Are You Concerned about Limited Function Evaluations: Data-Augmented Pareto Set Learning for Expensive Multi-Objective Optimization

Yongfan Lu, Bingdong Li*, Aimin Zhou

Shanghai Institute of AI for Education, School of Computer Science and Technology,
and Key Laboratory of MEA (Ministry of Education), East China Normal University, Shanghai 200062, China
51255901053@stu.ecnu.edu.cn, {bdli, amzhou}@cs.ecnu.edu.cn

Abstract

Optimizing multiple conflicting black-box objectives simultaneously is a prevalent occurrence in many real-world applications, such as neural architecture search, and machine learning. These problems are known as expensive multi-objective optimization problems (EMOPs) when the function evaluations are computationally or financially costly. Multi-objective Bayesian optimization (MOBO) offers an efficient approach to discovering a set of Pareto optimal solutions. However, the data deficiency issue caused by limited function evaluations has posed a great challenge to current optimization methods. Moreover, most current methods tend to prioritize the quality of candidate solutions, while ignoring the quantity of promising samples. In order to tackle these issues, our paper proposes a novel multi-objective Bayesian optimization algorithm with a data augmentation strategy that provides ample high-quality samples for Pareto set learning (PSL). Specifically, it utilizes Generative Adversarial Networks (GANs) to enrich data and a dominance prediction model to screen out high-quality samples, mitigating the predicament of limited function evaluations in EMOPs. Additionally, we adopt the regularity model to expensive multi-objective Bayesian optimization for PSL. Experimental results on both synthetic benchmarks and real-world applications demonstrate that our algorithm outperforms several state-of-the-art and classical algorithms.

Introduction

Expensive multi-objective optimization problems (EMOPs) are commonly seen in various real-world applications (Jablonka et al. 2021; Baia et al. 2022; Xie et al. 2021; Yang et al. 2023). These problems typically entail conflicting objectives and costly evaluations, such as antenna structure design (Ding et al. 2019), clinical drug trials (Yu, Ramakrishnan, and Meinzer 2019), and neural network structure search (Lu et al. 2019), etc. For example, sampling all designs of 11 configuration options for x264 in software configuration tuning can cost up to 1,536 hours (Chen and Li 2021). Consequently, the number of function evaluations is severely limited due to time/financial budget. Furthermore, it is desirable to discover a set of optimal solutions (Pareto optimal set) rather than a single one, which imposes further difficulty.

*Corresponding Author: bdli@cs.ecnu.edu.cn.

To address these challenges, multi-objective Bayesian optimization (MOBO) (Laumanns and Ocenasek 2002), a generalization of single-objective BO (Močkus 1975a) to EMOPs, has shown to be a promising approach. Bayesian Optimization (BO) is a highly effective global optimization strategy that has demonstrated tremendous success in handling black-box optimization problems (Jones, Schonlau, and Welch 1998; Snoek, Larochelle, and Adams 2012). The primary concept behind BO is to construct surrogate models based on probability, which approximate the black-box functions and employ acquisition functions to explore globally optimal solutions. Multi-objective Bayesian Optimization (MOBO) combines the principles of Bayesian optimization and multi-objective optimization. One popular MOBO strategy is random scalarization, which maps the multi-objective problem into a group of single-objective problems. Another effective strategy is based on efficient acquisition functions, such as the expected hypervolume improvement (EHVI) (Couckuyt, Deschrijver, and Dhaene 2014) and predictive entropy search (PES) (Hoffman and Ghahramani 2015).

However, the data deficiency issue caused by limited function evaluations still poses a great challenge to the current MOBO methods. Most current methods tend to prioritize the quality of candidate solutions while ignoring the quantity of promising samples. Moreover, it is often the case that the search space expands exponentially as the dimensionality of decision variables increases (Tian et al. 2021), which further worsens the data deficiency issue. As a result, most MOBO algorithms for EMOPs are confined to small-scale variables, typically no more than 10 dimensions, due to the limitations imposed by function evaluations. For example, in the DGEMO experiment (Konakovic Lukovic, Tian, and Matusik 2020), the number of decision variables varies between 2 and 7, while in both the PSL-MOBO (Lin et al. 2022) and qNEHVI (Daulton, Balandat, and Bakshy 2021) experiments, the number of decision variables does not exceed 6. Concerning PSL-MOBO, a recently proposed Pareto set learning based MOBO, the preference learning and surrogate model can be quite unstable in expensive scenarios. Such instability leads to significant deviations between the approximate Pareto set and the true Pareto set. In other words, the quality of the approximate Pareto set is highly dependent on the accuracy of the surrogate model prediction

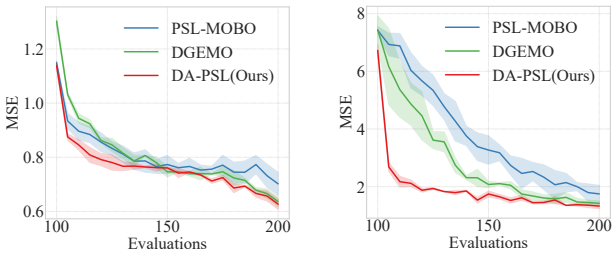

Figure 1: The average prediction MSE profiles for the surrogate models in PSL-MOBO, DGEMO, and DA-PSL (Ours) on DTLZ2 (left: $M = 3, d = 6$, right : $M = 3, d = 20$).

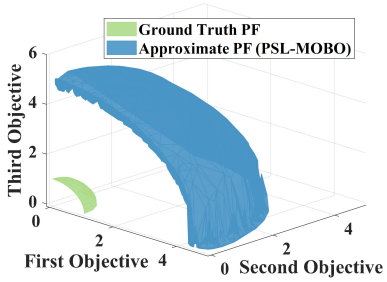

Figure 2: Approximate Pareto fronts got by PSL-MOBO and ground truth PF of DTLZ2 ($M = 3, d = 20$).

and the performance of the Pareto set learning model. As shown in Figure 1 (a), the mean-square error (MSE) for the surrogate models of PSL-MOBO and DGEMO is quite high in the early stage on 6-dimensional DTLZ2 problem. The situation becomes even worse when the number of decision variables is 20 (Figure 1 (b)). This further leads to a substantial deviation between the learned Pareto front approximation of PSL-MOBO and the ground truth Pareto front (Figure 2).

