45] Ke Shang, Tianye Shu, Hisao Ishibuchi, Yang Nan, and Lie Meng Pang. Benchmarking large-scale subset selection in evolutionary multi-objective optimization. *Information Sciences*, 622: 755–770, 2023.

[46] Bofeng Shi, Turab Lookman, and Dezhen Xue. Multi-objective optimization and its application in materials science. *Materials Genome Engineering Advances*, 1(2):e14, 2023.

[47] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22(2):276–295, 2017.

[48] Jiang Siwei, Cai Zhihua, Zhang Jie, and Ong Yew-Soon. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In *2011 Seventh international conference on natural computation*, volume 3, pages 1260–1264. IEEE, 2011.

[49] Yanan Sun, Gary G Yen, and Zhang Yi. Igd indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 23(2): 173–187, 2018.

[50] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

[51] Ye Tian, Langchun Si, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Local model-based pareto front estimation for multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(1):623–634, 2022.

[52] Lihui Wang, Amos HC Ng, and Kalyanmoy Deb. *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, 2011.

[53] Zihan Wang, Bochao Mao, Hao Hao, Wenjing Hong, Chunyun Xiao, and Aimin Zhou. Enhancing diversity by local subset selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2022.

[54] Jeff Wu. Space-filling designs, 2016. URL `https://www2.isye.gatech.edu/~jeffwu/isye8813/spacefilling_designs.pdf`.

[55] Mengyuan Wu, Sam Kwong, Yuheng Jia, Ke Li, and Qingfu Zhang. Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 641–648, 2017.

[56] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pages 10607–10616. PMLR, 2020.

[57] Jie Xu, Andrew Spielberg, Allan Zhao, Daniela Rus, and Wojciech Matusik. Multi-objective graph heuristic search for terrestrial robot design. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 9863–9869. IEEE, 2021.

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

[59] Xiaoyuan Zhang, Xi Lin, and Qingfu Zhang. Pmgda: A preference-based multiple gradient descent algorithm. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

[60] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. *arXiv preprint arXiv:2308.00788*, 2023.

[61] Yong Zheng and David Xuejun Wang. A survey of recommender systems with multi-objective optimization. *Neurocomputing*, 474:141–153, 2022.

Supplementary Material

Table of Contents

A Complete proofs of theoretical results **16**
A.1 Upper and lower bounds for space filling design . . . 16
A.2 Configuration of d^{Pack} for bi-objective problems . . . 16
A.3 Theoretical results for optimization bounds . . . 18
A.4 Proof of the “equivalent conversion” argument . . . 18

B Experiment details **19**
B.1 Metrics . . . 19
B.2 Full name of multiobjective methods . . . 21
B.3 Detailed hyperparameters and licences . . . 21
B.4 Visualization for four objective problems . . . 22
B.5 Full numerical results . . . 22
B.6 Results on DTLZ5 and DTLZ6 . . . 25

C Method details **26**
C.1 Practical algorithms . . . 26
C.2 Problem formulations . . . 26
C.3 Conversion between a preference and a preference angle . . . 27
C.4 Baseline methods used in fairness classification problem . . . 28
C.5 Duplicated solutions issues caused by the mTche aggregation function . . . 28

D Miscellanies **29**
D.1 Broader impacts . . . 29
D.2 Limitations . . . 29

The appendix comprises four sections:

1. Appendix A: Detailed proofs for the main paper’s conclusions.
2. Appendix B: Additional experimental details.
3. Appendix C: Method details omitted from the main paper.
4. Appendix D: Discussion on the broader impact of our method.

A Complete proofs of theoretical results

This section provides complete proofs for the theoretical results. We first provide the properties for max-packing and space-filling designs in Appendices A.1 and A.2. Lastly, in Appendix A.3, we prove for the bi-level optimization bound by using neural network as an approximation for the inner loop optimization problem.

A.1 Upper and lower bounds for space filling design

In the following content, we proves for Theorem 3 in the main paper, i.e.,

$$\frac{1}{4}d^{\mathrm{Pack}} \leq d^{\mathrm{FD}} \leq \mathrm{FD}(\mathbb{Y}^{\mathrm{Pack}}) \leq d^{\mathrm{Pack}},$$

Proof. Following the derivation to attain Equation (3) in Pronzato [39], we can prove $d^{\mathrm{FD}} \leq d^{\mathrm{Pack}}$ as well as the claim that the fill distance between $\mathcal{T}$ and the optimal design $\mathbb{Y}^{\mathrm{Pack}}$ induced by d^{Pack} is upper bounded by d^{Pack}.

Next, we prove $d^{\mathrm{FD}} \geq \frac{1}{4}d^{\mathrm{Pack}}$ by contradiction. Consider $\mathbb{Y}^{\mathrm{FD}}$ is the design that induces d^{FD}, we have that each point in $\mathbb{Y}^{\mathrm{Pack}}$ must be within a d^{FD}-ball centered at a point in $\mathbb{Y}^{\mathrm{FD}}$, and the condition $d^{\mathrm{FD}} < \frac{1}{4}d^{\mathrm{Pack}}$ requires that a specific ball will only contain a single $\boldsymbol{y}^{(k)} \in \mathbb{Y}^{\mathrm{Pack}}$ otherwise $\mathbb{Y}^{\mathrm{Pack}}$ will not be a d^{Pack}-packing.

Since $d^{\mathrm{FD}} < \frac{1}{4}d^{\mathrm{Pack}}$, for all $k \in [K]$ the corresponding d^{FD}-ball is completely contained in the larger $(d^{\mathrm{Pack}}/2)$-ball centered at $\boldsymbol{y}^{(k)}$. However, we note $\mathcal{T}$ is connected, and thus there exists a certain $y \in \mathcal{T}$ outside all the K $(d^{\mathrm{Pack}}/2)$-balls; this certain point will not be covered by all the d^{FD}-ball centered at points in $\mathbb{Y}^{\mathrm{FD}}$ as well. Finally, the existence of the certain point contradicts the claim that $\mathbb{Y}^{\mathrm{FD}}$ is a d^{FD}-covering. ☐

Lastly, we prove for Theorem 4 in the main paper by a contradiction.

Proof. Since $d^{\mathrm{Pack}}_{K+1} > d^{\mathrm{Pack}}_{K}$, K is the maximal packing number under distance d^{Pack}_{K}.

Assume $\mathbb{Y}^{\mathrm{Pack}}_{K}$ is not a d^{Pack}_{K}-covering. Then there exists $\boldsymbol{y}^{(K+1)}$ such that $\rho(\boldsymbol{y}^{(K+1)}, \boldsymbol{y}^{(i)}) < d^{\mathrm{Pack}}_{K}$ for all $i \in [K]$. This contradicts the assumption, as $\mathbb{Y}^{\mathrm{Pack}}_{K} \cup \{\boldsymbol{y}^{(K+1)}\}$ forms a d^{Pack}_{K}-covering, implying a packing number of $K+1$ when K was maximal. Thus, $\mathbb{Y}^{\mathrm{Pack}}_{K}$ is a d^{Pack}_{K}-covering. ☐

A.2 Configuration of d^{Pack} for bi-objective problems

In the following, we prove Theorem 5 in the main paper. Before that, we prove for Lemma 8 to describe a property of $\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$, the distance between $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ when $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)} \in \mathcal{T}$.

Lemma 8. *For biobjective problem function g is strictly decreasing with respect to the first element ($y_1^{(1)}$) of $\boldsymbol{y}^{(1)}$ and strictly increasing with the first element ($y_1^{(2)}$) of $\boldsymbol{y}^{(2)}$.*

Proof. We consider a new vector $\tilde{\boldsymbol{y}}^{(1)} \in \mathcal{T}$ such that $\tilde{y}_1^{(1)} < y_1^{(1)}$. Since $\tilde{\boldsymbol{y}}^{(1)} \in \mathcal{T}$, $\boldsymbol{y}^{(1)}$ and $\tilde{\boldsymbol{y}}^{(1)}$ cannot dominate each other, implying $\tilde{y}_2^{(1)} > y_2^{(1)}$. The distance between this new solution $\tilde{\boldsymbol{y}}^{(1)}$ and $\boldsymbol{y}^{(2)}$ is:

$$\begin{aligned}
\rho(\tilde{\boldsymbol{y}}^{(1)}, \boldsymbol{y}^{(2)}) &= \sqrt{(\tilde{y}_1^{(1)} - y_1^{(2)})^2 + (\tilde{y}_2^{(1)} - y_2^{(2)})^2} \\
&= \sqrt{(y_1^{(1)} - \delta_1 - y_1^{(2)})^2 + (y_2^{(1)} + \delta_2 - y_2^{(2)})^2} \qquad (\delta_1, \delta_2 > 0) \\
&= \sqrt{(y_1^{(1)} - y_1^{(2)})^2 + \delta_1^2 - 2\delta_1(y_1^{(1)} - y_1^{(2)}) + (y_2^{(1)} - y_2^{(2)})^2 + \delta_2^2 + 2\delta_2(y_2^{(1)} - y_2^{(2)})} \\
&\geq \sqrt{(y_1^{(1)} - y_1^{(2)})^2 + (y_2^{(1)} - y_2^{(2)})^2 + C} \qquad (C > 0) \\
&> \rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}).
\end{aligned} \tag{11}$$

The previous equations show that $\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$ is strictly decreasing with respect to the first element of $\boldsymbol{y}^{(1)}$. Similarly, considering a new vector $\tilde{\boldsymbol{y}}^{(2)} \in \mathcal{T}$, $\tilde{y}_1^{(2)} > y_1^{(2)} (\tilde{y}_2^{(2)} < y_2^{(2)})$, using the same calculation method used in Equation (11), it is proved that $\rho(\boldsymbol{y}^{(1)}, \tilde{\boldsymbol{y}}^{(2)}) > \rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$. This indicates that $\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$ is strictly increasing with respect to the first element of $\boldsymbol{y}^{(2)}$. $\square$

With this established Lemma, we are now geared up for the complete proof.

Proof. The proof consists of two parts.

Part 1 proves that the neighboring distances $\rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(i+1)})$ are equal for all $i \in [K-1]$.

