# Batched Data-Driven Evolutionary Multi-Objective Optimization Based on Manifold Interpolation*†

Ke Li[1] and Renzhi Chen[2]

[1]Department of Computer Science, University of Exeter, EX4 4QF, Exeter, UK
[2]PLA Academy of Military Science, Beijing, China
*Email: k.li@exeter.ac.uk

**Abstract:** Multi-objective optimization problems are ubiquitous in real-world science, engineering and design optimization problems. It is not uncommon that the objective functions are as a black box, the evaluation of which usually involve time-consuming and/or costly physical experiments. Data-driven evolutionary optimization can be used to search for a set of non-dominated trade-off solutions, where the expensive objective functions are approximated as a surrogate model. In this paper, we propose a framework for implementing batched data-driven evolutionary multi-objective optimization. It is so general that any off-the-shelf evolutionary multi-objective optimization algorithms can be applied in a plug-in manner. In particular, it has two unique components: 1) based on the Karush-Kuhn-Tucker conditions, a manifold interpolation approach that explores more diversified solutions with a convergence guarantee along the manifold of the approximated Pareto-optimal set; and 2) a batch recommendation approach that reduces the computational time of the optimization process by evaluating multiple samples at a time in parallel. Experiments on 136 benchmark test problem instances with irregular Pareto-optimal front shapes against six state-of-the-art surrogate-assisted EMO algorithms fully demonstrate the effectiveness and superiority of our proposed framework. In particular, our proposed framework is featured with a faster convergence and a stronger resilience to various PF shapes.

**Keywords:** Multi-objective optimization, surrogate modeling, Karush–Kuhn–Tucker conditions, evolutionary algorithm

## 1 Introduction

Real-world problems in science, engineering and design often involve multiple conflicting objectives, as known as multi-objective optimization problems (MOPs). For example, in the optimal design of a water distribution system, many indicators need to be considered to achieve a trade-off between capital and/or operational cost and performance type benefits such as pressure deficit, reliable configurations under abnormal conditions and network resilience index. There does not exist a global optimal solution that optimizes all conflicting objectives. Instead, multi-objective optimization mainly aims to find a set of trade-off alternatives that compromise among different objectives before being handed over for multi-criterion decision-making.

Evolutionary algorithms (EAs) have been widely recognized as a major approach for multi-objective optimization given its population-based property for approximating a set of non-dominated solutions in a single run [1]. Over the past three decades and beyond, many efforts have been dedicated to the developments of evolutionary multi-objective optimization (EMO) algorithms. According to their environmental selection mechanisms, the existing EMO algorithms can be classified into three categorizes: 1) dominance-based methods, e.g., elitist non-dominated sorting genetic algorithm (NSGA-II) [2], 2) indicator-based methods, indicator-based EA (IBEA) [3], and 3) decomposition-based methods, e.g., multi-objective EA based on decomposition (MOEA/D) [4].

*Both authors made equal contributions to this paper.
†This manuscript is accepted for publication in the IEEE Transactions on Evolutionary Computation.

In practice, it is not uncommon that the objective functions of real-world problems are as a black box and are expensive to evaluate, either computationally or economically. For example, computational fluid dynamic simulations can take from minutes to hours to carry out a single function evaluation (FE) [5]. Due to the population-based and iterative nature, EAs usually require a vast amount of FEs to obtain reasonably acceptable solutions. This is unrealistic when FEs are expensive thus it significantly hinders a wider uptake of EAs in real-world scenarios. To mitigate this issue, surrogate models, built by data collected from expensive FEs, have emerged as a powerful tool to assist EAs for solving expensive optimization problems, also known as data-driven evolutionary optimization[1] [7]. It consists of two intertwined design components [6, 7].

- The first one is the surrogate modeling of the expensive objective functions. Many off-the-shelf machine learning approaches, e.g., support vector machine (SVM) [8], Gaussian process regression (GPR) or Kriging model [9–11] and radial basis function networks (RBFN) [12–14], can serve this purpose.
- The other one is the model management that mainly aims to select promising solution(s) output from a surrogate-assisted search process for conducting expensive FEs. In particular, such search process can either be driven by the surrogate objective functions directly or combined with the uncertainty inferred from the model, also known as the acquisition function, e.g., expected improvement [15], upper confidence bound [16] and probability of improvement [17], in GPR-assisted EAs [18]. Afterwards, the newly evaluated solutions will thus be used to update the surrogate model accordingly.

In practice, many physical experiments can be carried out in parallel given the availability of more than one infrastructure. For example, laboratory technicians often perform experiments with duplicated setups in parallel to mitigate empirical bias. Likewise, in automated machine learning, training and validating machine learning models are usually distributed into multiple cores or GPUs for hyper-parameter optimization. An effective parallelization, also known as batch recommendation in data-driven evolutionary optimization, are of practical importance to significantly save the computational time by reducing the number of iterations. However, this line of research is relatively lukewarm in the data-driven evolutionary optimization community [10, 19, 20].

As discussed in [21], many prevalent test problems (e.g., DTLZ1 to DTLZ4 [22], WFG4 to WFG9 [23] and LSMOP1 to LSMOP8 [24]) have similar 'regular' Pareto-optimal fronts (PFs) characterized as a triangle defined as: $\sum_{i=1}^{p} f_i^p(\mathbf{x}) = 1$ where $0 \leq f_i(\mathbf{x}) \leq 1$ and $i \in \{1, \cdots, m\}$. $p = 1$ in DTLZ1 and LSMOP1 to LSMOP4, $p = 2$ in DTLZ2 to DTLZ4, WFG4 to WFG9 and LSMOP5 to LSMOP8. Unfortunately, it is unrealistic to have such regular PF in real-world MOPs [21]. On the contrary, due to the complex and non-linear relationship between objectives, it is not uncommon to have an irregular PF featured as disconnected, incomplete, degenerated, and/or badly-scaled. Although there have been growing interests for handling MOPs with irregular PFs in the EMO community (e.g., [25–27]), it has rarely been considered in the context of data-driven EMO, except for [28].

As discussed in [29], under certain smoothness assumptions, it can be induced from the Karush–Kuhn–Tucker (KKT) conditions [30] that the Pareto-optimal set (PS) is a piecewise continuous $(m-1)$-dimensional manifold in the decision space [31] where $m$ is the number of objectives. Such manifold property can be used to interpolate new Pareto-optimal solutions such as [32, 33]. However, using KKT conditions needs the access of the analytical information of the objective functions, which is hardly accessible for black-box multi-objective optimization scenarios.

To address the above issues, this paper proposes a batched data-driven EMO framework based on manifold interpolation for solving expensive MOPs with various PF shapes. It consists of the following four major design components.

- `Surrogate modeling`: For proof-of-concept purposes, GPR and RBFN are respectively used to build the surrogate model for each computationally expensive objective function in this paper.

[1] It is called surrogate-assisted EA interchangeably [6] in the literature.

- `Evolutionary search`: This step searches for an approximated PF based on the surrogate objective functions. In particular, any existing EMO algorithm can be used to serve this purpose where we use NSGA-II, IBEA and MOEA/D for proof-of-concept purposes.
- `Manifold interpolation`: Based on the KKT conditions, this step is designed to interpolate new candidate solutions along the manifold of the approximated surrogate Pareto-optimal set with regard to the non-dominated solutions obtained in the `evolutionary search` step. Note that since the surrogate models considered in this paper are continuously differentiable, we can naturally derive their first- and second-order derivatives of the predicted objective functions.
- `Batch recommendation`: Two types of simple and effective batch recommendation mechanisms are proposed to pick up multiple candidate solutions from the non-dominated solutions obtained in the `manifold interpolation` step for expensive FEs. In particular, one is directly derived from the native environmental selection mechanism of the EMO algorithm used in the `evolutionary search` step while the other is based on the individual Hypervolume contribution.

In experiments, we generate 12 algorithm instantiations of our proposed framework based on the combinations of two surrogate models, three baseline EMO algorithms and two batch recommendation mechanisms. Extensive experiments on 168 benchmark test problem instances with both regular and irregular PFs and a real-world application on hyper-parameter optimization fully demonstrate the effectiveness and superiority of our proposed algorithms against 7 state-of-the-art data-driven EMO algorithms. In particular, our proposed framework is featured with a faster convergence and a stronger resilience to various PF shapes.

The rest of this paper is organized as follows. Section 2 first gives some essential preliminaries including definitions pertinent to this paper along with a pragmatic overview of the existing developments in data-driven EMO. Section 3 delineates the technical details of our proposed framework step by step. The experimental results are presented and analyzed in Section 5. At the end, Section 6 concludes this paper and sheds some lights on potential future directions.

## 2 Preliminaries

In this section, we first give some basic concepts pertinent to this paper. Thereafter, we briefly overview some selected developments of data-driven EMO.

### 2.1 Basic Definitions in Multi-Objective Optimization

The MOP considered in this paper is defined as:

$$\begin{aligned} &\text{minimize} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned}, \tag{1}$$

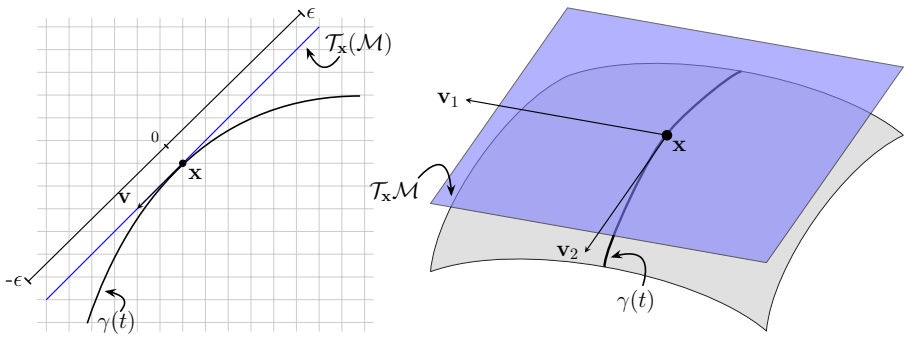
where $\mathbf{x} = (x_1, \cdots, x_n)^T$ is a decision vector and $\mathbf{F}(\mathbf{x})$ is an objective vector. $\Omega = [x_i^L, x_i^U]_{i=1}^n \subseteq \mathbb{R}^n$ defines the search space. $\mathbf{F} : \Omega \to \mathbb{R}^m$ is the corresponding attainable set in the objective space $\mathbb{R}^m$.

**Definition 1.** *Given two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \Omega$, $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$, denoted as $\mathbf{x}^1 \preceq \mathbf{x}^2$, if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i \in \{1, \cdots, m\}$ and $\mathbf{F}(\mathbf{x}^1) \neq \mathbf{F}(\mathbf{x}^2)$.*

**Definition 2.** *A solution $\mathbf{x}^* \in \Omega$ is said to be Pareto-optimal if and only if $\nexists \mathbf{x}' \in \Omega$ such that $\mathbf{x}' \preceq \mathbf{x}^*$.*

**Definition 3.** *The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS), i.e., $PS = \{\mathbf{x}^* | \nexists \mathbf{x}' \in \Omega$ such that $\mathbf{x}' \preceq \mathbf{x}^*\}$ and their corresponding objective vectors form the Pareto-optimal front (PF), i.e., $PF = \{\mathbf{F}(\mathbf{x}^*) | \mathbf{x}^* \in PS\}$.*

**Theorem 1** (KKT conditions [30])**.** *Let $\mathbf{x}^*$ be a Pareto-optimal solution of the MOP with $m$ constraints $\{g_i(\mathbf{x}) \leq 0\}_{i=1}^m$ and the set of vectors $\{\nabla g_j(\mathbf{x}^*) | j$ is the index of an active constraint$\}$ are*




One-dimensional manifold Two-dimensional manifold

Figure 1: A conceptual illustration of the tangent vector(s) $\mathbf{v}$ of a point $\mathbf{x}$ on a manifold along with its corresponding tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$.