On the above-mentioned basis, we propose a novel multi-objective Bayesian optimization (MOBO) approach that utilizes data-augmented Pareto set learning to address the challenges posed by limited function evaluations in expensive multi-objective optimization. The main contributions are summarized as follows:

- We propose a novel data augmentation method that utilizes Generative Adversarial Networks (GANs) to enrich data for Pareto set learning in EMOPs. Specifically, already-evaluated solutions are categorized as $fake$ and $real$ optima based on a comparison strategy and utilized as a training set.
- We design a lightweight dominance prediction model for estimating the quality of solutions. High-quality solutions screened out by the dominance prediction model are utilized for better Pareto set learning (PSL) with the regularity model.
- We have conducted comprehensive experiments on both synthetic benchmarks and real-world applications, which demonstrate that our proposed algorithm can achieve competitive performance compared to several state-of-the-art MOBO algorithms in EMOPs with a restricted evaluation budget. Furthermore, we validated the effectiveness and efficiency of the proposed data augmentation strategy.

Preliminaries

Expensive Multi-objective Optimization

Without loss of generality, a multi-objective optimization problem (MOP) can be mathematically stated as follows :

$$\begin{aligned} &minimize\ \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))^T \\ &subject\ to\ \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ is the decision vector with d variables, $\boldsymbol{f}(\cdot)$: $\Omega \to \Lambda$ is the black-box objective function including M ($M \geq 2$) objectives, Ω is the (nonempty) *decision space* and Λ is the *objective space*. It is called an expensive MOP when evaluating $\boldsymbol{f}(\boldsymbol{x})$ is expensive and time-consuming computation or high-cost trials are involved. In such cases, the goal of optimizing an MOP is to approximate the PF within a limited evaluation budget.

Definition 1 (Pareto dominance) *Given two solutions $\boldsymbol{x}$, $\boldsymbol{y}$ in the feasible region Ω, $\boldsymbol{x}$ is said to dominate $\boldsymbol{y}$ (denoted as $\boldsymbol{x} \prec \boldsymbol{y}$) if and only if $\forall i \in \{1, 2, ..., M\}, f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ and $\exists j \in \{1, 2, ..., M\}, f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$. (Yu 1974)*

Definition 2 (Pareto optimal) *A solution $\boldsymbol{x}^* \in \Omega$ is Pareto optimal if no other solution $\boldsymbol{x} \in \Omega$ can dominate it.*

Definition 3 (Pareto Set and Pareto Front) *The solution set consisting of all the Pareto optimal solutions is called the Pareto set (PS): PS=$\{\boldsymbol{x} \in \Omega | \forall \boldsymbol{y} \in \Omega, \boldsymbol{y} \nprec \boldsymbol{x}\}$ and the corresponding objective vector set of the PS is the Pareto front (PF).*

The goal of optimizing an MOP is to approximate the PF considering the following two sub-goals (Li, Yang, and Liu 2014): 1) proximity: the approximation set is as close as possible to the Pareto front. 2) distribution: the approximation set spreads as diversely as possible.

Bayesian Optimization (BO)

For an unknown black-box function $\boldsymbol{f}$, neither its analytical form nor its derivatives are available. The central goal of Bayesian optimization is to design an effective strategy for selecting promising solutions, which balances exploration and exploitation. In each iteration of BO, surrogate models are constructed based on previously evaluated solutions for each objective independently. Typically, Gaussian process models are adopted to approximate $\boldsymbol{f}$. Subsequently, an acquisition function is employed to navigate the search direction. The posterior distribution of the surrogate model is updated based on the latest evaluated solutions, which are more promising under the specific acquisition function.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have enjoyed remarkable success as a generative model capable of learning complex data characteristics (Goodfellow et al. 2020). In general, a pair of GANs consists of two neural networks:

a $generator$ G and a $discriminator$ D. The $generator$ maps Gaussian noise $\boldsymbol{z} \in P_{\boldsymbol{z}}$ (where $P_{\boldsymbol{z}}$ represents the distribution of Gaussian noise $\boldsymbol{z}$) to a distribution $G(\boldsymbol{z})$, thereby synthesizing meaningful instances from a prior distribution. The $discriminator$ is trained to distinguish between $fake$ samples $\hat{\boldsymbol{x}} \in P_{\hat{\boldsymbol{x}}}$ (where $P_{\hat{\boldsymbol{x}}}$ represents the distribution of generated data) synthesized by the $generator$ and $real$ samples $\boldsymbol{x}^{real}$ provided by the training set. Generally speaking, the $generator$ aims to learn the distribution of training instances and generate $fake$ instances to deceive the $discriminator$, while the $discriminator$ is intended to distinguish between generative and $real$ samples. Ideally, the training procedure ends with the $discriminator$ being unable to distinguish the authenticity of a particular sample. Specifically, the training process is akin to a game between the $generator$ and the $discriminator$, and the loss function V is defined as follows:

$$\min_G \max_D V(G,D) = E_{\boldsymbol{x}\in P_{\boldsymbol{x}}}[\log D(\boldsymbol{x})] + E_{\boldsymbol{z}\in P_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z})))] \quad (2)$$

where $G(\boldsymbol{z})$ denotes the data generated from the input noise variable $\boldsymbol{z}$, while $D(\cdot)$ represents the predicted probability by the $discriminator$. The predicted probability tends towards 0 for fake instances and 1 for genuine instances, where $\boldsymbol{x}$ represents the data from true instances.

Our Method

Overview

We present a data-augmented Pareto set learning method for MOBO, denoted as DA-PSL (Algorithm 1 and Figure 3). DA-PSL includes three core components: data augmentation with two neural networks, Pareto set learning with regularity model, and batch selection strategy based on Gaussian process models. Firstly, it initializes a set of samples $\boldsymbol{X}_0 \subset \mathcal{X}$, where $\mathcal{X} \subset \mathcal{R}^d$, drawn by employing *Latin Hypercube Sampling* (LHS) (McKay, Beckman, and Conover 2000). Then, the three components are invoked iteratively.