Part 2 proves that $\boldsymbol{y}^{(1)} = \boldsymbol{p}^{(1)}$ and $\boldsymbol{y}^{(K)} = \boldsymbol{p}^{(2)}$. $\boldsymbol{p}^{(1)}$ and $\boldsymbol{p}^{(2)}$ are the two endpoints of a PF.

Part 1. We prove $\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}) = \ldots = \rho(\boldsymbol{y}^{(K-1)}, \boldsymbol{y}^{(K)})$ by contradiction. We denote $d^{(i)}$ the distance between $\boldsymbol{y}^{(i)}$ and $\boldsymbol{y}^{(i+1)}$ and i' and j' are two indices such that

$$\begin{cases} d^{(i')} > d^{(i'+1)}, \ldots, d^{(j')}, \\ d^{(i'+1)}, \ldots, d^{(j'-1)} > d^{(j')}. \end{cases} \tag{12}$$

Without loss of generality, assume $i' < j'$. We now aim to derive a contradiction under the condition given by Equation (12). The approach is to iteratively decrease the first element of $\boldsymbol{y}^{(i'+1)}$ by a small margin $\varepsilon > 0$, yielding $\tilde{\boldsymbol{y}}^{(i'+1)}$. By Lemma 8, this adjustment ensures that $\rho(\boldsymbol{y}^{(i')}, \tilde{\boldsymbol{y}}^{(i'+1)}) < \rho(\boldsymbol{y}^{(i')}, \boldsymbol{y}^{(i'+1)})$ while $\rho(\tilde{\boldsymbol{y}}^{(i+1)}, \boldsymbol{y}^{(i+2)}) > \rho(\boldsymbol{y}^{(i+1)}, \boldsymbol{y}^{(i+2)})$. Specifically, for each k such that $i'+1 \leq k \leq j'-2$, $\boldsymbol{y}^{(k)}$ is updated to $\tilde{\boldsymbol{y}}^{(k)}$ according to the following rules:

$$\begin{cases} \rho(\boldsymbol{y}^{(i')}, \boldsymbol{y}^{(i'+1)}) - \varepsilon = \rho(\tilde{\boldsymbol{y}}^{(i')}, \tilde{\boldsymbol{y}}^{(i'+1)}), & \\ \rho(\boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k+1)}) = \rho(\tilde{\boldsymbol{y}}^{(k)}, \tilde{\boldsymbol{y}}^{(k+1)}), & i'+1 \leq k \leq j'-2, \\ \rho(\boldsymbol{y}^{(j'-1)}, \boldsymbol{y}^{(j')}) + \epsilon = \rho(\tilde{\boldsymbol{y}}^{(j'-1)}, \tilde{\boldsymbol{y}}^{(j')}). & \end{cases} \tag{13}$$

When ε is sufficiently small, $\tilde{j}' = \arg\min\{\rho(\tilde{\boldsymbol{y}}^{(i')}, \tilde{\boldsymbol{y}}^{(i'+1)}), \ldots, \rho(\tilde{\boldsymbol{y}}^{(j'-1)}, \tilde{\boldsymbol{y}}^{(j')})\} = j'$, meaning the minimal index before and after adjustment remains unchanged. However, the value of the adjusted distance $\rho(\tilde{\boldsymbol{y}}^{(j')}, \tilde{\boldsymbol{y}}^{(j'+1)})$ is increased from $\rho(\boldsymbol{y}^{(j')}, \boldsymbol{y}^{(j'+1)})$ by ε, showing the original design is not a max-packing design, leading to a contradiction. If conditions

$$\begin{cases} d^{(i')} < d^{(i'+1)}, \ldots, d^{(j')}, \\ d^{(i'+1)}, \ldots, d^{(j'-1)} < d^{(j')}, \end{cases} \tag{14}$$

hold, a similar argument as above can be used to derive a contradiction. Since the selection of i' and j' is arbitrary, this implies that for any interval between i' and j', both Equation (12) and Equation (14) cannot hold simultaneously. This leads to the following equality, which aligns with our intended design goal:

$$\rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}) = \ldots = \rho(\boldsymbol{y}^{(K-1)}, \boldsymbol{y}^{(K)}). \tag{15}$$

Remark. The key component of our proof is the validity of Lemma 8, which applies to a *two-dim PF* only. This leads to a contradiction with both Equation (12) and Equation (14), forming the core of our argument.

Part 2. We prove $\boldsymbol{y}^{(1)} = \boldsymbol{p}^{(1)}$ by contradiction. The proof for $\boldsymbol{y}^{(K)} = \boldsymbol{p}^{(2)}$ follows similarly. Assuming $\boldsymbol{y}^{(1)} \neq \boldsymbol{p}^{(1)}$, we replace $\boldsymbol{y}^{(1)}$ with $\boldsymbol{p}^{(1)}$, forming a new configuration $[\tilde{\boldsymbol{y}}^{(1)}(\boldsymbol{p}^{(1)}), \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}]$. Based on Part 1, where we showed equal neighboring distances between vectors, the condition

$$\rho(\tilde{\boldsymbol{y}}^{(1)}, \boldsymbol{y}^{(2)}) > \rho(\boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}) = \ldots = \rho(\boldsymbol{y}^{(K-1)}, \boldsymbol{y}^{(K)})$$

holds.

Next, we replace $\boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K-1)}$ with $\tilde{\boldsymbol{y}}^{(2)}, \ldots, \tilde{\boldsymbol{y}}^{(K-1)}$ on the PF, such that $\tilde{y}_1^{(i)} = y_1^{(i)} - \epsilon^{(i)}$, for $2 \leq i \leq K-1$, ensuring

$$\rho(\tilde{\boldsymbol{y}}^{(i)}, \tilde{\boldsymbol{y}}^{(i+1)}) > \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(i+1)}), \quad i \in [K-1].$$

By Lemma 8, moving $\boldsymbol{y}^{(1)}$ to $\tilde{\boldsymbol{y}}^{(1)}$ results in $\rho(\tilde{\boldsymbol{y}}^{(1)}, \boldsymbol{y}^{(2)}) > \rho(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$. Iteratively shifting $\boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K-1)}$ ensures $\tilde{y}_1^{(2)} < y_1^{(2)}$ and so on, until $\tilde{y}_1^{(K-1)} < y_1^{(K-1)}$, completing the process.

This leads to a contradiction, proving that $\boldsymbol{y}^{(1)} = \boldsymbol{p}^{(1)}$ and $\boldsymbol{y}^{(K)} = \boldsymbol{p}^{(2)}$.

□

A.3 Theoretical results for optimization bounds

In this part, we prove for Theorem 7, which bounds the optimization error caused by neural network in the bi-level optimization problem (Equation (10)).

Proof. Consider the function $\rho(\cdot, \cdot)$, which measures the distance between two vectors. Given our assumption, we derive the error between distances computed under $\boldsymbol{h}$ and $\boldsymbol{h}_\phi$. For any two points $\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}$ in the image of $\boldsymbol{h}$, the error in their distances compared to $\boldsymbol{h}_\phi$ can be bounded as follows:

$$\begin{aligned} |\rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) - \rho(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}), \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)}))| &= |\|\boldsymbol{y}^{(i)} - \boldsymbol{y}^{(j)}\| - \|\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}) - \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)})\|| \\ &\leq \|\boldsymbol{y}^{(i)} - \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}) + \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)}) - \boldsymbol{y}^{(j)}\| \\ &\leq \|\boldsymbol{y}^{(i)} - \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)})\| + \|\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)}) - \boldsymbol{y}^{(j)}\| \\ &\leq \epsilon + \epsilon = 2\epsilon. \end{aligned} \tag{16}$$

This follows from the triangle inequality and the assumption $|\boldsymbol{h}_\phi(\boldsymbol{\vartheta}) - h(\boldsymbol{\vartheta})| \leq \epsilon$.

Given this pairwise bound, the overall configuration of points is such that the minimization of distances among points under $\boldsymbol{h}_\phi$ will either match or exceed the minimization under $\boldsymbol{h}$ within the bounds of 2ϵ:

$$\min_{1 \leq i < j \leq K} \rho(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}), \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)})) \leq \min_{1 \leq i < j \leq K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) + 2\epsilon. \tag{17}$$

Considering the maximization of these minimum distances across all configurations $\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}$, we have:

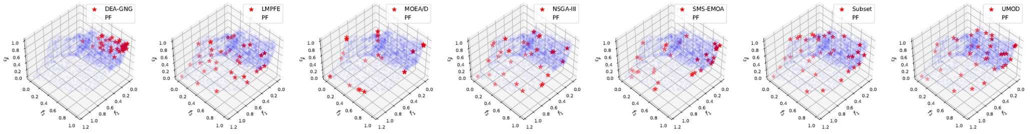
$$\begin{aligned} \left| \max_{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}} \min_{1 \leq i < j \leq K} \rho(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}), \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j)})) - \max_{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}} \min_{1 \leq i < j \leq K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) \right| \\ \leq \max_{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}} 2\epsilon = 2\epsilon. \end{aligned} \tag{18}$$

Thus, the error in the optimal maximal minimal distance under the model transformation can be bounded by 2ϵ, completing the proof. □

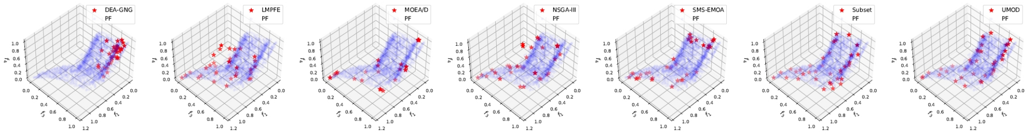
A.4 Proof of the "equivalent conversion" argument

This argument is actually a direct corollary the following two lemmas.