*linearly independent. There exists a vector* $\mu = (\mu_1, \cdots, \mu_m)^T \in \mathbb{R}^m$*, such that:*

$$\begin{aligned} \sum_{i=1}^{m} \alpha_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{k} \lambda_j \nabla g_j(\mathbf{x}^*) = 0 \\ \lambda_j g_j(\mathbf{x}^*)|_{j=1}^{k} = 0 \end{aligned}, \tag{2}$$

*where* $\alpha_i \geq 0$*,* $\forall i \in \{1, \cdots, m\}$ *and* $\sum_{i=1}^{m} \alpha_i = 1$*.*

**Remark 1.** *The objective and constraint functions are assumed to be continuously differentiable in the KKT conditions.*

**Remark 2.** *The MOP (1) considered in this paper does not consider constraints, thus we ignore the* $\sum_{j=1}^{k} \lambda_j \nabla g_j(\mathbf{x}^*)$ *part of equation (2) in the latter derivations.*

**Corollary 1.** *The PF is a* $(m-1)$*-dimensional piecewise continuous manifold under the KKT conditions. For any solution* $\mathbf{x}^*$ *in the PS, there exists an open neighborhood* $\Xi(\mathbf{x}^*)$ *such that the intersection of the PF and* $\{\mathbf{F}(\tilde{\mathbf{x}}) | \tilde{\mathbf{x}} \in \Xi(\mathbf{x}^*)\}$ *is a* $(m-1)$*-dimensional continuously differentiable manifold in* $\mathbb{R}^m$ *[32], so as the PS.*

**Definition 4.** *Let* $\mathcal{M}$ *be a continuously differentiable manifold,* $\gamma : (-\epsilon, \epsilon) \to \mathcal{M}$ *be a continuously differentiable curve on this manifold and it passes through* $\mathbf{x} \in \mathcal{M}$ *where* $\epsilon > 0$*. Use* $t \in (-\epsilon, \epsilon)$ *to parameterize* $\gamma$ *as* $\gamma(t)$ *where* $\gamma(0) = \mathbf{x}$*, the tangent vector of* $\gamma(0)$*, denoted as* $\mathbf{v}$*, is defined as:*

$$\mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}t} f \circ \gamma(t) \Big|_{t=0}, \tag{3}$$

*where* $f \circ \gamma(t) : (-\epsilon, \epsilon) \to \mathcal{M} \to \mathbb{R}$ *is a composite mapping.*

**Definition 5.** *The set of all tangent vectors at* $\mathbf{x}$ *is called the tangent space of* $\mathcal{M}$ *at* $\mathbf{x}$*, denoted as* $\mathcal{T}_{\mathbf{x}}\mathcal{M}$*.*

**Theorem 2.** *Let* $\mathcal{M}$ *be a smooth manifold and* $\mathbf{x} \in \mathcal{M}$*, then* $\dim(\mathcal{T}_{\mathbf{x}}\mathcal{M}) = \dim(\mathcal{M})$*, where* $\dim(\cdot)$ *returns the corresponding dimensionality.*

**Remark 3.** *Fig. 1 gives a conceptual illustration of the tangent vector(s)* $\mathbf{v}$ *of a point* $\mathbf{x}$ *on a one- and a two-dimensional manifold, respectively, along with its corresponding tangent space* $\mathcal{T}_{\mathbf{x}}\mathcal{M}$*. In particular, the number of tangent vectors is* $\dim(\mathcal{T}_{\mathbf{x}}\mathcal{M})$*.*

### 2.2 Gaussian Process Regression Model

Given a set of training data $\mathcal{D} = \{(\mathbf{x}^i, f(\mathbf{x}^i)\}_{i=1}^N$, a GPR model aims to learn a latent function $g(\mathbf{x})$ by assuming $f(\mathbf{x}^i) = g(\mathbf{x}^i) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is an independently and identically distributed Gaussian noise. For each testing input vector $\mathbf{z}^* \in \Omega$, the mean and variance of the target $f(\mathbf{z}^*)$ are predicted as [34]:

$$\overline{g}(\mathbf{z}^*) = m(\mathbf{z}^*) + \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}(\mathbf{f} - \mathbf{m}(X)), \tag{4}$$

$$\mathbb{V}[g(\mathbf{z}^*)] = k(\mathbf{z}^*, \mathbf{z}^*) - \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}\mathbf{k}^*, \tag{5}$$

where $X = (\mathbf{x}^1, \cdots, \mathbf{x}^N)^T$ and $\mathbf{f} = (f(\mathbf{x}^1), \cdots, f(\mathbf{x}^N))^T$. $\mathbf{m}(X)$ is the mean vector of $X$, $\mathbf{k}^*$ is the covariance vector between $X$ and $\mathbf{z}^*$, and $K$ is the covariance matrix of $X$. In particular, a covariance function, also known as a kernel, is used to measure the similarity between a pair of data samples $\mathbf{x}$ and $\mathbf{x}' \in \Omega$. This paper uses the Matérn 5/2 kernel without loss of generality and it is defined as:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_n^2 \left(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2}\right) \exp\left(-\frac{\sqrt{5}d}{\rho}\right), \tag{6}$$

where $\rho$ is a positive hyper-parameter of the covariance function and $d = \sqrt{(\mathbf{x} - \mathbf{x}')^T \cdot (\mathbf{x} - \mathbf{x}')}$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$. The predicted mean $\overline{g}(\mathbf{z}^*)$ is directly used as the prediction of $f(\mathbf{z}^*)$, and the predicted variance $\mathbb{V}[g(\mathbf{x}^*)]$ quantifies the uncertainty. As recommended in [34], the hyperparameters are learned by maximizing the log marginal likelihood function defined as:

$$\begin{aligned} \log p(\mathbf{f}|X) = &-\frac{1}{2}\Big(\mathbf{f} - \mathbf{m}(X)\Big)^T (K + \sigma_n^2 I)^{-1}\Big(\mathbf{f} - \mathbf{m}(X)\Big) \\ &- \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{N}{2}\log 2\pi. \end{aligned} \tag{7}$$

In this paper, we assume that the mean function is a constant 0 and the inputs are noiseless.

### 2.3 Radial Basis Function Network

A RBFN is a neural network that uses radial basis functions as activation functions [35]. Here we consider a network with three layers and $k \geq 1$ hidden neurons. To interpolate a function by using a RBFN, it can be represented as the following system of linear equations:

$$\underbrace{\begin{bmatrix} \tilde{g}_{11} & \tilde{g}_{12} & \cdots & \tilde{g}_{1k} \\ \tilde{g}_{21} & \tilde{g}_{22} & \cdots & \tilde{g}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{g}_{N1} & \tilde{g}_{N2} & \cdots & \tilde{g}_{Nk} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}}_{\mathbf{b}}, \tag{8}$$

where $\forall \mathbf{x}^i \in \mathcal{D}$ and $i \in \{1, \cdots, N\}$, we have $b^i = f(\mathbf{x}^i)$. $\tilde{g}_{ij} = \sum_{j=1}^k \exp(-\frac{1}{2s^2}\|\mathbf{x}^i - \mathbf{c}^j\|^2)$ and $\mathbf{c}^j$, $j \in \{1, \cdots, k\}$, is one of the clustering center of $\mathcal{D}$ by using the $k$-means clustering. $s$ and $\mathbf{c}^i$ are hyper-parameters of the model determined by $\mathcal{D}$. The weights $\mathbf{w}^* = (w_1^*, \cdots, w_k^*)^T$ that minimizes the error at the output can be computed with $\mathbf{w}^* = \mathbf{G}^+\mathbf{b}$, i.e., the Moore–Penrose generalized inverse of a matrix [36]. For a testing input $\mathbf{z}^* \in \Omega$, the output of the RBFN is a scalar function of $\mathbf{z}^*$:

$$\overline{g}(\mathbf{z}^*) = \sum_{i=1}^k w_i^* \exp\Big(-\frac{1}{2s^2}\|\mathbf{z}^* - \mathbf{c}^i\|^2\Big). \tag{9}$$

### 2.4 Related Works

This section provides a pragmatic overview of the current developments of data-driven EMO. Interested readers are referred to some excellent survey papers [6, 7, 37].

ParEGO [9] is one of the earliest attempts that extends the classic efficient global optimization (EGO) algorithm to the context of multi-objective optimization. During each iteration, it randomly generates a weight vector to constitute a scalarizing function of the original MOP. It uses a Kriging model to fit a surrogate model of the underlying scalarizing function, based on which an EA is used to search for the next point of merit by optimizing the expected improvement. In [38], Emmerich et al. proposed to use Hypervolume measure as an alternative of scalarizing function to derive a couple of acquisition functions for multi-objective EGO. The similar idea is further exploited in [39] and [40]. Later, Zhang et al. [10] proposed a MOEA/D version of EGO, dubbed MOEA/D-EGO. It applies the GPR to fit a surrogate model for each expensive objective function, based on which they derived the estimated mean and variance of the corresponding scalarizing function. Then, a regular MOEA/D routine is used to search for the approximated PF. In addition, they developed a batch recommendation mechanism to pick up more than one candidate solution for expensive FEs at the end of each iteration. In [11], K-RVEA is proposed for expensive many-objective optimization problems. To tackle the problems with irregular PFs, Habib et al. [28] proposed HSMEA that takes advantages of the interplay of multiple surrogate models and two sets of reference vectors. In addition, it applies a local search to further exploit high quality infill solutions. To have a well balance between exploration and exploitation, Wang et al. proposed to tune the hyperparameters of the acquisition function in EGO according to the search dynamics on the fly [41]. Recently, Song et al. [42] proposed a Kriging-assisted two-archive EA for expensive many-objective optimization, dubbed KTA2. It uses an influential point-insensitive model to approximate each objective function. As for the infill criterion, it proposed an adaptive mechanism to identify the most important requirement on convergence, diversity, or uncertainty to determine an appropriate sampling strategy for re-evaluations using the expensive objective functions.

In addition to EGO, some other machine learning models have also been studied in data-driven EMO. For example, Voutchkov and Keane [43] proposed a simple idea to directly apply a GPR model to replace the expensive objective functions in NSGA-II. At the end of each iteration, the current best candidate solutions in terms of ranking and space filling properties are chosen for conducting expensive FEs. In view of the high computational complexity of GPR, Guo et al. [20] proposed a heterogeneous ensemble model based on least square SVM and RBFN for surrogate modeling. To identify the infill solution(s) for expensive FEs, an ensemble generation method is proposed to quantify the uncertainty of sample points. In [12–14], RBFN are used as the surrogate model to drive the search process. Instead of a regression model, Pan [44] and Zhang et al. [45] proposed to use a classification model to drive the surrogate search routine. Differently, Loshchilov [8] and Seah et al. [46] proposed to fit a surrogate model that predicts the Pareto dominance relation between pairs of solutions.

Different from the above mentioned works, another emerging area is to use transfer learning techniques to boost the search process. For example, Luo et al. [47] proposed to use a multi-task GPR model to build multiple surrogate models simultaneously for different subregions of the PF. In addition, a new infill criterion based on the surrogate landscape is proposed to determine the next candidate solution for conducting the expensive FE. Min et al. [48] proposed to use the transfer stacking technique to jump start the underlying problem-solving routine by leveraging the model built for other related problems. In [49], Yang et al. proposed an EA assisted by two surrogate models. One aims to guide the algorithm to quickly find a promising subregion in the search space and the other model focuses on leveraging good solutions according to the knowledge transferred from the first model.