Data Augmentation

In order to better approximate the true PS, we propose a data augmentation strategy that is summarized in Algorithm 2. Specifically, it utilizes Generative Adversarial Networks (GANs) to enrich data and a dominance prediction model to screen out high-quality samples. Initially, the shift based density estimation (SDE) (Li, Yang, and Liu 2013) is adopted to calculate the fitness value:

$$Fit(\boldsymbol{p}) = \min_{\boldsymbol{q}\in\boldsymbol{Y}_k\setminus\boldsymbol{p}} \sqrt{\sum_{i=1}^{M}(\max\{0, f_i(\boldsymbol{q}) - f_i(\boldsymbol{p})\})^2} \quad (3)$$

Here, $\boldsymbol{p}$ and $\boldsymbol{q}$ denote the solutions in $\boldsymbol{Y}_k$, and $f_i(\boldsymbol{p})$ denotes the i-th objective value of $\boldsymbol{p}$. The SDE considers the quality of samples in terms of convergence and diversity. A set of T (e.g., $\frac{|\boldsymbol{X}_k|}{3}$) candidate solutions in $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ with better SDE values, are treated as $real$ Pareto optimal samples, while the remaining are considered as $fake$ ones.

Algorithm 1: Data-Augmented Pareto Set Learning for Multi-Objective Bayesian Optimization

input : black-box function $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_M(\boldsymbol{x}))$, number of iterations K, batch size B, number of initial solutions N
output: final solutions $\{\boldsymbol{X}_K, \boldsymbol{Y}_K\}$, Pareto front $\mathcal{P}_f$

Initial N solutions $\{\boldsymbol{X}_0, \boldsymbol{Y}_0\}$ by LHS
for $k \leftarrow 0$ *to* $K-1$ **do**
  Train surrogate model GP_i^k based on $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ for each objective $f_i, i = 1, ..., M$
  $\boldsymbol{X}_k^{DP}, \boldsymbol{X}_k^{real} \leftarrow$ Data augmentation based on $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ **(Algorithm 2)**
  $\boldsymbol{X}_k^{PSL} \leftarrow$ Pareto set learning based on $\boldsymbol{X}_k^{DP}$ and $\boldsymbol{X}_k^{real}$ **(Algorithm 3)**
  $\boldsymbol{X}_k^{B} \leftarrow$ Batch selection based on GP^k, $\boldsymbol{X}_k^{PSL}$ and $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$
  Evaluate and update $\boldsymbol{X}_{k+1} \leftarrow \boldsymbol{X}_k \cup \boldsymbol{X}_k^B$, $\boldsymbol{Y}_{k+1} \leftarrow \boldsymbol{Y}_k \cup \boldsymbol{f}(\boldsymbol{X}_k^B)$
Approximate the Pareto front $\mathcal{P}_f$ by non-dominated solutions in $\boldsymbol{Y}_K$

Algorithm 2: Data Augmentation

input : all already-evaluated solutions in the k-th iteration $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$, number of real samples T
output: solutions after data augmentation $\boldsymbol{X}_k^{DP}$

$Fitness \leftarrow$ Calculate the fitness of each solution in $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ by Eq. 3
$\boldsymbol{X}_k^{real} \leftarrow$ Select top T candidate solutions with maximal $Fitness$
$\boldsymbol{X}_k^{fake} \leftarrow \boldsymbol{X}_k \setminus \boldsymbol{X}_k^{real}$
$\boldsymbol{\mu} \leftarrow$ mean($\boldsymbol{X}_k^{real}$) /*Mean vector*/
$\boldsymbol{\Sigma} \leftarrow$ cov($\boldsymbol{X}_k^{real}$) /*Covariance matrix*/
$\boldsymbol{X}_k^{input} \leftarrow \{X_k^{real} \cup X_k^{fake}, \{real\}^{T\times 1} \cup \{fake\}^{(len(\boldsymbol{X}_k)-T)\times 1}\}$ /* $\boldsymbol{X}_k^{input}$ is a tuple composed of decision variables and labels*/
Train a GAN with $\boldsymbol{X}_k^{input}$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and save its $generator$ G /*Training details can be found in Appendix A.1*/
$\boldsymbol{X}_k^{GAN} \leftarrow$ Generate solutions by $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{X}_k^{real}$ and G
$\boldsymbol{X}_k^{DP} \leftarrow DP_{model}(\boldsymbol{X}_k^{GAN}, \boldsymbol{X}_k^{real})$ /*Training and selection details can be found in Appendix A.2*/

Generation via GANs. The mean vector and covariance matrix of $real$ samples are computed as $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. We utilize a combination of $real$ and $fake$ samples with their corresponding labels, along with a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, as the training data of GANs. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is also the input of the $generator$ G. We subsequently obtain solutions $\boldsymbol{X}_k^{GAN}$, where $|\boldsymbol{X}_k^{GAN}| = 1000$, for all problems. However, it is worth noting that there is a risk of model collapse during the training of GANs (Arjovsky, Chintala, and Bottou 2017), which may result in

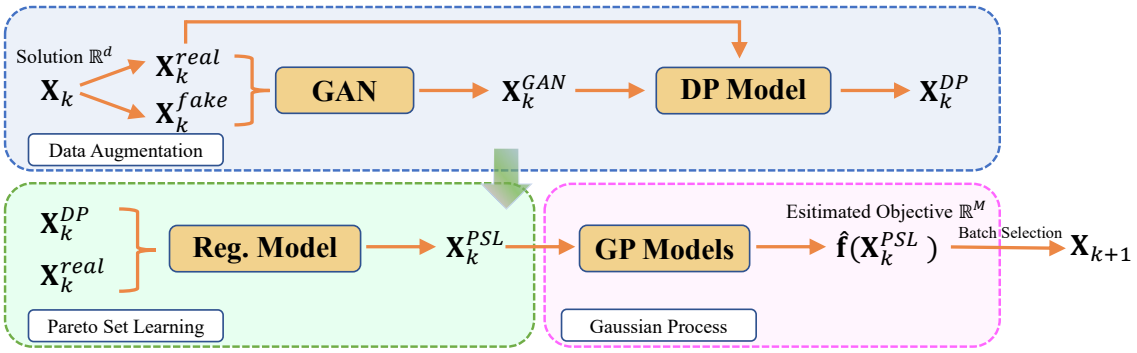

Figure 3: Flow Chart of Data-Augmented Pareto Set Learning for MOBO. (a) Data Augmentation. The already-evaluated solutions are partitioned into $real$ and $fake$ samples, and a multivariate normal distribution is constructed using the $real$ samples. The aforementioned data is utilized to train Generative Adversarial Networks (GANs), and new samples are generated from the $generator$. The trained dominance prediction model (DP Model) subsequently selects a batch of promising samples using a voting-score strategy. (b) Pareto Set Learning. The promising samples are employed to construct a regularity model via the local principal component analysis algorithm (Kambhatla and Leen 1997). The regularity model approximates the true Pareto set, and new solutions are sampled in the approximate Pareto set. (c) Batch Selection Strategy. The expected hypervolume improvement (EHVI) is employed as the acquisition function for batch selection. Furthermore, sequential greedy selection is utilized as an approach for batch selection to conserve computing resources.