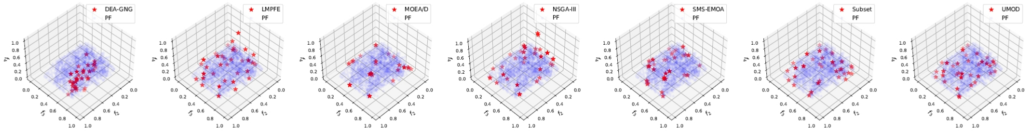
Lemma 9 (Adapted from [10], Theorem 3.1)**.** *A solution $\boldsymbol{x}$ is weakly Pareto optimal iff there exists a weight vector $\boldsymbol{\lambda}$ such that $\boldsymbol{x}$ is (one of) an optimal solution of the modified Tchebycheff function.*



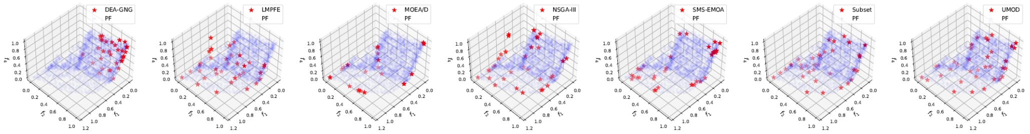
DEAGNG-P1 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P2 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P3 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P4 LMPFE MOEAD NSGA3 SMS Subset UMOD

Figure 8: Results on RE41.

Lemma 10 (Modified from [37], Theorem 2.6.2)**.** *If an aggregation function is decreasing w.r.t. vector* $\boldsymbol{f}(\boldsymbol{x})$ *(i.e.,* $g_{\boldsymbol{\lambda}}(\boldsymbol{f}(\boldsymbol{x})) \leq g_{\boldsymbol{\lambda}}(\boldsymbol{f}(\boldsymbol{x}'))$ *when* $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}'), \forall i \in [m]$ *and at least one index* j $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}')$*, then one of the optimal solution* $\boldsymbol{x}^*$ *of* $g_{\boldsymbol{\lambda}}(\boldsymbol{f}(\boldsymbol{x}))$ *is a* weakly *Pareto optimal solution for the original MOP. In addition, if the optimality is unique,* $\boldsymbol{x}^*$ *is* Pareto optimal.

Based on the first Lemma, we know that for any weakly Pareto objective, there is a corresponding preference vector that solving the modified Tchebycheff function can recover this vector. Furthermore, since we assume the uniqueness of the optimality of the modified Tchebycheff function, thus according to the second lemma, solving the modified Tchebycheff function only yields Pareto optimal solutions. Combining these two arguments, we achieve that for any Pareto optimal objective, there exists a preference vector such that solving the corresponding modified Tchebycheff function yield this Pareto optimal vector.

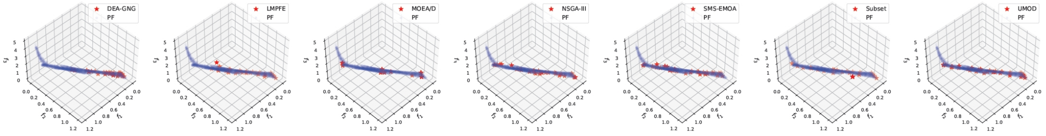
B Experiment details

This section has four parts. In Appendix B.1, we explain the metrics used in the experiments in details. In Appendix B.3, we list the necessary hyperparameters and license. In Appendix B.4, we visualize the results on four-objective problems by projection. Lastly, Appendix B.5 list for all numerical results for all experiments.

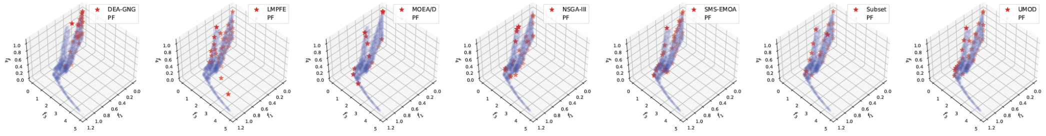
B.1 Metrics

To evaluate the uniformity and quality of these solutions, we use various performance indicators, detailed and mathematically expressed below.

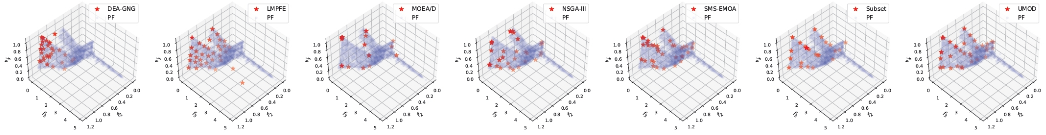
1. Hypervolume (HV) (↑) [24] both assesses convergence to a PF and solution diversity. Low HV values suggest poor convergence to the PF, while comparisons are significant when HVs are substantially high. The hypervolume indicator measures the dominated volume by at least one



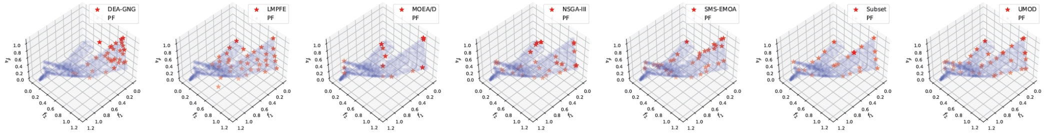
DEAGNG-P1 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P2 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P3 LMPFE MOEAD NSGA3 SMS Subset UMOD



DEAGNG-P4 LMPFE MOEAD NSGA3 SMS Subset UMOD

Figure 9: Results on RE42.

objective belongs to the set $\mathbb{Y}$ with a reference point $\boldsymbol{r}$.

$$\mathrm{HV}_{\boldsymbol{r}} = \mathrm{Vol}(\{\boldsymbol{y} | \exists \boldsymbol{y}' \in \mathbb{Y}, \boldsymbol{y}' \preceq \boldsymbol{y} \preceq \boldsymbol{r}\}).$$

2. IGD [26] indicator of a set $\mathbb{A}$ with a reference set $\mathbb{Z}$ is defined as:

$$\mathrm{IGD}(\mathbb{A}) = \frac{1}{|\mathbb{Z}|} \sum_{i=1}^{|\mathbb{Z}|} d_i,$$

where d_i represents the Euclidean distance from z_i to the nearest distance in the set of $\mathbb{A}$.

3. Sparsity (↓) [56] is a measure calculated from the squared distances among solution vectors that are sorted according to their non-dominance levels [14]. The mathematical definition is given by:

$$\text{Sparsity} = \frac{1}{N-1} \sum_{j=1}^{m} \sum_{i=1}^{N-1} \left(\tilde{y}_j^{(i)} - \tilde{y}_j^{(i+1)} \right)^2,$$

where $\tilde{\boldsymbol{y}}^{(i)}$ are the objective vectors arranged in a non-dominated sorting order from the set $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}\}$. Here, m represents the number of objectives, and N is the number of solutions. A lower Sparsity value indicates a more uniformly distributed set of solutions along the Pareto front. Specifically, for a Pareto front with a two-dimensional linear shape, the sparsity indicator reaches its minimum when the objectives are spaced equidistantly.

4. Spacing (↓) [43]: This metric assesses solution distribution uniformity by calculating the standard deviation of $\tilde{d}^{(i)}$, the minimal distance between a solution $\boldsymbol{y}^{(i)}$ and its nearest neighbor, where $d^{(i)} = \min_{j \in [m], j \neq i} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$. A lower spacing indicator implies evenly spaced solutions, reflecting uniform distribution. A spacing indicator of zero indicates that all Pareto objectives have equally minimal neighborhood distances.

Table 4: Full name table, which has three parts. The first part is related with evolutionary algorithms, the second part is related with gradient-based methods, the third part is related to indicators, while the last part is related to multiobjective optimization concepts.

Short Name	Full name
DEA-GNG	**D**ecomposition based **E**volutionary **A**lgorithm guided by **G**rowing **N**eural **G**as
LMPFE	Evolutionary algorithm with **L**ocal **M**odel based **P**areto **F**ront **E**stimation
NSGA3	**N**ondominated **S**orting **G**enetic **A**lgorithm 3
SMS-EMOA	**S** **M**etric **S**election based **E**volutionary **M**ultiobjective **O**ptimization **A**lgorithm
MOEA/D	**M**ulti**O**bjective **E**volutionary **A**lgorithm based on **D**ecomposition
MOEA/D-AWA	**MOEA/D** with **A**daptive **W**eight **A**djustment
UMOD	**U**niform **M**ultiobjective **O**ptimization based on **D**ecomposition
MOO-SVGD	**M**ulti**O**bjective **O**ptimization **S**tein **V**ariational **G**radient **D**escent
EPO	**E**xact **P**areto **O**ptimization
PMGDA	**P**reference based **M**ultiple **G**radient **D**escent **A**lgorithm
Agg-LS	**Agg**regation function based on **L**inear **S**calarization
Agg-PBI	**Agg**regation function based on **P**enalty **B**ased **I**ntersection
Agg-Tche	**Agg**regation function based **Tche**bycheff Scalarization
IGD	**I**nverted **G**eneral **D**istance
HV	**H**yper**V**olume
MOO	**M**ulti**O**bjective **O**ptimization
MOP	**M**ultiobjective **O**ptimization **P**roblem

5. **Uniformity** (↑) and **Smooth Uniformity** (↑) indicators, as introduced by [42], evaluate the distribution of solutions. The Uniformity indicator, δ_{Unif}, is defined as the minimum distance between any two solutions:

$$\delta_{\text{Unif}} = \min_{1 \le i < j \le K} \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}).$$

On the other hand, the Smooth Uniformity indicator, $\tilde{\delta}_{\text{Unif}}$, incorporates a logarithmic sum exponential function to average distances among solutions. It introduces a sensitivity parameter η (set to 20 in this study) to highlight the overall distribution of distances:

$$\tilde{\delta}_{\text{Unif}} = -\frac{2}{\eta K(K-1)} \log \sum_{1 \le i < j \le K} \exp(\eta \cdot \rho(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})).$$

B.2 Full name of multiobjective methods

To avoid confusion, we provide the full names of the baseline MOO methods in Table 4. For SMS-MOEA (MOEA using S-Metric Selection), the term "S-metric" refers to the "hypervolume" metric.

B.3 Detailed hyperparameters and licences

Table 5 lists the hyperparameters for implementing MOPs solved by evolutionary algorithms. The system used features an Intel Core i7-10700 CPU and a NVIDIA RTX 3080 GPU.