In the classic multi-objective optimization literature, the KKT conditions have been applied to solve bi-objective design optimization problems [50]. Later, this idea was generalized to MOPs with any number of objectives in theory [32, 51]. It is worth noting that all these approaches are developed upon the assumption that the objective functions are analytically accessible and differentiable. In addition, they only considered a local expansion of an known Pareto-optimal solution. Another line of research is [52] that developed a proximity measure based on KKT optimality theory. This measure was originally designed to evaluate the convergence of a set of non-dominated solutions with regard to the PS. Later, it has also been used as either a driving force or a termination criterion of a local search procedure in NSGA-III [53–55]. As for leveraging the manifold properties of PF and PS, Zhang et al. [29] and Zhou et al. [56] developed a multi-objective estimation of distribution algorithm, dubbed

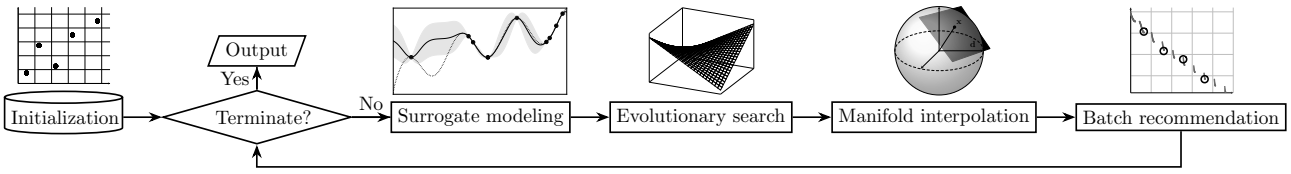



Figure 2: Flowchart of the our proposed batched data-driven EMO framework.

RM-MEDA. It has been one of the pioneers to build a probabilistic model of elite solutions as a $(m-1)$-dimensional manifold [29]. Then, offspring are reproduced by sampling from such model.

# 3 Proposed Algorithm

The flowchart of our proposed batched data-driven EMO framework based on manifold interpolation is shown in Fig. 2. It starts with an `initialization` step based on an experimental design method [57]. In this paper, we use the classic Latin hypercube sampling to serve this purpose without loss of generality. Then, we evaluate the objective function values of these initial samples and store them in the training dataset. During the main loop, the `surrogate modeling` step builds a surrogate model for each expensive objective function based on the up-to-date training dataset. Note that any continuously differentiable regression model can serve this purpose. For a proof-of-concept purpose, this paper applies the GPR and RBFN introduced in Sections 2.2 and 2.3 for surrogate modeling. As for the other three steps, we will delineate their implementations in the following paragraphs.

## 3.1 Evolutionary Search

The `evolutionary search` step aims to approximate the PF based on the surrogate model built in the `surrogate modeling` step. We argue that any existing EMO algorithm can be used as the surrogate optimizer in this step. In particular, we directly use the surrogate model to replace the expensive objective functions in the EMO algorithm. This paper applies three iconic EMO algorithms, i.e., NSGA-II, IBEA and MOEA/D, for a proof-of-concept purpose. For the sake of being self-contained, a description of their working mechanisms can be found in Appendix A.

## 3.2 Manifold Interpolation

After the `evolutionary search` step, we obtain a population of solutions $\mathcal{P}$ that approximate the surrogate PF. Here we assume that these solutions are Pareto-optimal thus they all satisfy the KKT conditions. According to Corollary 1, $\forall \mathbf{x} \in \mathcal{P}$, the PS segment with regard to an open neighborhood $\Xi(\mathbf{x})$ is a $(m-1)$-dimensional manifold, denoted as $\mathcal{M}_{\mathbf{x}}$, so does the corresponding PF segment. The basic idea of this `manifold interpolation` step is to interpolate a set of $\tilde{N} \gg 1$ new candidate solutions $\mathcal{S} = \{\hat{\mathbf{x}}^i\}_{i=1}^{\tilde{N}}$ along the tangent space of $\mathbf{x}$ as:

$$\hat{\mathbf{x}} = \mathbf{x} + \sum_{i=1}^{m-1} \eta_i \mathbf{v}_i, \tag{10}$$

where $\mathbf{v}_i$ is the $i$-th tangent vector at $\mathbf{x}$ and $\eta_i \in (0, 1]$ is a random scaling factor along that direction. In the following paragraphs, we will derive a closed form method to calculate the tangent vectors. To facilitate our derivation, as in Definition 4, we use a parametric form $\mathbf{x}(t)$ where $t \in (-\epsilon, \epsilon)$ to represent each solution on a smooth curve passing through $\mathbf{x}$ on $\mathcal{M}_{\mathbf{x}}$ where $\mathbf{x}(0) = \mathbf{x}$.

According to Corollary 1, we have $\forall \mathbf{x}(t) \in \Xi(\mathbf{x})$ satisfies the KKT conditions. We assume that there exist a time-varying parameter vector $\alpha(t) = (\alpha_1(t), \cdots, \alpha_m(t))^T \in \mathbb{R}^m$, $t \in (-\epsilon, \epsilon)$, such that:

$$\sum_{i=1}^{m} \alpha_i(t) \nabla f_i(\mathbf{x}(t)) = 0, \tag{11}$$

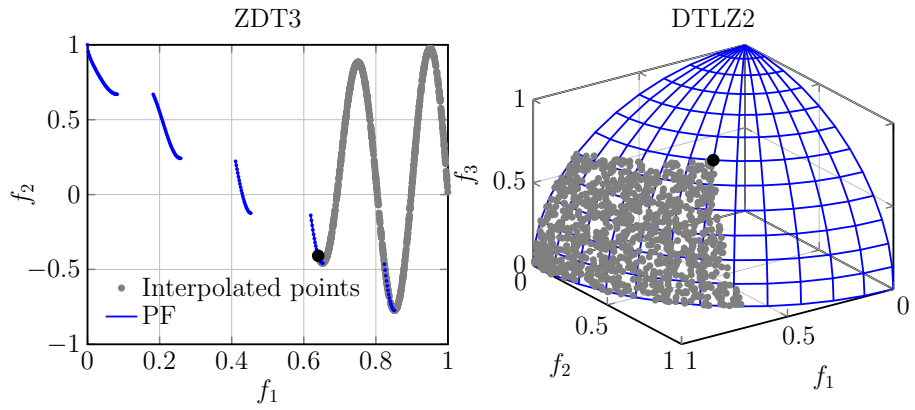



Figure 3: Examples of interpolated solutions (denoted as grey circles) generated by the manifold interpolation.

where $\alpha_i(t) \geq 0$ and $\sum_{i=1}^{m} \alpha_i(t) = 1$. $f_i(\mathbf{x}(t))$ is actually a composite mapping $f_i \circ \mathbf{x}(t) : (-\epsilon, \epsilon) \to \mathcal{M}_{\mathbf{x}} \to \mathbb{R}$ on the manifold as in Definition 4 where $i \in \{1, \cdots, m\}$. By taking the derivatives of equation (11) at $t = 0$, we have:

$$\frac{d}{dt} \sum_{i=1}^{m} \alpha_i(t) \nabla f_i(\mathbf{x}(t)) \Big|_{t=0} = 0,$$

$$\implies \sum_{i=1}^{m} \alpha_i'(0) \nabla f_i(\mathbf{x}(0)) + \Big( \sum_{i=1}^{m} \alpha_i(0) \nabla^2 f_i(\mathbf{x}(0)) \Big) \mathbf{x}'(0) = 0. \tag{12}$$

Given that $\sum_{i=1}^{m} \alpha_i'(t) = 0$, we rewrite equation (12) as a system of linear equations:

$$\begin{bmatrix} \mathbf{1}_{1\times m} & \mathbf{0}_{1\times n} \\ \mathbf{J}^T_{\mathbf{F}(\mathbf{x}(0))} & \mathbf{H}^T_{\mathbf{F}(\mathbf{x}(0))} \cdot \alpha(\mathbf{0}) \end{bmatrix} \begin{bmatrix} \alpha'(0) \\ \mathbf{x}'(0) \end{bmatrix} = 0, \tag{13}$$

where $\mathbf{J}_{\mathbf{F}(\mathbf{x}(0))}$ and $\mathbf{H}_{\mathbf{F}(\mathbf{x}(0))}$ are the $m \times n$ Jacobian matrix and $m \times m \times n$ Hessian tensor of $\mathbf{F}(\mathbf{x}(0)) = (f_1(\mathbf{x}(0)), \cdots, f_m(\mathbf{x}(0)))^T$, respectively. By solving this system of linear equations (13), we obtain $m - 1$ different $\mathbf{x}'(0)$, which constitute the $m - 1$ tangent vectors $\{\mathbf{v}_i\}_{i=1}^{m-1}$ in equation (10).

**Remark 4.** *Let us rewrite equation (12) as follows:*

$$\underbrace{\Big( \sum_{i=1}^{m} \alpha_i(0) \nabla^2 f_i(\mathbf{x}(0)) \Big)}_{\mathbf{H}_{\mathbf{F}(\mathbf{x}(0))} \cdot \alpha(\mathbf{0})} \underbrace{\mathbf{x}'(0)}_{\mathbf{v}} = - \sum_{i=1}^{m} \alpha_i'(0) \nabla f_i(\mathbf{x}(0)). \tag{14}$$

*The left hand side of equation (14) is thus a linear combination of* $\{\nabla f_i(\mathbf{x}(0))\}_{i=1}^{m}$ *that constitute a subspace spanned by them. We take the inverse of* $\mathbf{H}_{\mathbf{F}(\mathbf{x}(0))} \cdot \alpha(0)$ *and further derive equation (14) as:*

$$\mathbf{x}'(0) = \Big[ \mathbf{H}_{\mathbf{F}(\mathbf{x}(0))} \cdot \alpha(0) \Big]^{-1} \Big[ - \sum_{i=1}^{m} \alpha_i'(0) \nabla f_i(\mathbf{x}(0)) \Big]. \tag{15}$$

**Remark 5.** *As shown in equation (10), this* `manifold interpolation` *step implements a random walk along the tangent space of* $\mathbf{x}$*. In principle, the generated solutions constitute a piece-wise linear approximation to the corresponding PS and PF segments within a neighborhood. Fig. 3 gives two examples of manifold interpolation at a given point on the 2-objective ZDT3 and the 3-objective DTLZ2.*

To constitute the Jacobian matrix and the Hessian tensor used in equation (13), we need to access the first- and second-order derivatives of the underlying objective functions. In this paper, since the objective functions are modeled by GPR and RBFN, which are continuously differentiable, we can

naturally derive the first- and second-order derivatives of the predicted mean function with regard to a solution $\mathbf{x}$. Specifically, as for the GPR, we have:

$$\frac{\partial \overline{g}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathbf{k}^*}{\partial \mathbf{x}} K^{-1}\mathbf{f}, \quad \frac{\partial^2 \overline{g}(\mathbf{x})}{\partial \mathbf{x}^2} = \frac{\partial^2 \mathbf{k}^*}{\partial \mathbf{x}^2} K^{-1}\mathbf{f}, \tag{16}$$

where the first- and second-order derivatives of $\mathbf{k}^*$, i.e., the covariance vector between $\mathcal{P}$ and $\mathbf{x}$, are calculated as:

$$\frac{\partial \mathbf{k}^*}{\partial \mathbf{x}} = -\frac{5\mathbf{d}}{3\rho^2}\left(1+\frac{\sqrt{5}\mathbf{d}}{\rho}\right)\sigma_n^2 \exp\left(-\frac{\sqrt{5}\mathbf{d}}{\rho}\right)\frac{\partial \mathbf{d}}{\partial \mathbf{x}}, \tag{17}$$

$$\begin{aligned}\frac{\partial^2 \mathbf{k}^*}{\partial \mathbf{x}^2} = &-\frac{5}{3\rho^2}\left(1+\sqrt{5}\mathbf{d}-\frac{5\mathbf{d}^2}{\rho^2}\right)\sigma_n^2 \exp\left(-\frac{\sqrt{5}\mathbf{d}}{\rho}\right)\left(\frac{\partial \mathbf{d}}{\partial \mathbf{x}}\right)^2 \\ &-\frac{5\mathbf{d}}{3\rho^2}\left(1+\frac{\sqrt{5}\mathbf{d}}{\rho}\right)\sigma_n^2 \exp\left(-\frac{\sqrt{5}\mathbf{d}}{\rho}\right)\frac{\partial^2 \mathbf{d}}{\partial \mathbf{x}^2},\end{aligned} \tag{18}$$

where $\mathbf{d}$ is the vector of distances between $\mathcal{P}$ and $\mathbf{x}$. As for the RBFN, we have the first-order derivative as:

$$\frac{\partial \overline{g}(\mathbf{x})}{\partial x_i} = -\frac{1}{s^2}\sum_{j=1}^{k}(\mathbf{x}-\mathbf{c}^j)\overline{g}(\mathbf{x}), \tag{19}$$

and the second-order derivative as:

$$\frac{\partial^2 \overline{g}(\mathbf{x})}{\partial x_i \partial x_j} = \begin{cases} \frac{1}{s^4}\sum_{t=1}^{k}(x_i - c_j^t)(x_i - c_j^t)\overline{g}(\mathbf{x}), & \text{if } i \neq j \\ \frac{1}{s^4}\sum_{t=1}^{k}\left((x_i - c_i^d)^2 - s^2\right)\overline{g}(\mathbf{x}), & \text{if } i = j \end{cases}. \tag{20}$$

In summary, the working mechanism of this `manifold interpolation` step is given as follows.