generating suboptimal solutions. In order to mitigate this risk, we propose to add several samples as complements to GANs, which are generated by genetic operators such as crossover and mutation. Further details on the architecture and training process can be found in Appendix A.1.

Discrimination via Dominance Prediction Model. It is favorable to screen out the high-quality solutions from G for better data augmentation. Specifically, we utilize $\boldsymbol{X}_k^{real}$ as reference solutions to evaluate $\boldsymbol{X}_k^{GAN}$. The relationship between each solution of $\boldsymbol{X}_k^{real}$ and $\boldsymbol{X}_k^{GAN}$, i.e., whether it is dominated, non-dominated, or dominates other solutions, is crucial for selection. We use a voting-score strategy and reserve the top $|\boldsymbol{X}_k^{DP}|$ solutions with higher scores to build the Pareto set learning model, where $|\boldsymbol{X}_k^{DP}|$ is set to 100. More details can be found in Appendix A.2.

Discussion. Several data augmentation methods have been introduced for tackling MOPs. Both GMOEA (He et al. 2020) and DDMOPSO-A (Zhang et al. 2021) utilize GANs for data augmentation. GMOEA lacks customization for EMOPs, as it primarily targets general MOPs. Additionally, DDMOPSO-A is less suitable for solving EMOPs since it is an offline data-driven algorithm where newly evaluated solutions are not included in data augmentation. DK-RVEA (Liu and Wang 2022) proposes an unconventional data augmentation strategy by reusing old samples. Thus, it is unable to generate sufficient training data for the PSL model.

In contrast to the existing methods, our proposed data augmentation for PSL differs in two aspects: 1) For GANs, we introduce an imbalanced distribution of the $real$ and $fake$ samples in a greedy manner. In addition, the input latent variables of $generator$ follow a $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution in the $real$ samples distribution. 2) A lightweight dominance prediction (DP) model is adopted for selecting high-quality samples. DP exhibits better stability than other classification models (e.g., CSEA, MCEA/D (Sonoda and Nakata 2022)), as it considers the relationship between each generated solution and all the already-evaluated solutions.

Pareto Set Learning with Regularity Model

Model Construction. In DA-PSL, we utilize a regularity model from (Zhang, Zhou, and Jin 2008) as the Pareto set learning model. We choose to employ eigendecomposition (Kambhatla and Leen 1997) to partition $\boldsymbol{X}_k^{DP} \cup \boldsymbol{X}_k^{real}$, as opposed to the K-means clustering method (Hartigan and Wong 1979). This is due to that the centroid of each cluster should be a $(M-1)$-dimensional hyper-rectangle rather than a point, as indicated in the below property:

Theorem 1 (Claus Hillermeier (Hillermeier 2001)) *Under certain (mild) conditions, the PF and PS of a continuous MOP both distribute on a piecewise continuous $(M-1)$-dimensional manifold, respectively.*

Next, further details are presented. Initially, we partition $\boldsymbol{X}_k^{DP} \cup \boldsymbol{X}_k^{real}$ into P disjoint clusters S^1, S^2, ..., S^P based on Euclidean distance from each individual to the center of the clusters. This process is performed iteratively until each group is no longer updated. Subsequently, we calculate the mean values for each cluster:

$$\bar{\boldsymbol{x}}^j = \frac{1}{|S^j|} \sum_{\boldsymbol{x} \in S^j} \boldsymbol{x} \tag{4}$$

Then the covariance matrix C^j for each cluster is calculated and the first $M-1$ principal components (Hastie and Stuetzle 1989) of C^j are preserved:

$$\begin{aligned} C^j &= \frac{1}{|S^j|-1} \sum_{\boldsymbol{x} \in S^j} (\boldsymbol{x} - \bar{\boldsymbol{x}}^j)(\boldsymbol{x} - \bar{\boldsymbol{x}}^j)^T \\ U &= U_1^j, U_2^j, ..., U_{M-1}^j \end{aligned} \tag{5}$$

Algorithm 3: Pareto Set Learning

input : solutions selected by DP_{model} $\boldsymbol{X}_k^{DP}$, real samples $\boldsymbol{X}_k^{real}$
output: solutions generated by PSL model $\boldsymbol{X}_k^{PSL}$

$PS_{model} \leftarrow$ LocalPCA($\boldsymbol{X}_k^{DP} \cup \boldsymbol{X}_k^{real}$);
$\boldsymbol{X}_k^{PSL} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** N **do**
  Use PS_{model} to generate one solution, namely s;
  $\boldsymbol{X}_k^{PSL} \leftarrow \boldsymbol{X}_k^{PSL} \cup \{s\}$;

where U_i^j is an eigenvector associated with the i-th largest eigenvalue of C^j. The projection Φ^j of S^j is calculated:

$$\Phi^j = (\boldsymbol{x} - \bar{\boldsymbol{x}}^j)^T U^j, \Phi^j \in [a^j, b^j] \tag{6}$$

where a^j and b^j denote the lower and upper boundaries of Φ_j. Through the above processing, an $(M-1)$-dimensional Pareto set learning model Ψ^j for S^j is defined as follows:

$$\Psi^j = \{\boldsymbol{x} \in \mathbb{R}^d | \boldsymbol{x} = \bar{\boldsymbol{x}}^j + \sum_{i=1}^{M-1} \eta_i^j U_i\} \tag{7}$$

where η_i^j is randomly sampled from $[a_i^j - \frac{1}{4}(b_i^j - a_i^j)$, $b_i^j + \frac{1}{4}(b_i^j - a_i^j)]$, the purpose of 50% extension along the boundaries of Ψ^j is to provide a better promising approximation to the whole PS.