Our method is implemented in the MOEA/D framework using Pymoo [7], without modifying the MOEA/D hyperparameters. The main difference is the use of a PF model (a fully-connected neural network) to map preference angles to Pareto objectives and update preference vectors. The hyperparameters for the PF model are listed in Table 5.

For fairness classification problems, we use an additional fully connected network to classify input features into corresponding classes. The parameters of this network serve as the decision variables (x) for a multiobjective problem. Details of the fairness classification problems are shown in Table 6.

Table 6: Fairness classification problem details and network architectures. Act. is the short name for activation.

Dataset	Features	Architecture	Act. function	# Params	Samples	Sensitivity
Adult	88	88-60-25-1	ReLU	6891	34188	Sex
Compass	20	20-60-25-1	ReLU	2811	4319	Sex

Table 5: Hyper-parameters used in UMOD-MOEA

Hyper-parameters	Values	Hyper-parameters	Values
Crossover	SBX (Simulated Binary Crossover)	Mutation	PM (Polynomial mutation)
SBX mating threshold	0.5	SBX offsprings	2
PM mutation probability	0.9	PFL optimizer	SGD
PFL network	(m-1)-128-128-128 $\to$ m	PFL activation function	ReLU
PFL Learning rate	1e-3	PFL training epoch	1000
Number of preferences ($m = 2$)	8	Number of preferences ($m = 3$)	21
Number of preferences ($m = 4$)	35	Preference initialization	Das-Dennis [12]
Number of neighborhood in MOEA/D	3	Probability of mating	0.3
Number of experiments (MOEA)	31	Number of experiments (MOML)	5
Number of fitness value (2 obj)	40,000	Number of fitness value (3 obj)	126,000
Number of fitness value (4 obj)	350,000		

(License) To close this subsection, we would like to mention that the license used for Adult follows Creative Commons Attribution 4.0 International (CC BY 4.0) license and Compas is supported by Database Contents License (DbCL) v1.0 license.

B.4 Visualization for four objective problems

This section presents the visualization results for four-objective problems (see Figures 8 and 9). Due to the difficulty of visualizing four-dimensional space, we project the Pareto objectives into 3-D spaces: (f_1, f_2, f_3), (f_1, f_2, f_4), (f_1, f_3, f_4), and (f_2, f_3, f_4), labeled P-1 to P-4 in Figures 8 and 9.

Despite losing some information in the projections, meaningful conclusions can still be drawn from these figures:

1. DEAGNG and LMPFE can find partial parts of the true Pareto front. For the P1, P2, and P4 projections, DEAGNG typically finds only a small portion in the upper right of the 3-D Pareto front, while LMPFE misses a small part of this region. In contrast, the proposed method captures a more extensive span of the Pareto front, covering the largest area.

2. For a four-objective problem, MOEA/D finds many duplicate Pareto objectives on the PF boundary by using fixed preference vectors. This highlights the importance of finding optimal preference vectors to achieve a more uniform PF.

3. The proposed UMOD method finds more uniform Pareto objectives on the Pareto front compared to NSGA3, SMS-EMOA, and the Subset selection method. This aligns with Table 9 in the main paper, showing UMOD has a much lower IGD indicator than these three methods.

B.5 Full numerical results

We report the full numerical results for bi-objective, tri-objective, and four-objective problems in Tables 7 to 9, respectively. Each experiment was conducted with 31 random seeds. For each data point, $X(Y)(C)$, X is the mean value, (Y) is the standard deviation, and (C) is the rank among all eight methods.

The hypervolume, Uniform, and Smooth Uniform (SUniform) are preferred larger, while IGD, spacing, sparsity, and fill distances are preferred smaller. The optimal result averaged across all seeds is marked in bold. In the last row of each table, we calculate the mean rank of each indicator, with the highest one in bold.

The tables show that, with a small solution budget, the uniformity indicators IGD, Spacing, Uniform, SUniform, and FD achieve the best results, outperforming previous methods significantly.

SMS-EMOA achieves the highest HV value among all methods, but HV is only a rough measure of uniformity, validating that optimizing hypervolume alone does not ensure the best uniform distribution.

These tables also validate the effectiveness of maximizing pairwise distances, a proper surrogate for fill distance. This suggests that in practice, maximal packing distance can more tightly bound the minimal fill distance. An interesting finding is that the IGD indicator, an important measure in MOO serving as the average covering radius of the size-K optimized set, is optimized by maximizing the pairwise distance. This interesting empirical finding is worthy of further investigation.

Table 7: Full results for biobjective problems, based on 31 random seeds, include the standard deviation and rankings across all methods. The ranking values in the last row are averaged across all problems.