Step 1: Initialize the candidate solution set $\mathcal{S} = \emptyset$.

Step 2: For $i = 1, \cdots, N$, do

- Step 2.1: Calculate the tangent vectors of $\mathbf{x}^i \in \mathcal{P}$ by solving the system of linear equations given in equation (13).
- Step 2.2: Use equation (10) to generate a set of $\overline{N} = \frac{\tilde{N}}{N}$ candidate solutions $\overline{\mathcal{S}} = \{\hat{\mathbf{x}}^k | \hat{\mathbf{x}}^k = \mathbf{x}_i + \sum_{j=1}^{m-1}\eta_j \mathbf{v}_j \text{ and } \eta_j \in (0,1]\}_{k=1}^{\overline{N}}$.
- Step 2.3: Remove invalid solutions in $\overline{\mathcal{S}}$ outside of $\Omega$.
- Step 2.4: $\mathcal{S} = \mathcal{S} \bigcup \overline{\mathcal{S}}$.

Step 3: Use the GPR model to predict the objective function values of solutions in $\mathcal{S}$.

Step 4: Output the non-dominated solutions in $\mathcal{S}$.

### 3.3 Batch Recommendation

This step is also known as the infill criterion in the surrogate-assisted EA literature. It aims to pick up $\xi \geq 1$ promising solutions from $\mathcal{C} = \mathcal{P} \bigcup \mathcal{S}$ and evaluate them by using the expensive objective functions. These newly evaluated solutions are then used to update the training dataset for the next iteration. Different from most, if not all, works using GPR as the surrogate model, our infill criterion does not rely on an uncertainty quantification measure, also known as acquisition function in the Bayesian optimization literature [58]. Furthermore, selecting a batch of samples to evaluate can significantly reduce the overhead for surrogate modeling. More specifically, we propose two alternative ways to implement this `batch recommendation` step.

- The first one is based on the individual Hypervolume contribution (IHV), independent of the underlying baseline algorithm. We calculate the IHV of each candidate solution $\mathbf{x} \in \mathcal{C}$ as:

$$\texttt{IHV}(\mathbf{x}) = \texttt{HV}(\mathcal{C}) - \texttt{HV}(\mathcal{C} \setminus \{\mathbf{x}\}), \tag{21}$$

  where $\texttt{HV}(\mathcal{C})$ evaluates the Hypervolume (HV) [59] of $\mathcal{C}$. Then, the top $\xi$ solutions in $\mathcal{C}$ with the largest IHV are picked up for the expensive evaluations.

- The other one is directly derived from the native environmental selection of the baseline EMO algorithm used in the `evolutionary search` step.

  – If the baseline algorithm is NSGA-II, we propose a four-step process for the batch recommendation.

  Step 1: Identify the non-dominated solutions in $\mathcal{C}$ and store them in $\overline{\mathcal{C}}$.

  Step 2: Use $N$ evenly distributed weight vectors to divide the objective space into $N$ subregions and associate each solution in $\overline{\mathcal{C}}$ to its closest weight vector with the smallest acute angle.

  Step 3: Pick up the best solution with the largest crowding distance for each subregion to constitute $\tilde{\mathcal{C}}$.

  Step 4: Pick up the top $\xi$ solutions from $\tilde{\mathcal{C}}$ with the largest crowding distances.

  **Remark 6.** *Note that the calculation of crowding distances is carried out separately with regard to each subregion. In other words, each solution only consider its closest neighbor lying in the same subregion.*

  – If the baseline algorithm is IBEA, we can directly use its fitness function to choose $\xi$ best solutions.

  – If the baseline algorithm is MOEA/D, we propose the following three-step process for the batch recommendation.

  Step 1: For each subproblem $g(\cdot|\mathbf{w}^i, \mathbf{z}^*)$ where $i \in \{1, \cdots, N\}$, identify the best solution $\mathbf{x}^{i*}$ in $\mathcal{C}$:

$$\mathbf{x}^{i*} = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{C}} g(\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*), \tag{22}$$

  Step 2: Calculate the fitness improvement on each subproblem with regard to the previous iteration.

$$\Delta_i = \frac{g(\hat{\mathbf{x}}^{i*}|\mathbf{w}^i, \mathbf{z}^*) - g(\mathbf{x}^{i*}|\mathbf{w}^i, \mathbf{z}^*)}{g(\hat{\mathbf{x}}^{i*}|\mathbf{w}^i, \mathbf{z}^*)}, \tag{23}$$

  where $i \in \{1, \cdots, N\}$ and $\hat{\mathbf{x}}^{i*}$ is the best solution of the $i$-th subproblem in the previous iteration.

  Step 3: Pick up the top $\xi$ solutions having the largest fitness improvements.

### 3.4 Algorithm Instances

As introduced in Section 3.1, any existing EMO algorithm can be used in the `evolutionary search` step. With regard to the two batch recommendation methods introduced in Section 3.3, we propose 6 algorithm instances for proof-of-concept purposes, dubbed as `DMI-x-IHV` by using the IHV for the batch recommendation or `DMI-x` by using the native environmental selection, respectively, where `x` is NSGA-II, IBEA or MOEA/D. Since we consider two different surrogate models, i.e., GPR and RBFN, in the `surrogate modeling` step, there are 12 algorithm instances in total.

### 3.5 Time Complexity Analysis

The `initialization` step has a linear complexity and is negligible. The complexity of the `surrogate model` step depends on the model, i.e., GPR is $\mathcal{O}(N^3)$ and RBFN is $\mathcal{O}(N^2)$. The `evolutionary search` step is governed by the corresponding baseline EMO algorithm, i.e., NSGA-II and IBEA are $\mathcal{O}(N^2)$ and MOEA/D is $\mathcal{O}(TN)$ where $T$ is the neighborhood size. In the `manifold interpolation` step, interpolating new solutions is linear while the complexity is governed by its last step that selects non-dominated solutions from $\mathcal{S}$, i.e., $\mathcal{O}(\tilde{N}^2)$. If we directly use the native environmental selection of the baseline EMO algorithm used in the `evolutionary search` step, the time complexity of the `batch recommendation` step is governed by the corresponding algorithm, i.e., NSGA-II and IBEA are $\mathcal{O}(\frac{\tilde{N}^2}{N})$ and MOEA/D is $\mathcal{O}(TN)$. Otherwise, it is $\mathcal{O}(|\mathcal{C}|^m)$ if we use IHV instead. All in all, the most time consuming part is the `batch recommendation` step. Therefore, the worst case complexity of our proposed framework is $\mathcal{O}(|\mathcal{C}|^m)$.

## 4 Experimental Setup

This section introduces the experimental settings for validating the effectiveness of our proposed batched data-driven EMO framework compared against seven state-of-the-art algorithms.

### 4.1 Benchmark Test Problems

In our empirical study, we consider benchmark test problems with both regular and irregular PF shapes to constitute our benchmark suite. More specifically, ZDT1, ZDT2, ZDT4, ZDT6, and DTLZ1 to DTLZ4 constitute the problems with regular PFs. As for those with irregular PFs, we have ZDT3 [60], DTLZ7 [22], minus DTLZ2 [61], mDTLZ2 [62], WFG2 [23], WFG41 to WFG48 [63]. Furthermore, based on ZDT3, DTLZ7 and WFG2, we develop a series of problems (dubbed ZDT3$\star$, DTLZ7$\star$ and WFG2$\star$) with a controllable number disconnected regions and imbalanced sizes. Their mathematical definitions and properties can be found in Appendix B. The number of objectives is set to $m = 2$ for the ZDT and $m = 3$ for the DTLZ problems. As for the WFG problems, we consider both 2- and 3-objective cases. The number of variables is set as $n \in \{5, 10, 20, 30\}$ respectively for each benchmark test problem. In total, there are 168 test problem instances considered in our experiments.

### 4.2 Peer Algorithms and Parameter Settings

To validate the competitiveness of our proposed algorithms, we compare their performance with seven state-of-the-art algorithms in the literature, including `ParEGO` [9], `MOEA/D-EGO` [10], `K-RVEA` [11], `EIM` [19], `TSMEA` [64], `HSMEA` [28] and `KTA2` [42]. We do not intend to delineate their working mechanisms here while interested readers are referred to their original papers for details.

The parameter settings are listed as follows.

- Number of function evaluations (FEs): The initial sampling size is set to $11 \times n - 1$ for all algorithms and the maximum number of FEs is set as 150 and 250 for $m = 2$ and 3, respectively.
- Reproduction operators: The parameters associated with the simulated binary crossover and polynomial mutation are set as $p_c = 1.0$, $\eta_c = 20$, $p_m = \frac{1}{n}$, $\eta_m = 20$. As for those use differential evolution for offspring reproduction, we set $CR = F = 0.5$.
- Kriging models: As for the algorithms that use Kriging for surrogate modeling, the corresponding hyperparameters of the MATLAB Toolbox DACE [65] is set to be within the range $[10^{-5}, 10^5]$.
- Batch size $\xi$: It is set as $\xi = 10$ for our proposed algorithms and $\xi = 5$ is set in `MOEA/D-EGO`, `K-RVEA` and `HSMEA`.
- Number of interpolated solutions $\tilde{N}$: This parameter controls the number of solutions tend to be generated by the `manifold interpolation` step and it is set as $\tilde{N} = 100$ in our experiments.

- Number of repeated runs: Each algorithm is independently run on each test problem for 31 times with different random seeds.

### 4.3 Performance Metrics and Statistical Tests

In our experiments, we use the HV and IGD+ [66] as the performance metrics to evaluate the performance of different peer algorithms. To have a statistical interpretation of the significance of comparison results, we use the following three statistical measures in our empirical study.

- Wilcoxon signed-rank test [67]: This is a non-parametric statistical test that makes no assumption about the underlying distribution of the data and has been recommended in many empirical studies in the EA community [68]. In particular, the significance level is set to $p = 0.05$ in our experiments.