Solutions generation. We sample $|\boldsymbol{X}_k^{PSL}|$ solutions on the Pareto set learning model Ψ^j according to the proportion of individuals in each cluster, where $|\boldsymbol{X}_k^{PSL}|$ is set to 1000 for all problems:

$$\boldsymbol{x} = \bar{\boldsymbol{x}}^j + \sum_{i=1}^{M-1} \eta_i^j U_i + \gamma, \quad \gamma = \sum_{i=M}^{d} \lambda_i^j U_i^j \epsilon_i \tag{8}$$

where ϵ_i and λ_i ($i = M$ to d) denote independent $\mathcal{N}(0,1)$ noises and the $d - M + 1$ smallest eigenvalues, respectively, so that γ is a small perturbation for better diversity.

Discussion. As mentioned previously, the approximation accuracy of the PSL model in PSL-MOBO may not be satisfactory on some complicated problems. This phenomenon may be attributed to the instability of the PSL model in learning preference mapping and the Pareto set features. Different from PSL-MOBO, we introduce the regularity model (RM) for Pareto set learning in MOBO due to two reasons: 1) RM exhibits superior stability in optimization, particularly when solving complicated problems; 2) RM is more computationally efficient compared to PSL-MOBO. Additional details are presented in Appendix E.1.

Selection Strategy

Gaussian Process. We construct independent Gaussian Process (GP) models to model each objective (Daulton, Balandat, and Bakshy 2020). For a single-objective GP model, there exists a prior distribution, which can be defined as $f(\boldsymbol{x}) \sim \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}))$, where m and k denote the mean function $m : \mathcal{X} \subset \mathbb{R}$ and the kernel function $k : \mathcal{X} \times \mathcal{X} \subset \mathbb{R}$, respectively. The GP posterior distribution is updated by maximizing the log marginal likelihood $\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the parameters of the kernel function, with the already-evaluated solutions $\{\boldsymbol{X}, \boldsymbol{Y}\}$. Finally, we obtain the posterior distribution of the GP as $f(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}, \boldsymbol{x}))$, where the kernel vector is $\boldsymbol{\mu}(\boldsymbol{x}) = m(\boldsymbol{x}) + \boldsymbol{k}\boldsymbol{K}^{-1}\boldsymbol{Y}$ and the kernel matrix is $\boldsymbol{\Sigma}(\boldsymbol{X}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}\boldsymbol{K}^{-1}\boldsymbol{k}^T$, with $\boldsymbol{k} = k(\boldsymbol{x}, \mathcal{X})$ and $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X})$ (we apply the Matern 5/2 kernel in our method).

Batch Selection. We select a small subset $\boldsymbol{X}_k^B = \{x_b | b = 1, ..., B\}$ from the solutions sampled by the PSL model according to the hypervolume (HV) indicator (Zitzler and Thiele 1999). Given a solution set $S \subset \mathbb{R}^d$ and a reference vector set $r \subset \mathbb{R}^d$, $HV(S)$ is defined as follows:

$$\textbf{HV}(S) = \Lambda(\{q \in \mathbb{R}^d | \exists p \in S : p \leq q \textbf{ and } q \leq r\}) \tag{9}$$

where $\Lambda(\cdot)$ denotes the Lebesgue measure. Our batch selection strategy aims to maximize the hypervolume improvement (**HVI**) of $\boldsymbol{X}_k^B$ with respect to $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$:

$$\begin{aligned}\textbf{HVI}(\hat{f}(\boldsymbol{X}_k^B), \boldsymbol{Y}_k) =& \textbf{HV}(\boldsymbol{Y}_k \cup \hat{f}(\boldsymbol{X}_k^B)) \\ &- \textbf{HV}(\hat{f}(\boldsymbol{X}_k^B))\end{aligned} \tag{10}$$

where $\hat{f}(\cdot)$ represents the estimated objective value. To save computing resources, we adopt the widely-used sequential greedy selection in maximizing Eq. 10. The selected subset $\boldsymbol{X}_k^B$ is evaluated by true FE. The resulting $\{\boldsymbol{X}_k^B, \boldsymbol{Y}_k^B\}$ is unioned with $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ to obtain $\{\boldsymbol{X}_{k+1}, \boldsymbol{Y}_{k+1}\}$.

Experiments and Results

Synthetic Benchmarks and Real-world Applications. At first, we conduct a series of experiments on several widely-used synthetic multi-objective benchmarks, including ZDT1-3 (Zitzler, Deb, and Thiele 2000) and DTLZ2-7 (Deb et al. 2005). The considered problems in the experiments have 2 and 3 objectives, and the number of decision variables varies from 10 to 50. Notably, we present the results for a specific instance where $d = 20$, and additional results can be found in Appendix E.2. Subsequently, we further assessed the scalability of DA-PSL by adopting seven real-world engineering design problems.

Baselines and Implementation. Several SOTA and classical methods are considered for experimental comparison, including NSGA-II (Deb et al. 2002), MOEA/D-EGO (Zhang et al. 2009), TSEMO (Bradford, Schweidtmann, and Lapkin 2018), USeMO-EI (Belakaria et al. 2020), DGEMO (Konakovic Lukovic, Tian, and Matusik 2020), PSL-MOBO (Lin et al. 2022), qNparEGO (Knowles 2006), qEHVI (Daulton, Balandat, and Bakshy 2020) and qNEHVI (Daulton, Balandat, and Bakshy 2021). The implementations come from PSL-MOBO and DGEMO's open-source link, and BoTorch (Balandat et al. 2020). We implement DA-PSL [1] in Pytorch.