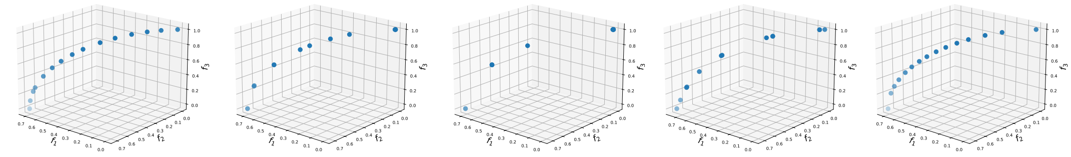
	Indicator	DEA-GNG	LMPFE	Subset	NSGA3	SMS-EMOA	MOEA/D	MOEA/D-AWA	UMOD
ZDT1	HV	1.03 (0.00) (6)	1.03 (0.00) (2)	1.02 (0.00) (7)	1.03 (0.00) (5)	1.04 (0.00) (1)	1.03 (0.00) (4)	1.03 (0.00) (3)	**1.04 (0.00) (0)**
	IGD	5.68 (0.29) (7)	5.42 (0.21) (6)	5.21 (0.40) (1)	5.28 (0.00) (3)	5.22 (0.07) (2)	5.30 (0.00) (4)	5.42 (0.02) (5)	**5.19 (0.00) (0)**
	Spacing	6.35 (1.74) (7)	2.44 (1.82) (2)	3.51 (0.47) (3)	5.90 (0.00) (5)	1.91 (0.46) (1)	5.92 (0.01) (6)	3.69 (0.04) (4)	**0.12 (0.06) (0)**
	Sparsity	4.80 (0.19) (7)	4.55 (0.12) (4)	**2.59 (0.18) (0)**	4.60 (0.00) (5)	4.48 (0.01) (2)	4.61 (0.00) (6)	4.51 (0.01) (3)	4.39 (0.00) (1)
	Uniform	1.33 (0.11) (6)	1.67 (0.32) (2)	0.89 (0.10) (7)	1.53 (0.00) (3)	1.85 (0.03) (1)	1.51 (0.00) (4)	1.51 (0.00) (5)	**2.07 (0.01) (0)**
	SUniform	0.47 (0.12) (6)	0.68 (0.13) (2)	0.09 (0.09) (7)	0.49 (0.00) (4)	0.74 (0.01) (1)	0.48 (0.00) (5)	0.53 (0.00) (3)	**0.77 (0.00) (0)**
	Fill Distance	1.60 (0.23) (6)	1.23 (0.09) (2)	2.00 (0.18) (7)	1.55 (0.00) (5)	1.16 (0.05) (1)	1.50 (0.00) (4)	1.43 (0.03) (3)	**1.04 (0.01) (0)**
ZDT2	HV	0.71 (0.00) (5)	0.65 (0.07) (6)	0.63 (0.00) (7)	0.71 (0.00) (4)	**0.71 (0.00) (0)**	0.71 (0.00) (3)	0.71 (0.00) (2)	0.71 (0.00) (1)
	IGD	5.43 (0.13) (4)	10.80 (6.71) (7)	6.69 (0.00) (6)	5.31 (0.00) (2)	5.84 (0.14) (5)	5.31 (0.00) (3)	5.29 (0.00) (1)	**5.23 (0.01) (0)**
	Spacing	3.92 (1.97) (6)	2.11 (0.88) (1)	3.02 (0.00) (5)	3.01 (0.00) (4)	4.87 (0.53) (7)	3.01 (0.01) (3)	2.30 (0.03) (2)	**0.25 (0.23) (0)**
	Sparsity	4.66 (0.09) (5)	8.87 (5.33) (7)	**2.26 (0.00) (0)**	4.54 (0.00) (3)	4.69 (0.06) (6)	4.54 (0.00) (4)	4.52 (0.00) (2)	4.45 (0.00) (1)
	Uniform	1.41 (0.27) (5)	1.25 (0.64) (6)	0.85 (0.00) (7)	1.63 (0.00) (4)	1.71 (0.11) (2)	1.63 (0.00) (3)	1.83 (0.02) (1)	**2.05 (0.06) (0)**
	SUniform	0.57 (0.10) (5)	0.16 (0.68) (6)	-0.04 (0.00) (7)	0.68 (0.00) (3)	0.63 (0.02) (4)	0.68 (0.00) (2)	0.71 (0.00) (1)	**0.78 (0.00) (0)**
	Fill Distance	1.36 (0.11) (4)	2.82 (1.88) (7)	2.39 (0.00) (6)	1.24 (0.00) (3)	1.63 (0.09) (5)	1.24 (0.00) (2)	1.23 (0.00) (1)	**1.07 (0.03) (0)**
ZDT3	HV	**0.91 (0.00) (0)**	0.90 (0.01) (2)	0.90 (0.00) (4)	0.89 (0.00) (7)	0.91 (0.02) (1)	0.90 (0.00) (3)	0.89 (0.02) (5)	0.89 (0.02) (6)
	IGD	38.60 (0.25) (1)	38.78 (0.31) (2)	**38.58 (0.00) (0)**	39.45 (0.03) (5)	39.23 (2.25) (4)	39.09 (0.00) (3)	39.94 (2.08) (7)	39.83 (2.10) (6)
	Spacing	6.60 (1.62) (4)	**3.08 (1.42) (0)**	12.38 (0.00) (7)	5.12 (2.21) (3)	7.85 (0.88) (5)	5.02 (0.00) (2)	8.07 (0.98) (6)	3.57 (1.66) (1)
	Sparsity	3.83 (0.16) (5)	3.83 (0.12) (6)	7.27 (0.00) (7)	3.82 (0.21) (4)	3.59 (0.50) (1)	3.65 (0.00) (2)	3.70 (0.54) (3)	**3.45 (0.40) (0)**
	Uniform	0.89 (0.50) (2)	**1.42 (0.26) (0)**	0.00 (0.00) (7)	0.69 (0.54) (4)	0.75 (0.07) (3)	0.66 (0.00) (5)	0.28 (0.29) (6)	1.20 (0.19) (1)
	SUniform	0.06 (0.28) (3)	**0.39 (0.10) (0)**	-0.91 (0.00) (7)	0.04 (0.33) (4)	-0.12 (0.09) (5)	0.09 (0.00) (2)	-0.32 (0.12) (6)	0.30 (0.04) (1)
	Fill Distance	8.88 (0.02) (4)	**8.87 (0.00) (0)**	8.87 (0.00) (1)	8.88 (0.00) (3)	9.22 (0.71) (5)	8.88 (0.00) (2)	9.23 (0.71) (7)	9.22 (0.71) (6)
ZDT4	HV	0.75 (0.17) (7)	1.03 (0.01) (3)	1.02 (0.00) (6)	1.03 (0.00) (2)	**1.04 (0.00) (0)**	1.03 (0.00) (4)	1.03 (0.00) (5)	1.04 (0.00) (1)
	IGD	38.29 (19.75) (7)	5.65 (0.19) (6)	5.49 (0.07) (5)	5.28 (0.00) (2)	**5.19 (0.03) (0)**	5.31 (0.01) (3)	5.45 (0.05) (4)	5.21 (0.01) (1)
	Spacing	5.47 (2.97) (5)	4.69 (1.93) (4)	2.72 (0.22) (2)	5.91 (0.00) (6)	1.60 (0.58) (1)	5.93 (0.01) (7)	3.66 (0.03) (3)	**0.22 (0.06) (0)**
	Sparsity	1.72 (1.93) (0)	4.71 (0.16) (7)	2.52 (0.00) (1)	4.60 (0.00) (5)	4.47 (0.01) (3)	4.62 (0.00) (6)	4.52 (0.02) (4)	4.39 (0.01) (2)
	Uniform	0.41 (0.55) (7)	1.18 (0.46) (5)	1.06 (0.01) (6)	1.53 (0.00) (2)	1.90 (0.08) (1)	1.52 (0.00) (3)	1.50 (0.02) (4)	**2.05 (0.01) (0)**
	SUniform	-0.75 (0.64) (7)	0.45 (0.22) (5)	0.10 (0.01) (6)	0.49 (0.00) (3)	0.75 (0.02) (1)	0.49 (0.00) (4)	0.53 (0.00) (2)	**0.77 (0.00) (0)**
	Fill Distance	8.54 (3.59) (7)	1.33 (0.09) (2)	2.03 (0.02) (6)	1.54 (0.00) (5)	1.10 (0.03) (1)	1.52 (0.02) (4)	1.45 (0.01) (3)	**1.06 (0.01) (0)**
ZDT6	HV	0.66 (0.00) (6)	0.66 (0.00) (3)	0.44 (0.00) (7)	0.66 (0.00) (2)	0.66 (0.00) (1)	0.66 (0.00) (4)	0.66 (0.00) (5)	**0.66 (0.00) (0)**
	IGD	4.81 (0.60) (6)	4.25 (0.09) (3)	15.28 (0.00) (7)	4.19 (0.00) (1)	4.20 (0.03) (2)	4.32 (0.00) (4)	4.37 (0.00) (5)	**4.18 (0.00) (0)**
	Spacing	5.04 (2.57) (7)	2.52 (0.99) (5)	2.12 (0.00) (3)	1.94 (0.00) (2)	1.30 (0.37) (1)	2.48 (0.00) (4)	3.50 (0.04) (6)	**0.12 (0.01) (0)**
	Sparsity	3.24 (0.28) (7)	2.95 (0.06) (6)	**1.83 (0.00) (0)**	2.89 (0.00) (3)	2.88 (0.02) (2)	2.91 (0.00) (4)	2.95 (0.00) (5)	2.86 (0.00) (1)
	Uniform	0.88 (0.21) (6)	1.23 (0.19) (4)	0.79 (0.00) (7)	1.32 (0.00) (2)	1.48 (0.05) (1)	1.23 (0.00) (3)	0.97 (0.01) (5)	**1.67 (0.00) (0)**
	SUniform	0.06 (0.15) (6)	0.26 (0.06) (4)	-0.29 (0.00) (7)	0.31 (0.00) (2)	0.33 (0.02) (1)	0.28 (0.00) (3)	0.22 (0.00) (5)	**0.35 (0.00) (0)**
	Fill Distance	1.27 (0.30) (6)	1.05 (0.08) (5)	3.89 (0.00) (7)	0.92 (0.00) (1)	0.94 (0.05) (2)	0.97 (0.00) (3)	0.97 (0.00) (4)	**0.81 (0.01) (0)**
RE21	HV	1.24 (0.00) (5)	1.23 (0.01) (7)	1.24 (0.00) (4)	1.24 (0.00) (2)	1.24 (0.00) (1)	1.24 (0.00) (6)	1.24 (0.00) (3)	**1.24 (0.00) (0)**
	IGD	4.63 (0.28) (5)	4.61 (0.13) (4)	5.15 (0.01) (6)	4.44 (0.00) (3)	4.23 (0.02) (1)	5.40 (0.02) (7)	4.33 (0.04) (2)	**4.12 (0.00) (0)**
	Spacing	6.63 (1.16) (6)	3.62 (0.93) (4)	1.38 (0.00) (1)	5.71 (0.01) (5)	3.19 (0.37) (2)	10.49 (0.04) (7)	3.47 (0.41) (3)	**0.12 (0.05) (0)**
	Sparsity	3.10 (0.21) (6)	2.90 (0.10) (4)	**1.74 (0.00) (0)**	3.02 (0.00) (5)	2.82 (0.02) (2)	3.87 (0.02) (7)	2.84 (0.01) (3)	2.70 (0.00) (1)
	Uniform	0.81 (0.12) (7)	0.95 (0.16) (5)	0.95 (0.00) (4)	1.16 (0.00) (2)	1.26 (0.05) (1)	0.93 (0.00) (6)	1.11 (0.03) (3)	**1.62 (0.01) (0)**
	SUniform	-0.03 (0.07) (5)	0.12 (0.08) (3)	-0.17 (0.00) (7)	0.08 (0.00) (4)	0.20 (0.01) (1)	-0.17 (0.00) (6)	0.13 (0.01) (2)	**0.31 (0.00) (0)**
	Fill Distance	1.45 (0.20) (4)	1.21 (0.12) (3)	2.62 (0.01) (7)	1.47 (0.00) (5)	1.09 (0.04) (1)	2.11 (0.01) (6)	1.15 (0.02) (2)	**0.83 (0.00) (0)**
RE22	HV	1.17 (0.00) (7)	1.18 (0.00) (6)	1.18 (0.00) (4)	1.18 (0.00) (5)	1.18 (0.00) (0)	1.18 (0.00) (3)	1.18 (0.00) (2)	1.18 (0.00) (1)
	IGD	4.63 (0.40) (5)	4.93 (0.25) (7)	4.48 (0.00) (4)	4.35 (0.00) (1)	4.42 (0.04) (2)	4.92 (0.00) (6)	4.47 (0.02) (3)	**4.25 (0.02) (0)**
	Spacing	2.78 (0.95) (2)	3.11 (1.00) (5)	6.57 (0.00) (6)	2.90 (0.00) (3)	3.08 (0.12) (4)	6.78 (0.01) (7)	2.25 (0.64) (1)	**0.34 (0.02) (0)**
	Sparsity	3.01 (0.34) (5)	3.26 (0.22) (7)	2.84 (0.00) (2)	2.77 (0.00) (1)	2.90 (0.03) (3)	3.12 (0.00) (6)	2.93 (0.03) (4)	**2.70 (0.01) (0)**
	Uniform	1.08 (0.16) (2)	0.96 (0.01) (5)	0.49 (0.00) (7)	1.06 (0.00) (3)	1.03 (0.01) (4)	0.83 (0.00) (6)	1.09 (0.16) (1)	**1.56 (0.01) (0)**
	SUniform	0.08 (0.01) (4)	0.02 (0.05) (5)	-0.12 (0.00) (7)	0.19 (0.00) (2)	0.20 (0.02) (1)	-0.06 (0.00) (6)	0.14 (0.03) (3)	**0.30 (0.01) (0)**
	Fill Distance	1.39 (0.32) (6)	1.54 (0.14) (7)	1.33 (0.00) (5)	1.06 (0.00) (2)	1.04 (0.06) (1)	1.33 (0.00) (4)	1.29 (0.00) (3)	**0.90 (0.01) (0)**
Rank	HV	5.14	4.14	5.57	3.86	**0.57**	3.86	3.57	1.29
	IGD	5	4.86	4	2.43	2.29	4.43	4	**1**
	Spacing	5.71	2.71	3.86	3.86	3	5.14	3.57	**0.14**
	Sparsity	5.29	5.71	1.43	3.71	2.71	4.86	3.43	**0.86**
	Uniform	5	3.86	6.43	3	1.71	4.29	3.57	**0.14**
	SUniform	5.14	3.57	6.86	3.14	2	4	3.14	**0.14**
	Fill Distance	5.43	3.57	5.57	3.43	2.29	3.57	3.29	**0.86**

Table 8: Full numerical and ranking results on three-objective problems.

