Do We Really Need to Approach the Entire Pareto Front in Many-Objective Bayesian Optimisation?

Chao Jiang *cxj249@student.bham.ac.uk*
School of Computer Science
University of Birmingham

Jingyu Huang *j.huang.4@bham.ac.uk*
School of Mathematics
University of Birmingham

Miqing Li* *m.li.8@bham.ac.uk*
School of Computer Science
University of Birmingham

Reviewed on OpenReview: *https://openreview.net/forum?id=ZsI4bmqUD8*

Abstract

Many-objective optimisation, a subset of multi-objective optimisation, involves optimisation problems with more than three objectives. As the number of objectives increases, the number of solutions needed to adequately represent the entire Pareto front typically grows substantially. This makes it challenging, if not infeasible, to design a search algorithm capable of effectively exploring the entire Pareto front. This difficulty is particularly acute in the Bayesian optimisation paradigm, where sample efficiency is critical and only a limited number of solutions (often a few hundred) are evaluated. Moreover, after the optimisation process, the decision-maker eventually selects just one solution for deployment, regardless of how many high-quality, diverse solutions are available. In light of this, we argue an idea that under a very limited evaluation budget, it may be more useful to focus on finding a single solution of the highest possible quality for the decision-maker, rather than aiming to approximate the entire Pareto front as existing many-/multi-objective Bayesian optimisation methods typically do. Bearing this idea in mind, this paper proposes a single point-based multi-objective search framework (SPMO) that aims to improve the quality of solutions along a direction that leads to a good tradeoff between objectives. Within SPMO, we present a simple acquisition function, called expected single-point improvement (ESPI), working under both noiseless and noisy scenarios. We show that ESPI can be optimised effectively with gradient-based methods via the sample average approximation (SAA) approach and theoretically prove its convergence guarantees on acquisition optimisation under the SAA. We also empirically demonstrate that the proposed SPMO is computationally tractable and outperforms state-of-the-arts on a wide range of benchmark and real-world problems.

1 Introduction

Multi-objective optimisation problems (MOPs) (Emmerich & Deutz, 2018; Zheng & Wang, 2024) involve scenarios where multiple objectives need to be optimised simultaneously. Unlike single-objective optimisation problems which typically have a single optimal solution, in MOPs there is a set of optimal solutions known as Pareto optimal solutions. The corresponding points in the objective space form what is known as the Pareto front. In general, a multi-objective optimisation algorithm aims to generate a set of solutions that

*Correspondence to: Miqing Li <m.li.8@bham.ac.uk>

well approximate the Pareto front, from which the decision-maker chooses a solution to deploy based on their preferences.

In modern applications, optimisation is becoming increasingly complex, with a growing number of requirements and objectives that need to be considered at the same time (Matrosov et al., 2015; Hierons et al., 2020; Lin et al., 2025). Taking the car cab design as an example, there are up to nine objectives to be optimised (Deb & Jain, 2013), including cabin space, fuel efficiency, acceleration time, and road noise at various speeds. This has given rise to a new research topic, many-objective optimisation, focusing on MOPs involving more than three objectives (Ishibuchi et al., 2008; Li et al., 2015a).

At the same time, many real-world multi-/many-objective optimisation problems are black-box and costly in terms of solution evaluation. This is evident across a range of fields, including chemistry (Park et al., 2018; Shields et al., 2021; Dunlap et al., 2023), materials science (Liang et al., 2021; Low et al., 2024; Peng et al., 2025), and transportation (Deb & Jain, 2013; Jain & Deb, 2013a; Cheaitou & Cariou, 2019; Deb et al., 2009). For example, in vehicle design optimisation, it can take about 20 hours to evaluate a vehicle design (Youn et al., 2004; Daulton et al., 2021). To tackle such problems, multi-objective Bayesian optimisation (MOBO) is a very effective approach (Garnett, 2023), along with other alternatives (e.g., surrogate-assisted evolutionary algorithms (Jin, 2011; Liang et al., 2024)). Over the past decades, a variety of effective MOBO methods have emerged, including scalarisation-based methods (Knowles, 2006; Paria et al., 2020; Lin et al., 2022) which convert a multi-objective problem into a number of single-objective problems, and Pareto-based methods which consider Pareto dominance relations over objectives (Daulton et al., 2020; Emmerich et al., 2006; Tu et al., 2022). Most works aim to find a good approximation of the entire Pareto front.