[1] https://github.com/ilog-ecnu/DAPSL

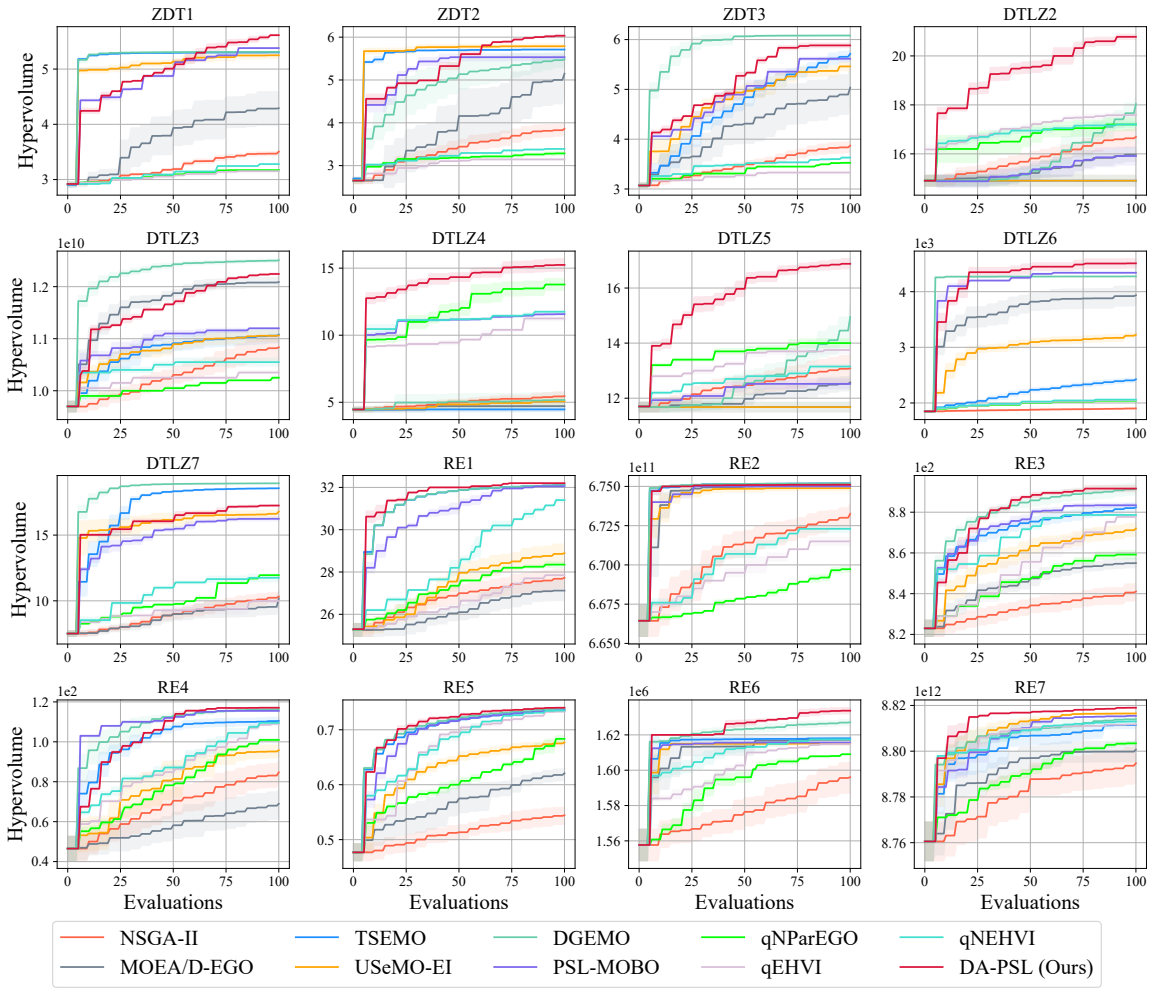

Figure 4: The HV values w.r.t. the number of FEs of 10 algorithms on synthetic test functions and real-world applications.

Experiment Settings. We initialize 100 solutions via Latin hypercube sampling for true FEs and proceed to execute BO for 20 batches with batch size 5 for each algorithm. Additionally, we ensure statistical rigor by running all methods 10 times independently. The HV indicator (Eq. 9) is used for evaluating the quality of 200 solutions in total. To ensure fair evaluation, we adopt the same reference point for each algorithm. More details are shown in Appendix B.

Experimental Results And Further Discussion

MOBO Performance. Figure 4 presents a comparison of the hypervolume indicator with respect to FE. Our proposed DA-PSL approach exhibits superior performance on most synthetic benchmarks in terms of both convergence speed and final values. Moreover, DA-PSL achieves excellent performance on real-world applications (RE) as well. Notably, DA-PSL outperforms PSL-MOBO, a recently proposed Pareto set learning method, on all types of problems. These results clearly establish the validity and superiority of our DA-PSL approach. Further details, including running time, ablation study, and additional experiments and discussions are shown in Appendix E.1, E.5, E.6, E.7, etc.

Validity of Data-Augmented Pareto Set Learning. The Pareto fronts approximated by DA-PSL and PSL-MOBO under the posterior mean are presented in Figure 5. It is evident that DA-PSL can better capture the salient characteristics of the ground truth PF, compared to PSL-MOBO, across synthetic benchmarks as well as real-world applications. For instance, PSL-MOBO only approximates a portion of the ground truth PF, whereas DA-PSL can effectively capture almost all the features of ZDT1. Moreover, our approach demonstrates satisfactory exploitation on the rocket injector design problem (RE5) (Vaidyanathan et al. 2003), which has a complex PF due to the Pareto optimal solutions being grouped into several regions, while PSL-MOBO only achieves good convergence on a few regions of RE5. Additionally, we include the non-dominated solutions without data augmentation strategy, which clearly fail to achieve satisfactory convergence and diversity on many problems, similar to PSL-MOBO. Further, we provide a detailed analysis of the superiority of DA-PSL in Appendix E.3.