	Indicator	DEA-GNG	LMPFE	Subset	NSGA3	SMS-EMOA	MOEA/D	MOEA/D-AWA	UMOD
DTLZ1	HV	1.51 (0.11) (7)	1.69 (0.00) (2)	1.69 (0.00) (6)	1.69 (0.00) (4)	**1.69 (0.00) (0)**	1.69 (0.00) (3)	1.69 (0.00) (5)	1.69 (0.00) (1)
	IGD	14.62 (3.52) (7)	4.91 (0.10) (4)	4.94 (0.00) (6)	**4.85 (0.00) (0)**	4.86 (0.05) (1)	4.86 (0.01) (2)	4.93 (0.09) (5)	4.87 (0.00) (3)
	Spacing	3.12 (0.87) (7)	1.14 (0.10) (3)	1.94 (0.00) (6)	**0.01 (0.00) (0)**	1.43 (0.35) (4)	0.02 (0.00) (1)	1.77 (1.75) (5)	0.09 (0.02) (2)
	Sparsity	**0.30 (0.22) (0)**	0.48 (0.02) (3)	0.36 (0.00) (1)	0.74 (0.00) (6)	0.41 (0.02) (2)	0.75 (0.00) (7)	0.73 (0.01) (4)	0.73 (0.00) (5)
	Uniform	0.02 (0.03) (7)	1.05 (0.05) (3)	0.68 (0.00) (6)	1.41 (0.00) (1)	0.86 (0.14) (4)	**1.41 (0.00) (0)**	0.81 (0.60) (5)	1.39 (0.01) (2)
	SUniform	-1.99 (0.20) (7)	-0.94 (0.01) (3)	-0.99 (0.00) (6)	-0.90 (0.00) (2)	-0.95 (0.01) (5)	-0.90 (0.00) (1)	-0.95 (0.05) (4)	**-0.90 (0.00) (0)**
	Fill Distance	4.59 (0.64) (7)	1.03 (0.13) (6)	1.01 (0.00) (4)	**0.77 (0.00) (0)**	0.96 (0.05) (3)	0.77 (0.00) (1)	1.01 (0.28) (5)	0.78 (0.00) (2)
DTLZ2	HV	1.01 (0.01) (7)	1.05 (0.00) (6)	1.08 (0.00) (1)	1.06 (0.00) (5)	**1.08 (0.00) (0)**	1.06 (0.00) (3)	1.06 (0.00) (4)	1.07 (0.00) (2)
	IGD	12.52 (0.64) (2)	12.46 (0.09) (1)	15.14 (0.00) (6)	12.54 (0.00) (3)	15.45 (0.25) (7)	12.55 (0.00) (5)	12.55 (0.00) (4)	**12.19 (0.04) (0)**
	Spacing	5.29 (1.72) (2)	2.64 (0.39) (1)	8.97 (0.00) (7)	5.45 (0.00) (3)	7.15 (0.51) (6)	5.45 (0.01) (4)	5.45 (0.01) (5)	**0.52 (0.21) (0)**
	Sparsity	**1.50 (0.13) (0)**	2.18 (0.13) (2)	2.42 (0.00) (3)	2.43 (0.00) (4)	2.75 (0.14) (7)	2.43 (0.00) (6)	2.43 (0.00) (5)	1.61 (0.13) (1)
	Uniform	1.58 (0.19) (5)	2.45 (0.13) (1)	0.96 (0.00) (7)	2.43 (0.00) (4)	1.51 (0.15) (6)	2.43 (0.00) (3)	2.43 (0.00) (2)	**3.17 (0.09) (0)**
	SUniform	0.36 (0.23) (5)	0.97 (0.04) (1)	-0.04 (0.00) (7)	0.95 (0.00) (4)	0.18 (0.07) (6)	0.95 (0.00) (2)	0.95 (0.00) (3)	**1.18 (0.02) (0)**
	Fill Distance	3.20 (0.60) (5)	2.69 (0.10) (4)	3.54 (0.00) (7)	2.59 (0.00) (1)	3.42 (0.16) (6)	2.59 (0.00) (3)	2.59 (0.00) (2)	**2.37 (0.08) (0)**
DTLZ3	HV	0.80 (0.20) (7)	1.02 (0.02) (6)	1.08 (0.00) (1)	1.06 (0.00) (4)	1.08 (0.00) (0)	1.06 (0.00) (2)	1.06 (0.00) (3)	1.06 (0.00) (5)
	IGD	31.92 (16.92) (7)	14.80 (1.03) (4)	15.15 (0.00) (5)	12.55 (0.01) (1)	15.41 (0.43) (6)	12.56 (0.01) (2)	12.57 (0.01) (3)	**12.31 (0.09) (0)**
	Spacing	13.23 (12.33) (6)	422.58 (557.38) (7)	8.98 (0.00) (5)	5.46 (0.01) (3)	7.49 (0.50) (4)	5.45 (0.01) (2)	5.34 (0.09) (1)	**1.93 (1.43) (0)**
	Sparsity	5.86 (5.81) (6)	5433.68 (8409.98) (7)	2.42 (0.00) (1)	2.43 (0.01) (4)	2.70 (0.17) (5)	2.43 (0.00) (3)	2.43 (0.01) (2)	**1.70 (0.09) (0)**
	Uniform	0.23 (0.28) (7)	1.20 (0.75) (5)	0.97 (0.00) (6)	2.44 (0.00) (1)	1.48 (0.24) (4)	2.44 (0.00) (3)	2.44 (0.00) (2)	**2.72 (0.54) (0)**
	SUniform	-1.27 (0.68) (7)	0.02 (0.57) (5)	-0.04 (0.00) (6)	0.95 (0.00) (3)	0.17 (0.09) (4)	0.95 (0.00) (2)	0.96 (0.01) (1)	**1.13 (0.12) (0)**
	Fill Distance	7.83 (2.94) (7)	3.84 (0.55) (6)	3.54 (0.00) (5)	2.59 (0.00) (1)	3.33 (0.23) (4)	2.59 (0.00) (3)	2.59 (0.00) (2)	**2.39 (0.10) (0)**
DTLZ4	HV	0.96 (0.12) (7)	1.06 (0.01) (6)	1.08 (0.00) (1)	1.06 (0.00) (4)	**1.08 (0.00) (0)**	1.06 (0.00) (3)	1.06 (0.00) (5)	1.07 (0.00) (2)
	IGD	19.84 (15.30) (7)	12.78 (0.16) (4)	15.14 (0.00) (5)	12.54 (0.00) (1)	15.48 (0.30) (6)	12.55 (0.00) (2)	12.60 (0.10) (3)	**12.19 (0.07) (0)**
	Spacing	5.44 (0.81) (3)	2.78 (0.34) (1)	8.97 (0.00) (7)	5.45 (0.00) (5)	7.67 (0.60) (6)	5.45 (0.00) (4)	5.36 (0.17) (2)	**0.83 (0.54) (0)**
	Sparsity	**1.55 (0.22) (0)**	2.20 (0.18) (2)	2.42 (0.00) (3)	2.43 (0.00) (4)	2.81 (0.13) (7)	2.43 (0.00) (5)	2.45 (0.05) (6)	1.77 (0.11) (1)
	Uniform	1.26 (0.64) (6)	2.55 (0.09) (1)	0.96 (0.00) (7)	2.43 (0.00) (2)	1.38 (0.16) (5)	2.43 (0.00) (4)	2.43 (0.00) (3)	**3.07 (0.15) (0)**
	SUniform	0.06 (0.78) (6)	1.00 (0.06) (1)	-0.04 (0.00) (7)	0.95 (0.00) (4)	0.15 (0.08) (5)	0.95 (0.00) (3)	0.97 (0.05) (2)	**1.18 (0.02) (0)**
	Fill Distance	5.13 (4.21) (7)	2.83 (0.24) (4)	3.54 (0.00) (6)	2.59 (0.00) (1)	3.41 (0.23) (5)	2.59 (0.00) (3)	2.59 (0.00) (2)	**2.41 (0.14) (0)**
RE37	HV	0.98 (0.07) (6)	1.03 (0.02) (3)	**1.10 (0.00) (0)**	1.01 (0.01) (5)	1.07 (0.00) (1)	0.98 (0.00) (7)	1.02 (0.02) (4)	1.07 (0.01) (2)
	IGD	17.20 (4.58) (7)	12.24 (0.82) (2)	12.17 (0.00) (1)	13.31 (0.27) (4)	12.86 (0.11) (3)	16.54 (0.05) (6)	14.15 (1.35) (5)	**10.74 (0.20) (0)**
	Spacing	7.47 (2.16) (4)	5.69 (1.56) (2)	5.74 (0.01) (3)	11.92 (1.21) (7)	**4.89 (1.90) (0)**	8.30 (0.47) (6)	8.22 (0.95) (5)	5.25 (1.32) (1)
	Sparsity	1.05 (0.50) (1)	1.52 (0.31) (4)	**0.86 (0.01) (0)**	1.74 (0.16) (5)	1.45 (0.10) (3)	3.08 (0.05) (7)	2.47 (0.40) (6)	1.39 (0.07) (2)
	Uniform	0.23 (0.30) (4)	**1.17 (0.56) (0)**	0.59 (0.00) (3)	0.01 (0.02) (6)	0.99 (0.09) (1)	0.00 (0.00) (7)	0.04 (0.05) (5)	0.94 (0.53) (2)
	SUniform	-1.04 (0.38) (5)	**0.08 (0.33) (0)**	-0.66 (0.00) (3)	-0.94 (0.12) (4)	-0.37 (0.04) (2)	-1.55 (0.01) (7)	-1.19 (0.28) (6)	0.07 (0.24) (1)
	Fill Distance	5.33 (1.46) (7)	3.36 (0.93) (1)	4.56 (0.00) (4)	3.44 (0.34) (2)	3.73 (0.18) (3)	4.82 (0.04) (6)	4.80 (0.03) (5)	**2.93 (0.32) (0)**
Rank	HV	6.8	4.6	1.8	4.4	**0.2**	3.6	4.2	2.4
	IGD	6	3	4.6	1.8	4.6	3.4	4	**0.6**
	Spacing	4.4	2.8	5.6	3.6	4	3.4	3.6	**0.6**
	Sparsity	**1.4**	3.6	1.6	4.6	4.8	5.6	4.6	1.8
	Uniform	5.8	2	5.8	2.8	4	3.4	3.4	**0.8**
	SUniform	6	2	5.8	3.4	4.4	3	3.2	**0.2**
	Fill Distance	6.6	4.2	5.2	1	4.2	3.2	3.2	**0.4**

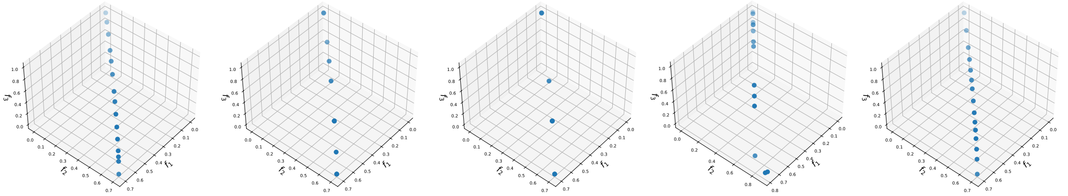
Table 9: Numerical results on four-objective problems.