However, with the increase of the objective number, the number of solutions needed to adequately represent the problem's Pareto front typically grows substantially. For example, for a 10-objective problem, it normally needs 220 points (i.e., $\binom{12}{3}$) even if only three divisions on each objective are considered (Das & Dennis, 1998). This difficulty is especially pronounced in Bayesian optimisation, where often only a few hundred solutions can be generated and evaluated. With such a limited budget, it is highly unlikely for an optimisation algorithm to reach or even be close to the Pareto front. On top of that, after the optimisation process, the decision-maker eventually selects just one solution to deploy, regardless of how many high-quality, diverse solutions are provided.

Given the above, this paper argues an idea that under a very limited evaluation budget, it may be more useful to focus on finding a single solution of the highest possible quality for the decision-maker, rather than aiming to approximate the entire Pareto front. That is, we may not need to care about diversifying solutions to represent the entire Pareto front, but focus on improving the quality of a single solution. Figure 1 illustrates this idea in a bi-objective case. As can be seen from the figure, in contrast to aiming for a set of diversified solutions which existing MOBO methods typically do (Li et al., 2025a; Lin et al., 2022), our method aims for a single solution with better convergence (i.e., closer to the Pareto front). Although it may yield a worse *hypervolume* (HV) value (Zitzler & Thiele, 1999) compared to the diverse solution set obtained by existing methods, it may be more likely to be chosen by the decision-maker as it achieves a more favourable trade-off among the objectives.

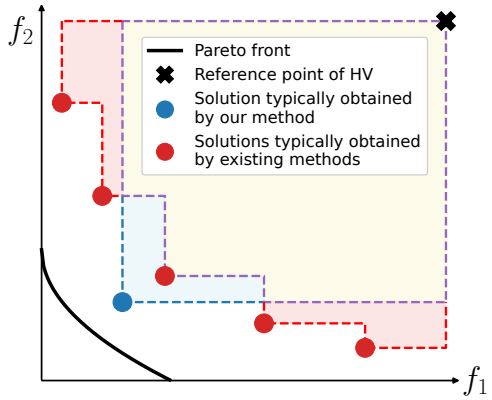


Figure 1: An illustration of our idea in a bi-objective case, in comparison with existing methods that aim to search for the entire Pareto front. The red points represent (nondominated) solutions that existing methods may obtain, and the blue point represents what our method aims for. It can be seen that the red points are much more diversified, hence, as a whole, having a better hypervolume (HV) value (Zitzler & Thiele, 1999). However, the blue point has better convergence (i.e., closer to the Pareto front) than any single red point, which may be more likely to be preferred by the decision-maker.

Bearing this idea in mind, this paper proposes a single point-based multi-objective search framework, called SPMO. The contributions of this work can be summarised as follows.

- We propose a novel MOBO search framework that does not aim to approximate the entire Pareto front, but rather focuses on improving the quality of solutions along a single direction that leads to a good tradeoff between objectives.
- Within SPMO, we present a simple acquisition function, called Expected Single-Point Improvement (ESPI). We show that ESPI can be optimised effectively with gradient-based methods via the sample average approximation (SAA) approach from Balandat et al. (2020) and also theoretically prove its theoretical convergence guarantees on acquisition optimisation under the SAA.
- We consider both noiseless and noisy cases, resulting in two versions of the proposed ESPI.
- We verify SPMO through an extensive experimental study, including in comparison with various state-of-the-arts, under both sequential and batch optimisation settings, through sensitivity analysis, with different metrics embedded, and on a range of benchmark and real-world problems.