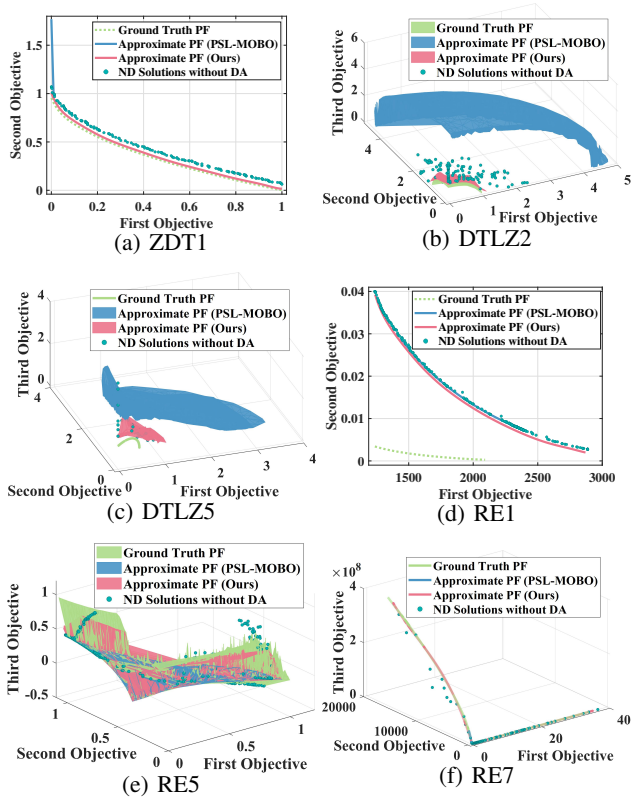

Figure 5: Approximate Pareto fronts obtained by DA-PSL and PSL-MOBO, and non-dominated solutions without data augmentation strategy.

Related Work

Multi-objective Bayesian Optimization (MOBO) is a powerful tool that combines the principles of Bayesian optimization and multi-objective optimization to solve multi-objective problems. One popular MOBO strategy is the random scalarization algorithm, which scalarizes the multi-objective problem into a distribution of single-objective problems. These approaches reduce the difficulty of optimization and apply classical single-objective BO to deal with, such as ParEGO (Knowles 2006) and TS-TCH (Paria, Kandasamy, and Póczos 2020). However, an intuitive weakness of these methods is that their performance is dependent on the quality of random scalarization (Daulton et al. 2022). Another effective strategy is based on efficient acquisition functions, such as the expected hypervolume improvement (EHVI) and predictive entropy search (PES). EHVI is regarded as an extension of the popular expected improvement (EI) (Močkus 1975b), which aims to produce a satisfactory coverage of the true Pareto front. PES is an alternative to EHVI and shows competitive performance. The representative algorithms include PESMO (Hernández-Lobato et al. 2016), MESMO (Belakaria, Deshwal, and Doppa 2019), and PFES (Suzuki et al. 2020).

There are two streams of works related to EHVI, referred to as sequential selection (Emmerich, Giannakoglou, and Naujoks 2006) and batch selection (Zhang et al. 2009). Sequential selection methods choose one solution for true function evaluation (FE) and update the surrogate model during each iteration. They usually tend to achieve better performance at the expense of time. Although a series of works focus on efficient EHVI computation in sequential selection (Hupkens et al. 2015), the time complexity increases with the number of objectives and the dimension of decision variables (Yang et al. 2019).

Batch selection methods usually choose a batch of solutions in a sequential greedy manner for true function evaluation (FE) before updating the surrogate model. Instead of directly selecting B solutions, the sequential greedy approach integrates over the posterior of the unevaluated samples corresponding to the already selected ones in the current batch and adds the best one into the batch one by one, greatly reducing time complexity with empirical and theoretical guarantees (Wu and Frazier 2016). DGEMO considers the diversity of both decision and objective space in the batch selection strategy. qNEHVI scales EHVI to highly parallel evaluations of noisy objectives. PSL-MOBO proposes a Pareto set learning model to map preference to its corresponding Pareto solution.

Pareto Set Learning (PSL) is an effective method for EMOPs from an infinite set approximation perspective, while most current approaches attach importance to a finite Pareto optimal set (Sazanovich et al. 2021). Most approaches focus on each solution itself and treat each of them as an independent individual. Nonetheless, the relationship between solutions provides crucial information for approximating the ground truth Pareto front (PF) and Pareto set (PS). Inspired by this phenomenon, PSL-MOBO incorporates preference information into deep neural networks and utilizes a Gaussian process model to predict the objectives of a given black-box function. Furthermore, PSL-MOBO updates model parameters with gradient descent and ultimately constructs a Pareto set model, which is regarded as a generalization of the decomposition-based multi-objective optimization algorithm (MOEA/D) (Zhang and Li 2007).

Conclusion

Given a limited number of function evaluations, accurately approximating the true Pareto set is a significant challenge due to data deficiency. In this study, we present a novel data-augmented Pareto set learning approach to expensive multi-objective optimization. To be specific, the data augmentation component includes two neural networks: GANs for generating samples and a dominance prediction model for screening samples. Additionally, we generalize the regularity model for Pareto set learning in expensive multi-objective Bayesian optimization. The core advantage of our approach lies in providing a sufficient number of high-quality samples for Pareto set learning, which differs from previous methods. Experimental results on both synthetic benchmarks and real-world applications illustrate that our proposed method outperforms classical and state-of-the-art approaches on most of the problems.

Acknowledgments

This work is supported by the Science and Technology Commission of Shanghai Municipality Grant (No. 22511105901).

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Baia, A. E.; Biondi, G.; Franzoni, V.; Milani, A.; and Poggioni, V. 2022. Lie to me: shield your emotions from prying software. *Sensors*, 22(3): 967.

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33: 21524–21538.

Belakaria, S.; Deshwal, A.; and Doppa, J. R. 2019. Max-value entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 32.

Belakaria, S.; Deshwal, A.; Jayakodi, N. K.; and Doppa, J. R. 2020. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10044–10052.

Bradford, E.; Schweidtmann, A. M.; and Lapkin, A. 2018. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2): 407–438.

Chen, T.; and Li, M. 2021. Multi-objectivizing software configuration tuning. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 453–465.

Couckuyt, I.; Deschrijver, D.; and Dhaene, T. 2014. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60: 575–594.

Daulton, S.; Balandat, M.; and Bakshy, E. 2020. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33: 9851–9864.

Daulton, S.; Balandat, M.; and Bakshy, E. 2021. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34: 2187–2200.

Daulton, S.; Eriksson, D.; Balandat, M.; and Bakshy, E. 2022. Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, 507–517. PMLR.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197.

Deb, K.; Thiele, L.; Laumanns, M.; and Zitzler, E. 2005. *Scalable test problems for evolutionary multiobjective optimization*. Springer.

Ding, D.; Li, D.; Li, Z.; and Yang, L. 2019. Compact circularly-polarized microstrip antenna for hand-held RFID reader. In *2019 8th Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 181–182. IEEE.