	Indicator	DEA-GNG	LMPFE	Subset	NSGA3	SMS-EMOA	MOEA/D	MOEA/D-AWA	UMOD
RE41	HV	0.63 (0.16) (7)	1.07 (0.01) (3)	**1.14 (0.00) (0)**	1.08 (0.00) (2)	1.00 (0.00) (5)	1.00 (0.00) (6)	1.02 (0.00) (4)	1.11 (0.00) (1)
	IGD	41.21 (12.46) (7)	17.56 (1.51) (2)	15.15 (0.06) (1)	19.47 (0.76) (3)	19.57 (1.34) (4)	24.95 (0.26) (6)	23.96 (0.67) (5)	**14.36 (0.34) (0)**
	Spacing	4.75 (0.94) (1)	**4.11 (0.74) (0)**	7.45 (0.07) (2)	10.81 (0.36) (7)	9.67 (0.04) (5)	9.25 (0.04) (4)	9.86 (1.66) (6)	8.16 (0.49) (3)
	Sparsity	**0.40 (0.11) (0)**	1.06 (0.09) (4)	0.75 (0.00) (1)	0.99 (0.03) (3)	1.28 (0.19) (5)	2.24 (0.06) (7)	2.03 (0.00) (6)	0.77 (0.01) (2)
	Uniform	0.07 (0.07) (4)	**2.04 (0.01) (0)**	1.10 (0.02) (1)	0.01 (0.01) (5)	0.16 (0.08) (3)	0.00 (0.00) (7)	0.00 (0.00) (6)	0.54 (0.34) (2)
	SUniform	-1.79 (0.32) (5)	**0.41 (0.03) (0)**	-0.24 (0.02) (1)	-1.21 (0.04) (3)	-1.31 (0.15) (4)	-2.00 (0.02) (7)	-1.88 (0.08) (6)	-0.41 (0.17) (2)
	Fill Distance	11.41 (0.12) (7)	5.74 (1.07) (5)	**3.39 (0.00) (0)**	5.27 (0.74) (3)	4.28 (0.52) (2)	5.71 (0.07) (4)	5.88 (0.02) (6)	4.15 (0.02) (1)
RE42	HV	0.55 (0.02) (3)	0.60 (0.01) (2)	0.61 (0.00) (1)	0.52 (0.02) (5)	0.55 (0.02) (4)	0.44 (0.00) (7)	0.51 (0.04) (6)	**0.62 (0.01) (0)**
	IGD	28.16 (8.45) (6)	23.75 (3.02) (3)	25.33 (0.09) (4)	21.01 (2.74) (0)	22.33 (2.39) (1)	30.31 (0.70) (7)	26.16 (0.62) (5)	23.73 (1.05) (2)
	Spacing	10.82 (5.18) (5)	8.51 (9.05) (3)	12.28 (0.06) (7)	11.81 (10.46) (6)	8.31 (1.18) (2)	**6.52 (0.47) (0)**	9.48 (1.32) (4)	8.01 (0.64) (1)
	Sparsity	5.96 (7.42) (7)	2.43 (3.44) (4)	2.00 (0.01) (3)	3.86 (4.58) (6)	1.18 (0.28) (1)	3.45 (0.27) (5)	1.96 (0.51) (2)	**0.77 (0.04) (0)**
	Uniform	0.03 (0.02) (4)	**1.03 (0.34) (0)**	0.00 (0.00) (6)	0.04 (0.03) (3)	0.05 (0.05) (2)	0.00 (0.00) (7)	0.00 (0.00) (5)	0.07 (0.05) (1)
	SUniform	-1.47 (0.12) (4)	**-0.61 (0.18) (0)**	-2.53 (0.00) (7)	-1.45 (0.12) (3)	-1.31 (0.18) (2)	-2.28 (0.04) (6)	-2.12 (0.16) (5)	-1.25 (0.14) (1)
	MaxGD	29.05 (8.88) (1)	35.21 (5.92) (5)	35.59 (0.01) (6)	**28.04 (7.95) (0)**	31.62 (2.71) (2)	34.41 (1.41) (3)	34.75 (2.23) (4)	37.36 (0.61) (7)



(a) UMOD (b) AWA (c) MOEA/D (d) NSGA3 (e) SMS-EMOA

Figure 10: Comparison of using different methods on DTLZ5.



(a) UMOD (b) AWA (c) MOEA/D (d) NSGA3 (e) SMS-EMOA

Figure 11: Comparison of using different methods on DTLZ6.

B.6 Results on DTLZ5 and DTLZ6

The PFs of DTLZ5 and DTLZ6 are identical, which are a degenerate 1-dimensional hyper-curve within the three-objective space. The visualization results are shown in Figures 10 and 11 and numerical results are shown in Table 10.

Table 10: Numerical results on DTLZ5 and DTLZ6 problems.

	Spacing	Sparsity	HV	Uniform	Smooth Uniform	IGD	FD
UMOD	**0.0073**	**0.0131**	0.6995	**0.0867**	**-0.0634**	**0.0294**	**0.085**
MOEAD	0.0871	0.0645	0.6352	0	-0.1978	0.143	0.3002
AWA	0.0909	0.0312	0.6697	0	-0.1802	0.0712	0.1697
SMS-MOEA	0.0454	0.0147	**0.7035**	0.0783	-0.0753	0.0307	0.1099
NSGA3	0.0599	0.0289	0.6623	0.0005	-0.1417	0.0683	0.1802
UMOD	**0.0125**	**0.0128**	0.7011	**0.0738**	**-0.0618**	**0.0285**	**0.0731**
MOEAD	0.0843	0.0599	0.6352	0	-0.2032	0.143	0.3002
AWA	0.0908	0.0312	0.6697	0	-0.1694	0.0712	0.1697
SMS-MOEA	0.0426	0.0143	**0.7036**	0.072	-0.0752	0.0305	0.1099
NSGA3	0.0524	0.0415	0.6623	0	-0.1357	0.0931	0.2972

Table 10 highlights that UMOD significantly outperforms other methods in ensuring evenly distributed solutions. In the decomposition-based framework, for such degenerated problems, different preferences may correspond to the same Pareto objective. The reason behind generating duplicate solutions is explained in Appendix C.5. And therefore, MOEA/D and MOEA/D-AWA tend to produce duplicate solutions. NSGA3 is also found to produce duplicate solutions easily. SMS-MOEA avoids duplicate solutions by maximizing the hypervolume. SMS-MOEA surpasses UMOD in hypervolume, but UMOD considerably outperforms SMS-MOEA in IGD and FD, which are of interests in this paper.

C Method details

C.1 Practical algorithms

In this section, we present practical algorithms to solve the maximal packing problem in the Pareto front (Equation (10)). We start by generating an initial uniform distribution of preference vectors. Then, we use either multiobjective evolutionary algorithms (MOEAs) or gradient-based MOO to solve for the preference angle and Pareto objective pairs. MOEAs are suitable for problems with local optimas, while gradient-based MOO is efficient for neural network problems with millions of decision variables. Next, we fit a Pareto front model to learn the expression of $\boldsymbol{h}_\phi$ and re-determine the preference angles by maximizing the pairwise distances. These two steps are repeated alternately.

Algorithm 1 Uniform Multiobjective Optimization (UMOD)

```
Input: Initial $K$ uniform preferences $\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(K)}\}$ by Das-Dennis method [12]. Initial solutions $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}\}$.
for $n = 1$ to $N$ do
    Run MOEA/D with mTche aggregation function or gradient-based MOO using $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}\}$ as initial solutions under preferences $\{\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(K)}\}$.
    Train a model $\boldsymbol{h}_\phi$ to predict Pareto objectives by the preference angles using mean square estimation with angle-objective pairs $(\boldsymbol{\vartheta}^{(i)}, \boldsymbol{y}^{(i)})$.
    Update $(\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)})$ by Algorithm 2.
    Recalculate preference vectors, $\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(K)}$.
    Update the initial solutions $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}\}$ by MOEA/D or gradient-based mTche using the last generation of solutions as a warm start.
end for
```

Algorithm 2 Recalculate Preference Angles (ALG_Update)

```
Input: The initial configuration $\{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}\}$ and $\boldsymbol{h}_\phi$.
for $i = 1$ to $N_{\text{opt}}$ do
    Calculate the indexes for the minimal pairwise objectives:
        $(i^*, j^*) = \arg \min_{1 \le i < j \le K} \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i)}, \boldsymbol{\vartheta}^{(j)}).$
    Update the positions of $(\boldsymbol{\vartheta}^{(i^*)}, \boldsymbol{\vartheta}^{(j^*)})$:
        $\begin{cases} \boldsymbol{\vartheta}^{(i^*)} \leftarrow \operatorname{clip}\left(\boldsymbol{\vartheta}^{(i^*)} + \eta \frac{\partial \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i^*)})}{\partial \boldsymbol{\vartheta}^{(i^*)}}) A_\phi, 0, \frac{\pi}{2}\right) \\ \boldsymbol{\vartheta}^{(j^*)} \leftarrow \operatorname{clip}\left(\boldsymbol{\vartheta}^{(j^*)} - \eta \frac{\partial \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j^*)})}{\partial \boldsymbol{\vartheta}^{(j^*)}} A_\phi, 0, \frac{\pi}{2}\right) \end{cases},$
    where $A_\phi = \frac{\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i^*)}) - \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j^*)})}{\rho(\boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(i^*)}), \boldsymbol{h}_\phi(\boldsymbol{\vartheta}^{(j^*)}))}^{\top}$.
end for
Output: The updated preference angles $\{\boldsymbol{\vartheta}^{(1)}, \ldots, \boldsymbol{\vartheta}^{(K)}\}$.
```

C.2 Problem formulations

For completeness, ZDT1, ZDT2, and DTLZ1 problems are described as follows.