2 Background and Related Work

2.1 Background

Many-Objective Optimisation. Many-objective optimisation, a subset of multi-objective optimisation, refers to an optimisation scenario having more than three objectives to be considered. Without loss of generality, this paper considers the problem of minimising a vector-valued function: $\boldsymbol{f}(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^m$, where $\boldsymbol{x} \in \mathcal{X}$ ($\mathcal{X} \subset \mathbb{R}^d$) and m is the number of objectives. In multi-/many-objective optimisation, a solution $\boldsymbol{x}_1$ is said to dominate $\boldsymbol{x}_2$, denoted by $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$, if $\forall i \in \{1, ..., m\}$, $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ and $\exists j \in \{1, ..., m\}$, $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$. If a solution $\boldsymbol{x}_1 \in \mathcal{X}$ is not dominated by any other solution, then $\boldsymbol{x}_1$ is said to be Pareto optimal. The collection of Pareto optimal solutions of a problem is called the Pareto set, and its mapping to the objective space is called Pareto front.

Bayesian Optimisation (BO). BO is a sample-efficient global optimisation approach that builds a probabilistic surrogate, typically a Gaussian process (GP), and uses an acquisition function $\alpha : \mathcal{X} \to \mathbb{R}$ to decide which points to evaluate. In this work, we model each objective with an independent Gaussian process $f_i \sim \mathcal{GP}(m_i(\boldsymbol{x}), k_i(\boldsymbol{x}, \boldsymbol{x}'))$, where $m_i(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is the ith mean function, and $k_i(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the ith covariance function. Given n observed points $\mathcal{D} = \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^n$ where $\boldsymbol{y}^{(t)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}) + \boldsymbol{\zeta}^{(t)}$ and the noise $\boldsymbol{\zeta}^{(t)} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{\sigma}_\zeta^2))$, the posterior distribution of the ith objective at a new location $\boldsymbol{x}$ is a Gaussian distribution: $f_i(\boldsymbol{x})|\mathcal{D} \sim \mathcal{N}(\mu_i(\boldsymbol{x}), \sigma_i^2(\boldsymbol{x}))$ where $\mu_i(\boldsymbol{x})$ and $\sigma_i^2(\boldsymbol{x})$ are the posterior mean and variance at $\boldsymbol{x}$, respectively. Detailed expressions of the posterior mean and variance are given in Appendix A.

2.2 Related Work

Over the past decades, various MOBO methods have been proposed (Konakovic Lukovic et al., 2020; Daulton et al., 2022a). They can be loosely divided into scalarisation-based and Pareto-based methods. In scalarisation-based methods (Knowles, 2006; Paria et al., 2020), a multi-objective problem is converted into a number of single-objective problems (Chugh, 2020). Hence, one can leverage acquisition functions (Lai & Robbins, 1985) from single-objective BO to decide which point to evaluate. For instance, ParEGO (Knowles, 2006), TS-TCH (Paria et al., 2020), and Pareto set learning (PSL) (Lin et al., 2022) use expected improvement (EI) (Jones et al., 1998), Thompson sampling (TS) (Thompson, 1933), and upper confidence bound (UCB) (Lai & Robbins, 1985), respectively. In contrast, Pareto-based methods consider Pareto dominance relations over objectives and directly solve a multi-objective problem without converting it into a number of single-objective subproblems (Emmerich et al., 2006; Tu et al., 2022). A popular idea is to use HV as maximising the HV value is equivalent to finding the entire Pareto front (Shang et al., 2020). Expected hypervolume improvement (EHVI) is commonly used in MOBO (Couckuyt et al., 2014; Daulton et al., 2020; 2021). Another idea is to leverage information theory to guide exploration toward regions likely contributing to the Pareto front. For

Algorithm 1: Single Point-based Multi-Objective (SPMO) Framework

Input: $\boldsymbol{f}$: Expensive black-box problem with m objectives; T: Maximum number of evaluations;
g: Metric that measures the quality of a single point; α: Acquisition function;
$\mathcal{D} := \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^{n_0}$: Initial observed solutions.