Emmerich, M. T.; Giannakoglou, K. C.; and Naujoks, B. 2006. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4): 421–439.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

Hastie, T.; and Stuetzle, W. 1989. Principal curves. *Journal of the American Statistical Association*, 84(406): 502–516.

He, C.; Huang, S.; Cheng, R.; Tan, K. C.; and Jin, Y. 2020. Evolutionary multiobjective optimization driven by generative adversarial networks (GANs). *IEEE transactions on cybernetics*, 51(6): 3129–3142.

Hernández-Lobato, D.; Hernandez-Lobato, J.; Shah, A.; and Adams, R. 2016. Predictive entropy search for multi-objective bayesian optimization. In *International conference on machine learning*, 1492–1501. PMLR.

Hillermeier, C. 2001. *Nonlinear multiobjective optimization: a generalized homotopy approach*, volume 135. Springer Science & Business Media.

Hoffman, M. W.; and Ghahramani, Z. 2015. Output-space predictive entropy search for flexible global optimization. In *NIPS workshop on Bayesian Optimization*, 1–5.

Hupkens, I.; Deutz, A. H.; Yang, K.; and Emmerich, M. T. 2015. Faster Exact Algorithms for Computing Expected Hypervolume Improvement. In *EMO (2)*, 65–79.

Jablonka, K. M.; Jothiappan, G. M.; Wang, S.; Smit, B.; and Yoo, B. 2021. Bias free multiobjective active learning for materials design and discovery. *Nature communications*, 12(1): 2312.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4): 455.

Kambhatla, N.; and Leen, T. K. 1997. Dimension reduction by local principal component analysis. *Neural computation*, 9(7): 1493–1516.

Knowles, J. 2006. ParEGO: A hybrid algorithm with online landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66.

Konakovic Lukovic, M.; Tian, Y.; and Matusik, W. 2020. Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems*, 33: 17708–17720.

Laumanns, M.; and Ocenasek, J. 2002. Bayesian optimization algorithms for multi-objective optimization. In *Parallel Problem Solving from Nature—PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings 7*, 298–307. Springer.

Li, M.; Yang, S.; and Liu, X. 2013. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3): 348–365.

Li, M.; Yang, S.; and Liu, X. 2014. Diversity comparison of Pareto front approximations in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12): 2568–2584.

Lin, X.; Yang, Z.; Zhang, X.; and Zhang, Q. 2022. Pareto Set Learning for Expensive Multi-Objective Optimization. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Liu, Z.; and Wang, H. 2022. A data augmentation based Kriging-assisted reference vector guided evolutionary algorithm for expensive dynamic multi-objective optimization. *Swarm and Evolutionary Computation*, 75: 101173.

Lu, Z.; Whalen, I.; Boddeti, V.; Dhebar, Y.; Deb, K.; Goodman, E.; and Banzhaf, W. 2019. Nsga-net: neural architecture search using multi-objective genetic algorithm. In *Proceedings of the genetic and evolutionary computation conference*, 419–427.

McKay, M. D.; Beckman, R. J.; and Conover, W. J. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1): 55–61.

Močkus, J. 1975a. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*, 400–404. Springer.

Močkus, J. 1975b. Optimization Techniques IFIP Technical Conference.

Paria, B.; Kandasamy, K.; and Póczos, B. 2020. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, 766–776. PMLR.

Sazanovich, M.; Nikolskaya, A.; Belousov, Y.; and Shpilman, A. 2021. Solving black-box optimization challenge via learning search space partition for local Bayesian optimization. In *NeurIPS 2020 Competition and Demonstration Track*, 77–85. PMLR.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Sonoda, T.; and Nakata, M. 2022. Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multiobjective problems. *IEEE Transactions on Evolutionary Computation*, 26(6): 1581–1595.

Suzuki, S.; Takeno, S.; Tamura, T.; Shitara, K.; and Karasuyama, M. 2020. Multi-objective Bayesian optimization using Pareto-frontier entropy. In *International Conference on Machine Learning*, 9279–9288. PMLR.

Tian, Y.; Si, L.; Zhang, X.; Cheng, R.; He, C.; Tan, K. C.; and Jin, Y. 2021. Evolutionary large-scale multi-objective optimization: A survey. *ACM Computing Surveys (CSUR)*, 54(8): 1–34.

Vaidyanathan, R.; Tucker, K.; Papila, N.; and Shyy, W. 2003. Cfd-based design optimization for single element rocket injector. In *41st Aerospace sciences meeting and exhibit*, 296.

Wu, J.; and Frazier, P. 2016. The parallel knowledge gradient method for batch Bayesian optimization. *Advances in neural information processing systems*, 29.

Xie, R.; Liu, Y.; Zhang, S.; Wang, R.; Xia, F.; and Lin, L. 2021. Personalized approximate pareto-efficient recommendation. In *Proceedings of the Web Conference 2021*, 3839–3849.

Yang, K.; Emmerich, M.; Deutz, A.; and Bäck, T. 2019. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75: 3–34.

Yang, P.; Zhang, L.; Liu, H.; and Li, G. 2023. Reducing Idleness in Financial Cloud via Multi-objective Evolutionary Reinforcement Learning based Load Balancer. *SCIENCE CHINA Information Sciences*.

Yu, P.-L. 1974. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. *Journal of optimization Theory and Applications*, 14: 319–377.

Yu, Z.; Ramakrishnan, V.; and Meinzer, C. 2019. Simulation optimization for bayesian multi-arm multi-stage clinical trial with binary endpoints. *Journal of Biopharmaceutical Statistics*, 29(2): 306–317.

Zhang, Q.; and Li, H. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6): 712–731.

Zhang, Q.; Liu, W.; Tsang, E.; and Virginas, B. 2009. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3): 456–474.

Zhang, Q.; Zhou, A.; and Jin, Y. 2008. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1): 41–63.

Zhang, Y.; Hu, W.; Qi, Y.; and Li, Y. 2021. Data-driven Multiobjective Particle Swarm Optimization based on Data Augmentation Strategy. In *2021 9th International Conference on Communications and Broadband Networking*, 6–10.

Zitzler, E.; Deb, K.; and Thiele, L. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2): 173–195.

Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4): 257–271.