ZDT1.

$$\begin{cases} f_1(\boldsymbol{x}) = x_1, \\ f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(f_1(\boldsymbol{x}), g(\boldsymbol{x})), \\ g(\boldsymbol{x}) = 1 + \dfrac{9}{n-1} \displaystyle\sum_{i=2}^{n} x_i, \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) = 1 - \sqrt{f_1(\boldsymbol{x})/g(\boldsymbol{x})}, \\ 0 \le x_i \le 1, \qquad i \in [n]. \end{cases} \tag{19}$$

The PF of ZDT1 is $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1$.

ZDT2.

$$\begin{cases} f_1(\boldsymbol{x}) = x_1, \\ f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(f_1(\boldsymbol{x}), g(\boldsymbol{x})), \\ g(\boldsymbol{x}) = 1 + \dfrac{9}{n-1} \displaystyle\sum_{i=2}^{n} x_i, \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) = 1 - f_1(\boldsymbol{x})/g(\boldsymbol{x})^2, \\ 0 \le x_i \le 1, \qquad i \in [n]. \end{cases} \tag{20}$$

The PF of ZDT2 is $f_2 = 1 - f_1^2, 0 \le f_1 \le 1$.

DTLZ1.

$$\begin{cases} f_1(\boldsymbol{x}) = \dfrac{1}{2} x_1 x_2 (1 + g(\boldsymbol{x})), \\ f_2(\boldsymbol{x}) = \dfrac{1}{2} x_1 (1 - x_2)(1 + g(\boldsymbol{x})), \\ f_3(\boldsymbol{x}) = \dfrac{1}{2} (1 - x_1)(1 + g(\boldsymbol{x})), \\ g(\boldsymbol{x}) = 100((n-2) + \displaystyle\sum_{i=3}^{n} (x_i - 0.5)^2 + \sum_{i=3}^{n} \cos(20\pi(x_i - 0.5))), \\ 0 \le x_i \le 1, \qquad i \in [n]. \end{cases} \tag{21}$$

The PF of DTLZ1 is $0.5\Delta_3$ (3-dim simplex).

C.3 Conversion between a preference and a preference angle

The preference vector $\boldsymbol{\lambda}$ and preference angles $\boldsymbol{\vartheta}$ are easily inter-convertible via the following equations. This one-to-one, differentiable mapping allows conversion between $\boldsymbol{\vartheta}$ and $\boldsymbol{\lambda}$. While $\boldsymbol{\lambda}$ belongs to the m-D simplex, $\boldsymbol{\vartheta}$ lies within the box constraint $[0, \frac{\pi}{2}]^{m-1}$. For optimization purposes, $\boldsymbol{\vartheta}$ is more manageable because it can be easily projected onto $[0, \frac{\pi}{2}]^{m-1}$, whereas projecting $\boldsymbol{\lambda}$ onto the m-D simplex is more complex.

$$\begin{cases} \vartheta_1 = \arg\cos(\sqrt{\lambda_1}), \\ \vartheta_2 = \arg\cos\left(\dfrac{\sqrt{\lambda_2}}{\sin\vartheta_1}\right), \\ \vartheta_3 = \arg\cos\left(\dfrac{\sqrt{\lambda_3}}{\sin\vartheta_1 \sin\vartheta_2}\right), \\ \quad\vdots \\ \vartheta_{m-1} = \arg\cos\left(\dfrac{\sqrt{\lambda_{m-1}}}{\prod_{i=1}^{m-2} \sin\vartheta_i}\right), \end{cases} \qquad \begin{cases} \lambda_2 = \sin^2(\vartheta_1)\cos^2(\vartheta_2), \\ \lambda_3 = \sin^2(\vartheta_1)\sin^2(\vartheta_2)\cos^2(\vartheta_3), \\ \quad\vdots \\ \lambda_m = \displaystyle\prod_{i=1}^{m-1} \sin^2(\vartheta_i). \end{cases} \tag{22}$$

C.4 Baseline methods used in fairness classification problem

This subsection describes baseline gradient methods for fairness classification problems (Section 5.2). We introduce three aggregation functions:

1. **Agg-LS** (Linear Scalarization):

$$g_{\boldsymbol{\lambda}}^{\mathrm{LS}}(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}).$$

2. **Agg-Tche** (Tchebycheff):

$$g_{\boldsymbol{\lambda}}^{\mathrm{Tche}}(\boldsymbol{x}) = \max_{i \in [m]} \{\lambda_i (f_i(\boldsymbol{x}) - z_i)\},$$

where $\boldsymbol{z}$ is a reference point (e.g., ideal point) which dominates the entire PF, i.e., $\boldsymbol{z} \preceq \boldsymbol{y}$, for all $\boldsymbol{y} \in \mathcal{T}$. For the modified Tchebycheff (**Agg-mTche**) function, the modification involves replacing λ_i with $\frac{1}{\lambda_i}$ in the equation above.

3. **Agg-PBI** (Penalty-Based Intersection):

$$g_{\boldsymbol{\lambda}}^{\mathrm{PBI}}(\boldsymbol{x}) = d_1 + \mu d_2 = \frac{\left\|(\boldsymbol{z} - \boldsymbol{f}(\boldsymbol{x}))^\top \boldsymbol{\lambda}\right\|}{\|\boldsymbol{\lambda}\|} + \mu \left\|\boldsymbol{f}(\boldsymbol{x}) - (\boldsymbol{z} - d_1 \boldsymbol{\lambda})\right\|,$$

where $\boldsymbol{z}$ is the same reference point as introduced. For gradient-based multiobjective optimization methods, solution $\boldsymbol{x}$ is updated as $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \frac{\partial g_{\boldsymbol{\lambda}}(\boldsymbol{x})}{\partial \boldsymbol{x}}$, where η is a small positive learning rate, and the upperscript agg denotes PBI, LS, or Tche. We also would like to briefly introduce the other three gradient based method, PMGDA (Preference-based Multiple Gradient Descent Algorithm) [59], EPO (Exact Pareto optimization) [36], and HVGrad (Gradient-based HV maximization method) [17]. PMGDA and EPO employ gradient-based techniques to precisely identify Pareto solutions at the intersection points where the Pareto objectives converge with the PF. On the other hand, HVGrad leverages gradient ascent on the hypervolume to maximize the hypervolume metric, thereby optimizing the overall dominance of the solution set.

C.5 Duplicated solutions issues caused by the mTche aggregation function

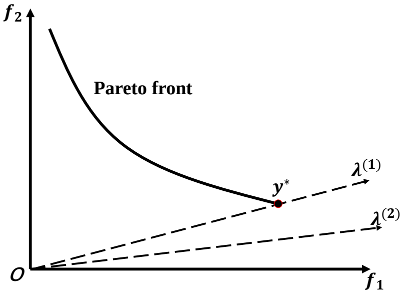


Figure 12: Duplicated solutions generated by Agg-mTche. Different preference vectors $\boldsymbol{\lambda}^{(1)}$ and $\boldsymbol{\lambda}^{(2)}$ correspond to the same optimal objective vector $\boldsymbol{y}^*$.

In this section, we discuss when Tchebycheff aggregation methods yield duplicated Pareto objectives. Appendix C.5 illustrates this, showing a preference vector $\boldsymbol{\lambda}^{(1)}$ intersecting the PF at the optimal solution $\boldsymbol{y}^*$, where

$$\frac{y_i^*}{\lambda_i} = \ldots = \frac{y_m^*}{\lambda_m}. \tag{23}$$

If the preference vector does not intersect the Pareto front ($\boldsymbol{\lambda}^{(2)}$), $\boldsymbol{y}^*$ is the Pareto front endpoint with the highest Tchebycheff value. In Appendix C.5, $\boldsymbol{y}^*$ is the optimal value, so preferences $\boldsymbol{\lambda}^{(1)}$ and $\boldsymbol{\lambda}^{(2)}$ correspond to the duplicated Pareto objective $\boldsymbol{y}^*$.

D Miscellanies

D.1 Broader impacts

By optimizing multiple conflicting objectives, these algorithms enhance decision-making in fields such as healthcare, product design, and trustworthy machine learning. In product design, multiobjective optimization balances trade-offs between capacity and cost, facilitating the effective release of products that represent the entire Pareto front. In trustworthy machine learning, the UMOD method designs a series of classifiers that balance fairness and accuracy, improving our understanding of different Pareto-optimal classifiers.

UMOD is a foundational algorithm, and its broader impact depends on its downstream applications. We believe UMOD itself does not have a direct negative social impact.

D.2 Limitations

While we have illustrated UMOD's success, we acknowledge some limitations. First, UMOD does not address disconnected Pareto fronts such as DTLZ7, as Theorems 3 and 4 assume a connected Pareto front to ensure uniform distribution. Second, we have not considered problems with discrete, binary decision variables, or constrained MOPs, as our approach requires a continuous Pareto front. Adapting UMOD to these complex MOO problems is our next goal. Finally, UMOD requires a neural model to estimate the Pareto front shape. Although training and preference updating with neural networks are efficient, this adds extra operations compared to traditional multiobjective algorithms, which brings inconvenience.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: We explicitly enumerate our contributions in the end of the introduction part.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We list the limitations in the last section (Section 6).

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The assumptions are included in theorems and the full proofs are provided in Appendices A.1 to A.3 respectively.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide experiment details in Appendix B.3.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] .

   Justification: The model implementation and the code are attached as supplementary materials. Source codes have been open to the public.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] .

   Justification: The details experiment settings are provided in Appendix B.3.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] .

   Justification: All numerical results are averaged on 31 random seeds.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: Details are provided in Section 5 (Experiment settings).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes] .

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] .

    Social impact is discussed in Appendix D.1.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper is not related to LLMs and image generators.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] .

    Justification: Licenses are provided in Appendix B.3.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] .

    Justification: The paper proposes a new model, and we choose the CC BY 4.0 license.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: This paper is not related with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: The proposed method is a fundamental algorithm and is not directly related to human subjects.