- Scott-Knott test [69]: Instead of merely comparing the raw HV and IGD+ values, we apply the Scott-Knott test to rank the performance of different peer techniques over 31 runs on each test problem. In a nutshell, the Scott-Knott test uses a statistical test and effect size to divide the performance of peer algorithms into several clusters. In particular, the performance of peer algorithms within the same cluster are statistically equivalent. Note that the clustering process terminates until no split can be made. Finally, each cluster can be assigned a rank according to the mean HV and IGD+ values achieved by the peer algorithms within the cluster. In particular, the smaller the rank is, the better performance of the algorithm achieves.

- $A_{12}$ effect size [70]: To ensure the resulted differences are not generated from a trivial effect, we apply $A_{12}$ as the effect size measure to evaluate the probability that one algorithm is better than another. Specifically, given a pair of peer algorithms, $A_{12} = 0.5$ means they are *equivalent*. $A_{12} > 0.5$ denotes that one is better for more than 50% of the times. $0.56 \leq A_{12} < 0.64$ indicates a *small* effect size while $0.64 \leq A_{12} < 0.71$ and $A_{12} \geq 0.71$ mean a *medium* and a *large* effect size, respectively.

Note that both Wilcoxon signed-rank test and $A_{12}$ effect size are also used in the Scott-Knott test for generating clusters.

## 5 Empirical Studies

We seek to answer the following research questions (RQs) through our empirical study in the following paragraphs.

- *RQ1*: Given a surrogate model, how is the performance comparison among its six algorithm instances?

- *RQ2*: How is the performance comparison when using different surrogate model?

- *RQ3*: How is the performance of our proposed algorithm instances compared against state-of-the-art peer algorithms in the literature?

- *RQ4*: What are the benefits of manifold interpolation?

- *RQ5*: What are the benefits of batch recommendation?

- *RQ6*: What are the impacts of hyper-parameters?

- *RQ7*: How is performance of the proposed algorithm in real-world applications?

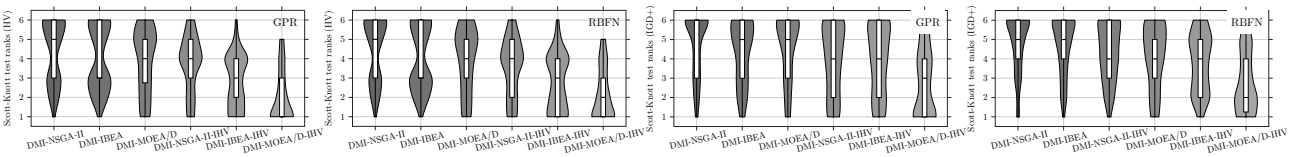



Figure 4: Violin plots of Scott-Knott test ranks of HV and IGD+ achieved by each of the six algorithm instances of our proposed framework by using GPR and RBFN in the `surrogate modeling` step, respectively (the smaller rank is, the better performance achieved).

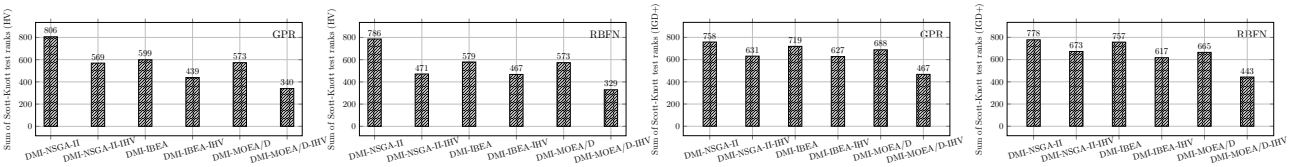



Figure 5: Bar charts of the total Scott-Knott test ranks of HV and IGD+ achieved by each of the six algorithm instances of our proposed framework by using GPR and RBFN in the `surrogate modeling` step, respectively (the smaller rank is, the better performance achieved).

### 5.1 Comparisons of six algorithm instances

Since we consider two different models (i.e., GPR and RBFN as introduced in Sections 2.2 and 2.3) in the `surrogate modeling` step, this subsection plans to study the performance comparison of six different algorithm instances under these two models separately. The statistical comparison results of both HV and IGD+ values, based on the Wilcoxon signed-rank test, among six algorithm instances introduced in Section 3.4 are given in Tables 1 to 8 of our supplementary materials[2]. From these results, we can see that the HV and IGD+ values obtained by different algorithms are close to each other; while the best algorithm alternates across different test problem instances.

To facilitate a better ranking among these algorithms, we apply the Scott-Knott test to classify them into different groups according to their performance on each test problem instance. Due to the large number of test problem instances used in our experiments, it will be messy if we list all ranking results ($168 \times 6 \times 2 = 2,016$ in total) obtained by the Scott-Knott test collectively. Instead, to have a better interpretation of the comparison among different algorithm instances, we pull all the Scott-Knott test results together (with regard to HV and IGD+ respectively) and show their distribution and variance as violin plots in Fig. 4. In addition, to facilitate an overall comparison, we further summarize the corresponding Scott-Knott test results obtained across all test problem instances for each algorithm instance and show them as the bar charts in Fig. 5. From these results, we can see that using the IHV in the batch recommendation has shown to be consistently better than the native environmental selection mechanism in NSGA-II, IBEA and MOEA/D. In particular, we clearly see that `DMI-MOEA/D-IHV` is the best algorithm instance of our proposed framework given that 1) its performance has been classified into the best group in most comparisons as the violin plots shown in Fig. 4; and 2) it obtains the smallest summation rank as shown in Fig. 5 (it is at least 30% better than the other five peer algorithms). `DMI-NSGA-II` is the worst algorithm instance, the inferior results obtained by which can be attributed to the use of the crowding distance. In particular, due to a large number of candidates solutions generated by the manifold interpolation, the overly crowded local niche makes the crowding distance less discriminative. As the example shown in Fig. 7, since the interpolated solutions are heavily crowded, the crowding distance always recommends the one lying in the boundary of the interpolated region whereas the internal solutions are ignored. In this case, it compromises the extra diversity provided by the `manifold recommendation` step. However, by using the IHV as an alternative of the crowding distance in the batch recommendation, the performance of `DMI-NSGA-II-IHV` is significantly promoted while it even obtains a better ranking than `DMI-MOEA/D` and `DMI-IBEA`.

At the end, we choose `DMI-MOEA/D-IHV` as a representative algorithm to compare the difference of

[2]The supplementary materials can be found in `https://tinyurl.com/3z4y57ec`.

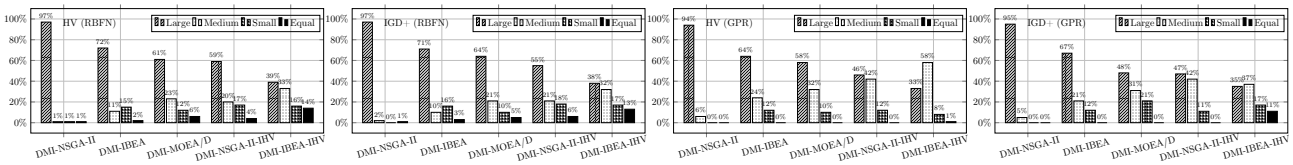



Figure 6: Percentage of the large, medium, small, and equal $A_{12}$ effect size of HV and IGD+ when comparing `DMI-MOEA/D-IHV` with other five peer algorithm instances by using GPR and RBNF, respectively.

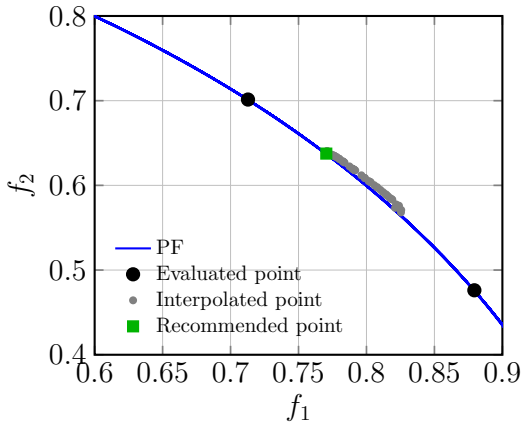



Figure 7: Illustrative example of the drawback of using the crowding distance in `DMI-NSGA-II`.

its performance with respect to the other five peer algorithms by using the $A_{12}$ effect size separately. Since the calculation of $A_{12}$ effect size is conducted in a pairwise manner, there are $168 \times 5 \times 2 = 1,680$ piecemeal $A_{12}$ comparison results. We again pull all results together and calculate the percentage of the equivalent, small, medium and large effect size, respectively, with respect each of the other five peer algorithms (note that there is barely equivalent case in these comparisons). From the statistical results shown in Fig. 6, it is interesting to note that `DMI-MOEA/D-IHV` has shown dominantly better results comparing to `DMI-NSGA-II` and `DMI-NSGA-II-IHV` where the large effect sizes are all over 90%. In contrast, the effect sizes with regard to `DMI-IBEA`, `DMI-IBEA-IHV` and `DMI-MOEA/D` are relatively comparable.

Answers to *RQ*1: We have the following takeaways from our experiments. 1) `DMI-MOEA/D-IHV` is the best algorithm instance of our proposed framework while `DMI-NSGA-II-IHV` and `DMI-NSGA-II` are the medium and worst ones respectively. 2) Owing to the unique characteristics of HV for measuring convergence and diversity simultaneously, the IHV has shown to be a better mechanism for guiding the batch recommendation. 3) In contrast, the crowding distance used in NSGA-II is too coarse-grained to pick up representative solutions from a large amount of candidates. 4) `MOEA/D` is the best baseline surrogate optimizer in the `evolutionary search` step while `NSGA-II` is the worst both for using the IHV and the native environmental selection in the batch recommendation. It is interesting to note that all these observations are consistent when using both GPR and RBFN in the `surrogate modeling` step.

### 5.2 Comparisons of using GPR and RBFN in the `surrogate modeling` step

Based on the results obtained in Section 5.1, we evaluate the $A_{12}$ effect size of HV and IGD+ between each of the best, medium and worst algorithm instances (i.e., `DMI-MOEA/D-IHV`, `DMI-NSGA-II-IHV` and `DMI-NSGA-II`) with regard to GPR and RBFN, respectively. As in Section 5.1, we calculate the percentage of different effect sizes obtained by a pair of dueling algorithms. From the results shown in Fig. 8, it is clear to see that using GPR as the surrogate model is consistently better than that of RBFN. In particular, more than 50% comparison results are classified as having a large effect size for `DMI-MOEA/D-IHV` by using GPR as the surrogate model have a large effect size against that of

using RBFN. It is interesting to note that the difference is narrowed on the worst algorithm instance `DMI-NSGA-II` where nearly 50% comparison results are classified as having a small effect size. All in all, these comparison results suggest that GPR is more capable to model the objective functions than RBFN. Nevertheless, the computational cost for building a GPR model is much higher than that of RBFN.

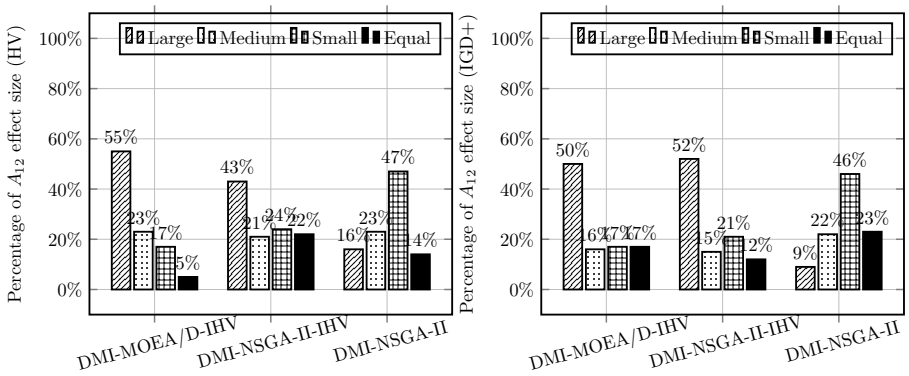



Figure 8: Percentage of the large, medium, small, and equal $A_{12}$ effect size of HV and IGD+, respectively, when comparing `DMI-MOEA/D-IHV`, `DMI-NSGA-II-IHV`, `DMI-NSGA-II` with GPR as the surrogate model against RBFN.

Answers to $RQ2$: Using GPR in the `surrogate modeling` step is more capable than RBFN in our proposed batched data-driven EMO framework. The only concern for GPR is its cubic worst case time complexity in model building.

### 5.3 Comparisons with other seven peer algorithms

Based on the results discussed in Section 5.1, we pick up the overall best, medium and worst algorithm instances of our proposed framework, i.e., `DMI-MOEA/D-IHV(GPR)`, `DMI-NSGA-II-IHV(GPR)` and `DMI-NSGA-II(RBFN)`, and compare their performance against the seven state-of-the-art peer algorithms introduced in Section 4.2. Similar to Section 5.1, we first pull all statistical comparison results of HV and IGD+ values, based on the Wilcoxon signed-rank test, among each of our 12 algorithm instances as introduced in Section 3.4 with regard to the other seven state-of-the-art peer algorithms as introduced in Section 4.2 in Tables 1 to 8 of our supplementary materials. From these results, we find that the HV and IGD+ values obtained by our algorithm instances are better than the other seven peer algorithms in most comparisons, even for `DMI-NSGA-II(RBFN)`, our least competitive algorithm instance. To have a better visual interpretation of the superiority achieved by our algorithm instances, let us look into the population distribution of the non-dominated solutions against the other seven peer algorithms. Due to the page limit, we only show a couple of examples in Figs. 9 and 10 while the complete results can be found in the supplementary materials. From these plots, it is clear to see that our proposed algorithms not only converge well to the PF, but are also resilient to the PF shapes. Especially for those with disconnected PF segments, our proposed algorithms can approximate all segments with a reasonable diversity. In contrast, the other peer algorithms are either struggling on converging to the PF or hardly approximate all disconnected PF segments. It is interesting to note that all algorithms have shown comparable results on WFG41 to WFG48 problems with two objectives. But the performance of the other seven peer algorithms degrade significantly on the three-objective cases. Another interesting observation is that the increase of the number of variables do not downgrade the performance of our proposed algorithm instances.

As in Section 5.1, we apply the Scott-Knott test to sort the performance of each algorithm instance against the other seven peer algorithms on all test problem instances. To facilitate a better interpretation of these massive comparison results, for each of our best, medium and worst algorithm instance, we pull $168 \times 8 \times 3 = 4,032$ comparison results collected from the Scott-Knott test together and show their distribution and variance as the violin plots in Fig. 11. From these results, we further confirm that our proposed algorithm instances are always better than the other peer algorithms in

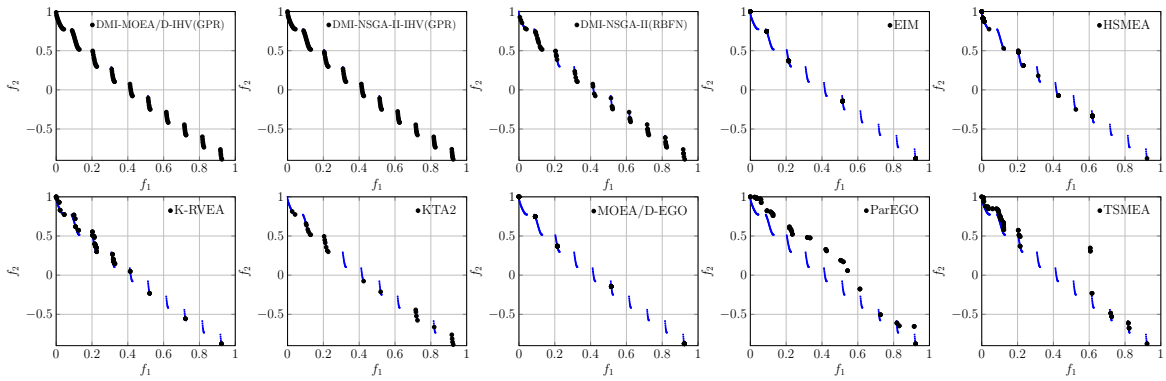



Figure 9: Non-dominated solutions found by different algorithms with the median HV values on ZDT31 ($n = 30$).

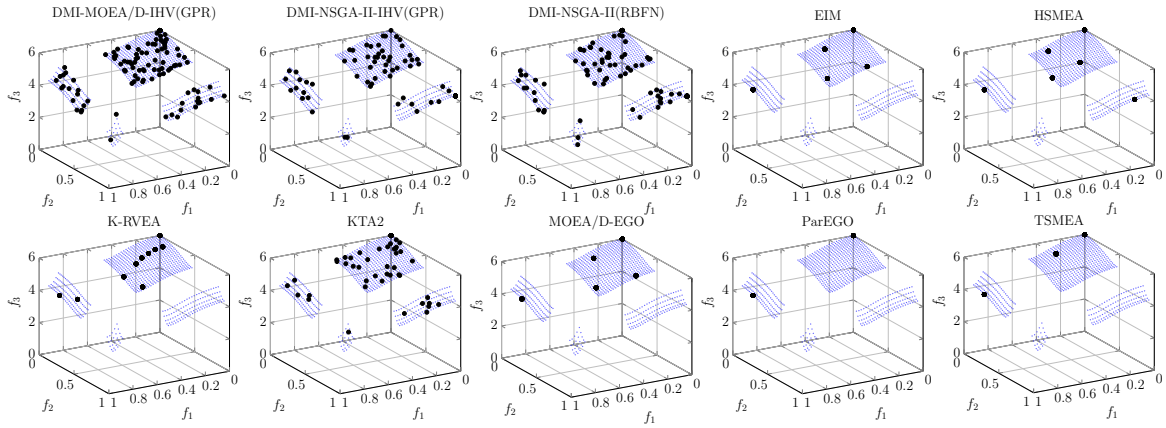



Figure 10: Non-dominated solutions found by different algorithms with the median HV values on DTLZ72 ($n = 30$).

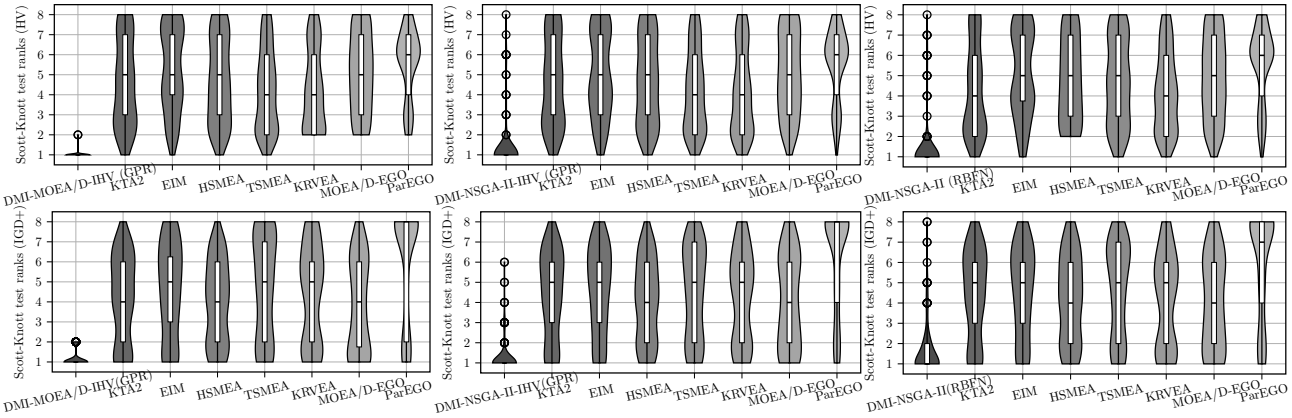



Figure 11: Violin plots of Scott-Knott test ranks achieved by our proposed algorithm instances compared against the other seven state-of-the-art peer algorithms (the smaller rank is, the better performance achieved).

the corresponding comparisons. Specifically, `DMI-MOEA/D-IHV(GPR)` is consistently ranked in the first place in all comparisons.

Again, we evaluate the $A_{12}$ effect size between each of `DMI-MOEA/D-IHV(GPR)`, `DMI-NSGA-II-IHV(GPR)` and `DMI-NSGA-II(RBFN)` with regard to the other seven state-of-the-art peer algorithms on each test problem instance. As in Section 5.1, we calculate the percentage of different effect sizes obtained by each algorithm instance against the other peer algorithms, respectively. As the bar charts shown in Fig. 12, we further confirm the overwhelming advantage observed from the Scott-Knott test where the percentage of the large effect size is always close to 100% in all comparisons ob-

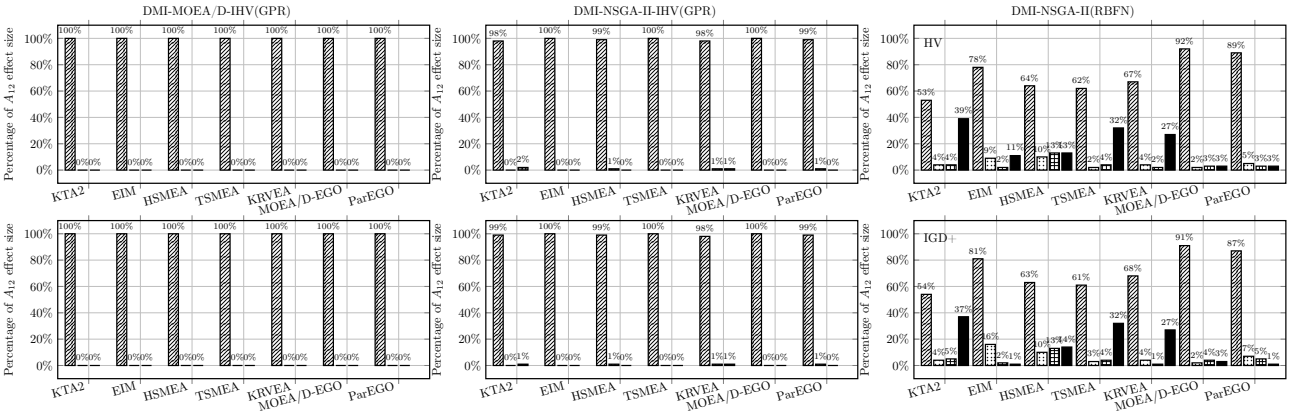



Figure 12: Percentage of the large, medium, and small $A_{12}$ effect size, respectively, when comparing each of our proposed algorithm instances compared against other seven state-of-the-art peer algorithms.

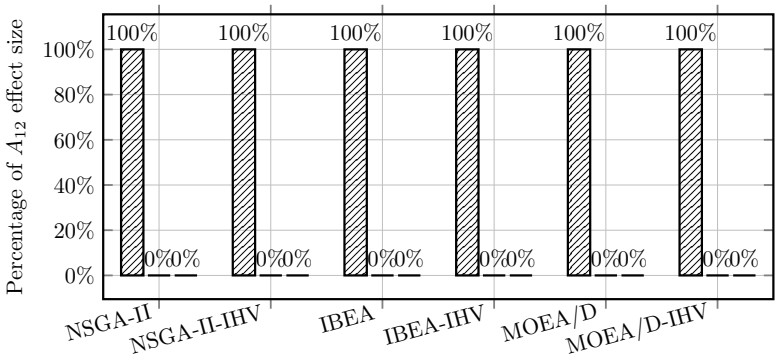



Figure 13: Percentage of the large, medium, and small $A_{12}$ effect size, respectively, when comparing each of our proposed six algorithm instances against its ablated variant without using the `manifold interpolation` step.

tained from `DMI-MOEA/D-IHV(GPR)` and `DMI-NSGA-II-IHV(GPR)`. In contrast, the superior performance `DMI-NSGA-II(RBFN)` is not as dominating as the other two, especially when comparing with the recently proposed `KTA2`.

Answers to *RQ*3: We have the following takeaways from this experiment. 1) All algorithm instances of our proposed framework have shown consistently better performance over the state-of-the-art surrogate-assisted EMO algorithms in the literature, even for `DMI-NSGA-II(RBFN)`, our least competitive algorithm instance. 2) The overwhelmingly better performance achieved by our proposed framework can be attributed to the `manifold interpolation` step that helps interpolate the approximated PS manifold thus significantly increases the population diversity for exploring disconnected regions.

### 5.4 Ablation study with regard to the manifold interpolation

The empirical study in Section 5.3 has shown overwhelmingly better performance of our proposed framework against the selected state-of-the-art surrogate-assisted EMO algorithms. Referring to Fig. 2, we can see the `manifold interpolation` step is the unique component of our proposed framework. To address RQ4, we plan to investigate the usefulness of this `manifold interpolation` step through an ablation study. To this end, we compare the performance between the algorithm instances under our proposed batched data-driven EMO framework against the corresponding ablated counterpart without using the `manifold interpolation` step. Accordingly, it is denoted as the one without the `DMI` prefix. Note that we only consider the best algorithm instance `DMI-MOEA/D-IHV(GPR)` in this study without loss of generality.

From the statistical comparison results of HV values, based on the Wilcoxon signed-rank test, shown in Tables 9 and 10 in the supplementary materials along with the $A_{12}$ effect size shown in Fig. 13, we have witnessed a clear performance degradation when ablating the `manifold interpolation` step without any exception. It is worth noting that their performance is worse than most of the selected state-of-the-art algorithms considered in Section 5.3 by referencing Tables 1 to 8 in the supplementary materials. As an example shown in Fig. 14, we can see that non-dominated solutions obtained by `MOEA/D-IHV` cannot fully approximate all disconnected PF segments. Without using the `manifold interpolation` step, `MOEA/D-IHV` is merely guided by the surrogate model which is highly likely to be guided to some local regions. This can be explained as the evolutionary population is far away from the PF at the early stage of the evolution. In contrast, the `manifold interpolation` step brings more candidates with additional diversity in the survival competition. Let us consider an illustrative example shown in Fig. 15. Without using the manifold interpolation, `MOEA/D-IHV` can only obtain the solution, denoted as the green square, lying the same PF segment of previously evaluated solutions. On the other hand, because of the interpolated solutions, `DMI-MOEA/D-IHV` is able to explore under discovered PF segment as spotted by the red square. Moreover, since the interpolated solutions are along the currently approximated PF rather than purely random solutions, they are prone to have a promising convergence property.

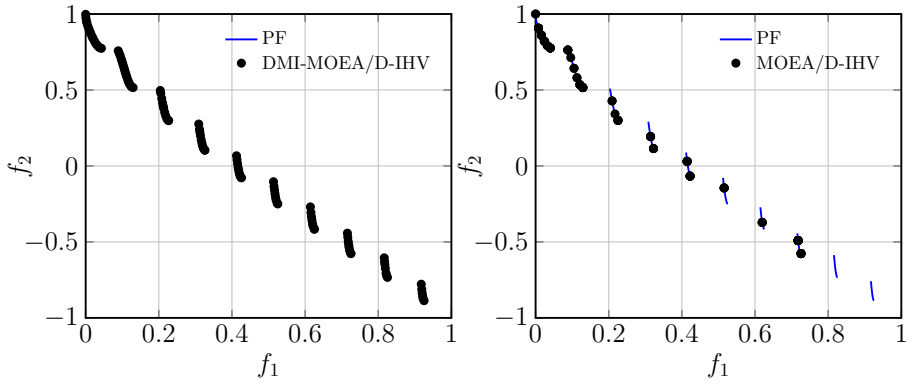



Figure 14: Comparative example of `DMI-MOEA/D-IHV` against its counterpart where the manifold interpolation is ablated on ZDT31 ($n = 30$).

Answers to *RQ*4: The `manifold interpolation` step is essential in our proposed framework. It not only brings sufficient diversity to expand the population, the interpolated solutions also have a promising convergence property given that they are interpolated along the approximated PS manifold. As a result, it enables our algorithms to have a faster convergence rate and a better ability to approximate different disconnected PF segments.

### 5.5 Parameter sensitivity study

In our proposed batched data-driven EMO framework, there are two hyper-parameters including the batch size $\xi$ in the `batch recommendation` step and the number of interpolated solutions $\tilde{N}$ in the `manifold interpretation` step. To address RQ5 and RQ6, we choose `DMI-MOEA/D-IHV(GPR)` as the baseline and empirically investigate its performance under different $\xi = \{1, 5, 10, 20\}$ and $\tilde{N} = \{50, 100, 200\}$ settings. In particular, $\xi = 1$ represents the sequential recommendation counterpart.

From the statistical comparison results of HV values, based on the Wilcoxon signed-rank test, shown in Tables 11 and 12 in the supplementary materials along with the $A_{12}$ effect size shown in Fig. 16, we can see that the performance of `DMI-MOEA/D-IHV(GPR)` with $\xi = 10$ is comparable with that of $\xi = 1$ and $\xi = 5$ where most of the differences are classified as statistically equivalent. Given a limited amount of FEs, a smaller $\xi$ leads to more iterations as in our proposed framework thus it is more time consuming. Fig. 17 gives the comparison of CPU wall clock time among different $\xi$ settings. From this figure, we can see that `DMI-MOEA/D-IHV(GPR)` is around $10\times$ slower when $\xi = 1$ than those of $\xi = 5, 10, 20$. In addition, as an example shown in Fig. 18, using a too small $\xi$ may compromise the chance for exploring under discovered PF segment(s) as the solution #7 (denoted as $\times$) lying in

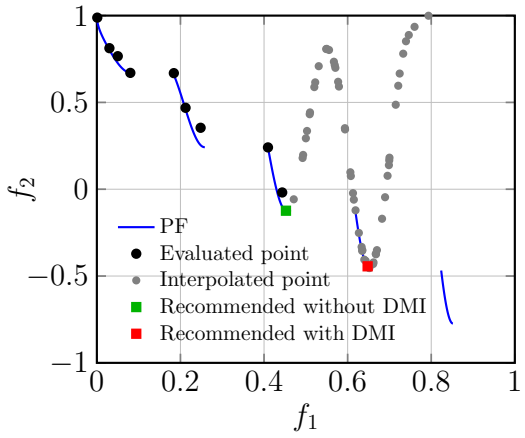



Figure 15: Illustrative example of the effectiveness of having `manifold interpolation` step.

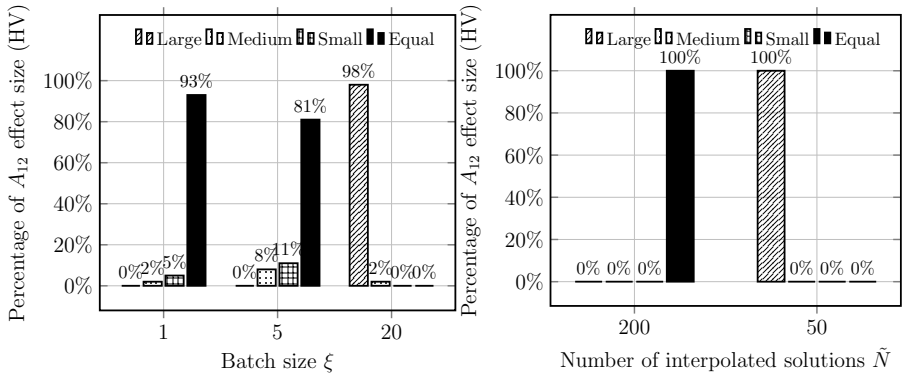



Figure 16: Percentage of the large, medium, small, and equal $A_{12}$ effect size, respectively, when comparing `DMI-MOEA/D-IHV` with our recommended settings (i.e., $\xi = 10$ and $\tilde{N} = 100$) against others (i.e., $\xi = 1, 5, 20$ and $\tilde{N} = 50, 200$ respectively).

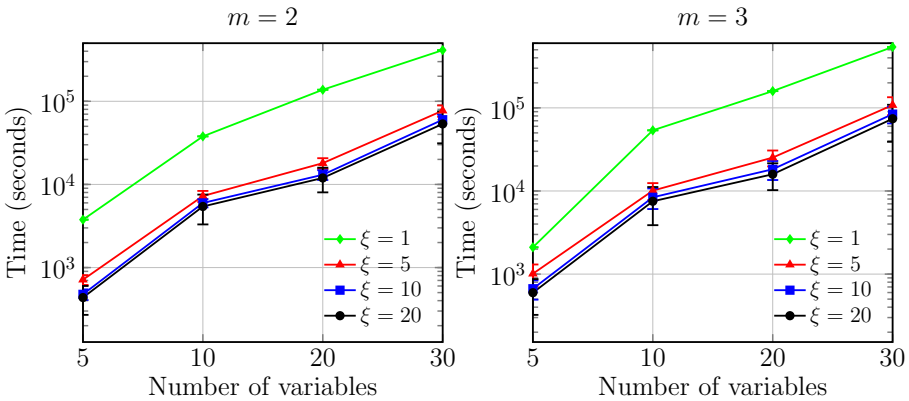



Figure 17: Collected comparisons of CPU wall clock time when using different $\xi$ settings.

a new segment. On the other hand, although it is faster when picking up more solutions in the `bath recommendation` step by setting a larger $\xi$, the surrogate model becomes less resilient to local optima due to the reduced iterations for updating the surrogate model. As the comparison results of $A_{12}$ effect size shown in Fig. 16, it is clear to see a large performance degradation when increasing $\xi$ to 20.

As the comparison results shown in Tables 13 and 14 in the supplementary materials along with the $A_{12}$ effect size shown in Fig. 16, it is interesting to note that the performance of `DMI-MOEA/D-IHV(GPR)` is significantly degraded when having too small interpolated solutions (i.e., $\tilde{N} = 50$) whereas it does not make statistically meaningful difference when we further increase $\tilde{N}$. However, the computational time is significantly increased in the `batch recommendation` step when having a large amount of interpolated solutions.

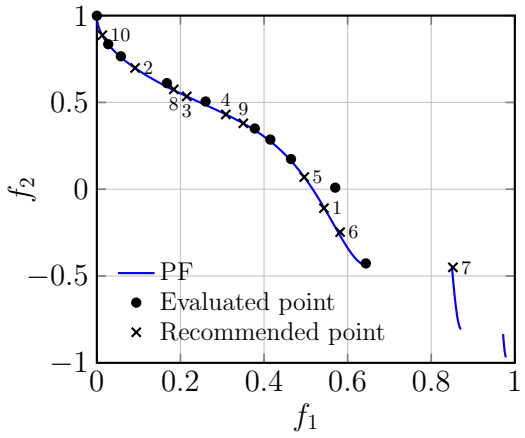



Figure 18: Illustrative example of batch recommendation results between $\xi = 5$ and $\xi = 10$. Specifically, Solutions #1 to #5 (denoted as $\times$) are recommended when $\xi = 5$ while solutions #1 to #10 are recommended by setting $\xi = 10$.

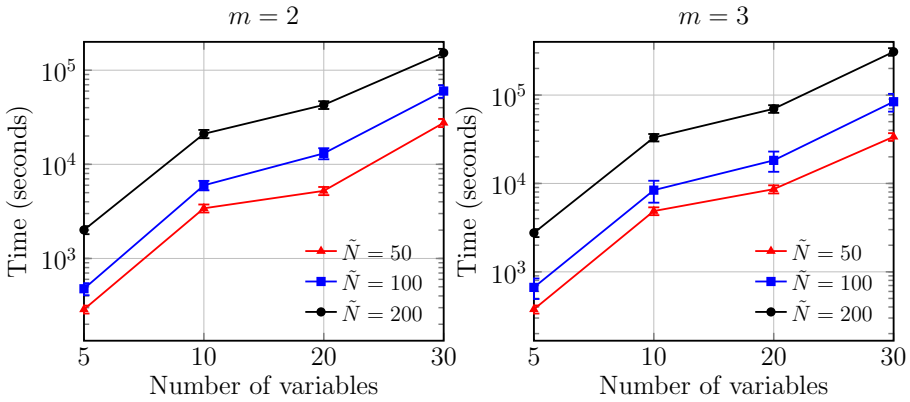



Figure 19: Collected comparisons of CPU wall clock time when using different $\tilde{N}$ settings.

Answers to $RQ5$: We have the following takeaways from our experiments. 1) Comparing against the batch recommendation, the sequential recommendation (i.e., $\xi = 1$) makes the chosen solution(s) to be less representative with regard to the PF. This applies to a smaller $\xi$ setting (e.g., $\xi = 5$). 2) In addition, it becomes much more time consuming when using a small $\xi$. 3) On the other hand, a too large $\xi$ renders the surrogate model less resilient to local optima.

Answers to $RQ6$: `DMI-MOEA/D-IHV(GPR)` is not very sensitive to the number of interpolated solutions $\tilde{N}$ generated in the `manifold interpolation` step. However, the computational time grow significantly with the increase of $\tilde{N}$.

### 5.6 Real-World Application in Hyper-parameter Optimization

Due to the page limit, the detailed discussion of this experiment can be found in Appendix C. Here we only come up with the conclusion.

Answers to $RQ7$: `DMI-MOEA/D-IHV(GPR)` has shown overwhelmingly better performance compared against the other seven state-of-the-art peer algorithms. In particular, we can appreciate that real-world problems are hardly with regular PFs as discussed in [21].

## 6 Conclusions and Future Directions

This paper proposed a batched data-driven EMO framework for solving computationally expensive MOPs. It has three distinctive features. First, this framework is so general that any existing EMO algorithm can be applied in a plug-in manner as the surrogate optimizer in the `evolutionary search`

step. Second, based on the KKT conditions, its `manifold interpolation` step interpolates along the approximated PS manifold to generate more diversified candidate solutions with a convergence guarantee. Last but not the least, it provides two types of approach in the `batch recommendation` step to evaluate multiple promising solutions for expensive FEs in parallel. Extensive experiments on 168 benchmark test problem instances with various irregular PFs fully demonstrate the effectiveness and overwhelming superiority against six state-of-the-art EMO algorithms. Note that our ablation study validates that the `manifold interpolation` step is essential within our proposed framework.

Data-driven evolutionary optimization has been an emerging area given the pressing requirements of sample-efficient real-world applications in various disciplines. In view of the strong performance and simple architecture of our proposed batched data-driven EMO framework, we envisage several aspects for future endeavors as follows.

- This paper only considers problems with two- and three- objectives given the already overwhelming superiority against the state-of-the-art. One of the future directions is to extend it for many-objective optimization problems. A typical challenge is the ineffectiveness of the sampling strategy suggested in equation (10) for high-dimensional problems. On the other hand, sampling too many candidate solutions during the `manifold interpolation` step incurs significantly mounting complexity in the `batch recommendation` step.

- In addition to the scalability in the objective space, the increase of the number of variables, as known as large-scale multi-objective optimization, also brings in significant challenges in both surrogate modeling and evolutionary optimization. One tentative way to combat the curse-of-dimensionality is divide-and-conquer that decomposes the original large-scale problem into smaller ones.

- Real-world problems are usually accompanied with various constraints, the existing of which render the search space to be teared up into fragments. These lead to challenges in sampling and surrogate modeling since infeasible solutions tend to be useless in model building.

- Last but not the least, many exciting real-world applications are featured with multiple conflicting objectives and computationally expensive FEs. It is impactful to apply data-driven evolutionary optimization for those complex black-box problems.

## Acknowledgment

This work was supported by UKRI Future Leaders Fellowship (MR/S017062/1), EPSRC (2404317), NSFC (62076056), Royal Society (IES/R2/212077) and Amazon Research Award.

## References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* New York, NY, USA: John Wiley & Sons, Inc., 2001.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[3] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *PPSN'VIII: Proc. of the 8th International Conference on Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 3242. Springer, 2004, pp. 832–842.

[4] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[5] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Comp. Int. Mag.*, vol. 4, no. 3, pp. 62–76, 2009.

[6] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.

[7] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, 2019.

[8] I. Loshchilov, M. Schoenauer, and M. Sebag, "Dominance-based Pareto-surrogate for multi-objective optimization," in *SEAL'10: Proc. of the 8th International Conference Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, vol. 6457. Springer, 2010, pp. 230–239.

[9] J. D. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, 2006.

[10] Q. Zhang, W. Liu, E. P. K. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.

[11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, 2018.

[12] G. Sun, G. Li, Z. Gong, G. He, and Q. Li, "Radial basis functional model for multi-objective sheet metal forming optimization," *Eng. Optim.*, vol. 43, no. 12, pp. 1351–1366, 2011.

[13] S. Z. Martínez and C. A. C. Coello, "MOEA/D assisted by RBF networks for expensive multi-objective optimization problems," in *GECCO'13: Proc. of the 2013 Genetic and Evolutionary Computation Conference*. ACM, 2013, pp. 1405–1412.

[14] T. Akhtar and C. A. Shoemaker, "Multi-objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection," *J. Glob. Optim.*, vol. 64, no. 1, pp. 17–32, 2016.

[15] J. Mockus, "Application of Bayesian approach to numerical methods of global and stochastic optimization," *J. Glob. Optim.*, vol. 4, no. 4, pp. 347–365, 1994.

[16] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *ICML'10: Proc. of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 1015–1022.

[17] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," *J. Basic Eng.*, vol. 86, no. 1, pp. 97–106, 1964.

[18] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[19] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 956–975, 2017.

[20] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, 2019.

[21] H. Ishibuchi, L. He, and K. Shang, "Regular Pareto front shape is not realistic," in *CEC'19: Proc. of the 2019 IEEE Congress on Evolutionary Computation*. IEEE, 2019, pp. 2034–2041.

[22] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced Information and Knowledge Processing. Springer, 2005, pp. 105–145.

[23] S. Huband, P. Hingston, L. Barone, and R. L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[24] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4108–4121, 2017.

[25] F. Gu and Y. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211–225, 2018.

[26] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376–390, 2019.

[27] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, "Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 439–453, 2020.

[28] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, 2019.

[29] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, 2008.

[30] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proc. of the 2nd Berkeley Symposium on Mathematical Statistics and Probability.* Berkeley, CA, USA: University of California Press, 1951, pp. 481–492.

[31] K. Miettinen, *Nonlinear Multiobjective Optimization.* Boston, USA: Kluwer Academic Publishers, 1999.

[32] C. Hillermeier, "Generalized homotopy approach to multiobjective optimization," *J. Optim. Theory Appl.*, vol. 110, no. 3, pp. 557–583, 2001.

[33] O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich, "Covering Pareto sets by multilevel evolutionary subdivision techniques," in *EMO'03: Proc. of the Second International Conference Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 2632. Springer, 2003, pp. 118–132.

[34] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[35] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.

[36] D. Serre, *Matrices: Theory and Applications.* New York, NY, USA: Springer, 2002.

[37] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, 2011.

[38] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, 2006.

[39] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted -metric selection," in *PPSN X: Proc. of the 10th International Conference on Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 5199. Springer, 2008, pp. 784–794.

[40] T. Wagner, M. Emmerich, A. H. Deutz, and W. Ponweiser, "On expected-improvement criteria for model-based multi-objective optimization," in *PPSN XI: Proc. of the 11th International Conference on Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 6238. Springer, 2010, pp. 718–727.

[41] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Inf. Sci.*, vol. 519, pp. 317–331, 2020.

[42] Z. Song, H. Wang, C. He, and Y. Jin, "A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, 2021.

[43] I. Voutchkov and A. Keane, *Multi-Objective Optimization Using Surrogates.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 155–175.

[44] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, 2019.

[45] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *CEC'15: Proc. of the 2015 IEEE Congress on Evolutionary Computation.* IEEE, 2015, pp. 2883–2890.

[46] C. Seah, Y. Ong, I. W. Tsang, and S. Jiang, "Pareto rank learning in multi-objective evolutionary algorithms," in *CEC'12: Proc. of the 2012 IEEE Congress on Evolutionary Computation.* IEEE, 2012, pp. 1–8.

[47] J. Luo, A. Gupta, Y. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front gaussian process surrogates," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1708–1721, 2019.

[48] A. T. W. Min, Y. Ong, A. Gupta, and C. K. Goh, "Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 15–28, 2019.

[49] C. Yang, J. Ding, Y. Jin, and T. Chai, "Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 409–423, 2020.

[50] J. Rakowska, R. T. Haftka, and L. T. Watson, "Tracing the efficient curve for multi-objective control-structure optimization," *Computing Systems in Engineering*, vol. 2, no. 5–6, pp. 461–471, 1991.

[51] A. Schulz, H. Wang, E. Grinspun, J. Solomon, and W. Matusik, "Interactive exploration of design trade-offs," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 131:1–131:14, 2018.

[52] K. Deb and M. Abouhawwash, "An optimality theory-based proximity measure for set-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 515–528, 2016.

[53] K. Deb, M. Abouhawwash, and H. Seada, "A computationally fast convergence measure and implementation for single-, multiple-, and many-objective optimization," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 4, pp. 280–293, 2017.

[54] M. Abouhawwash, H. Seada, and K. Deb, "Towards faster convergence of evolutionary multi-criterion optimization algorithms using karush kuhn tucker optimality based local search," *Comput. Oper. Res.*, vol. 79, pp. 331–346, 2017.

[55] H. Seada, M. Abouhawwash, and K. Deb, "Multiphase balance of diversity and convergence in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 503–513, 2019.

[56] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1167–1189, 2009.

[57] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments, Second Edition.* Springer-Verlag, 2018.

[58] P. I. Frazier, "A tutorial on Bayesian optimization," *CoRR*, vol. abs/1807.02811, 2018. [Online]. Available: http://arxiv.org/abs/1807.02811

[59] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[60] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[61] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, 2017.

[62] Z. Wang, Y. Ong, and H. Ishibuchi, "On scalable multiobjective test problems with hardly dominated boundaries," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 217–231, 2019.

[63] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 423–441, 2015.

[64] E. Bradford, A. M. Schweidtmann, and A. A. Lapkin, "Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm," *J. Glob. Optim.*, vol. 71, no. 2, pp. 407–438, 2018.

[65] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard, "DACE — A MATLAB Kriging toolbox," Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002.

[66] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *EMO'15: Proc. of the 8th International Conference on Evolutionary Multi-Criterion Optimization*, vol. 9019. Springer, 2015, pp. 110–125.

[67] F. Wilcoxon, "Individual comparisons by ranking methods," 1945.

[68] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[69] N. Mittas and L. Angelis, "Ranking and clustering software cost estimation models through a multiple comparisons algorithm," *IEEE Trans. Software Eng.*, vol. 39, no. 4, pp. 537–551, 2013.

[70] A. Vargha and H. D. Delaney, "A critique and improvement of the cl common language effect size statistics of mcgraw and wong," *J. Educ. Behav. Stat.*, vol. 25, no. 2, pp. 101–132, 2000.