1 **for** $t = n_0 + 1 : T$ **do**
2 $\quad GPs \leftarrow$ Train $\mathcal{GP}s(\mathcal{D})$ `// Train` m `Gaussian process models`
3 $\quad \boldsymbol{x}^{(t)} \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}; g, GPs)$ `// Maximise the acquisition function based on the single-point quality metric` $g(\cdot)$
4 $\quad \boldsymbol{y}^{(t)} \leftarrow \boldsymbol{f}(\boldsymbol{x}^{(t)}) + \boldsymbol{\zeta}^{(t)}$ `// Evaluate the solution` $\boldsymbol{x}^{(t)}$
5 $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}$ `// Augment the observed solutions`

Output: $\mathcal{D}$: Observed solutions.

instance, joint entropy search (JES) (Tu et al., 2022) selects points that maximise the joint information gain for optimal inputs (i.e., the approximated Pareto set) and outputs (i.e., the approximated Pareto front). All of the above methods aim to approach the entire Pareto front.

It is worth noting that, similar to our approach, a few studies do not attempt to approximate the entire Pareto front. Some methods instead target a specific region of the front, such as the central area (Gaudrie et al., 2018; 2020; Binois et al., 2020). Another line of work incorporates decision-maker preferences by dynamically adjusting the target region based on elicited or updated preferences during optimisation (Abdolshah et al., 2019; Astudillo & Frazier, 2020; Ozaki et al., 2024; Ip et al., 2025). In contrast, our method mainly assumes no prior knowledge of decision-maker preferences and seeks to identify a high-quality trade-off solution across objectives. A more detailed discussion of related work can be found in Appendix C.

3 The Proposed Method

In this section, we first give the proposed MOBO framework. We then present the considered acquisition function (called ESPI), which is based on a simple distance-based metric. We note that analytically solving ESPI is not feasible, and thus consider its Monte Carlo approximation. Lastly, we consider ESPI under noisy cases, namely noisy ESPI (NESPI), and also its MC approximation.

Single Point-based Multi-Objective (SPMO) Framework. Algorithm 1 gives the procedure of the proposed SPMO framework. As can be seen, SPMO is very similar to a standard MOBO algorithm, except for the step of maximising the acquisition function based on a single-point quality metric (line 3). In principle, any metric that can reflect the quality of a solution in achieving a good trade-off between objectives can be adopted. This includes distance-based, scalarisation-based, and HV-based metrics, such as the weighted sum and augmented Tchebycheff (Miettinen, 1999) with a fixed weight vector (e.g., $(\frac{1}{m}, \ldots, \frac{1}{m}) \in \mathbb{R}^m$ in the m-objective case), as well as the HV of a single solution. Here, we use a simple distance-based metric as the default choice, and compare it with different metrics in our experiments (Section 6; details in Appendix F.5).

Single-Point Improvement (SPI). We consider the distance of solutions to a utopian point: $g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{z}^*) = \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^*\| = \sqrt{\sum_{i=1}^{m} \big(f_i(\boldsymbol{x}) - z_i^*\big)^2}$, where m denotes the number of objectives, and $\boldsymbol{z}^* = (z_1^*, z_2^*, \ldots, z_m^*)$ is a utopian point, i.e., $z_i^* \leq \min_{\boldsymbol{x} \in \mathcal{X}} f_i(\boldsymbol{x})$. In many real-world cases, the utopian value of an objective can be loosely estimated, for example, by assuming idealised conditions such as zero cost, time or error (Branke et al., 2008). In our experimental evaluation, we perform a sensitivity analysis on the choice of the utopian point, and it shows that substantially different settings can yield consistent results.

We first consider noiseless cases, i.e., $\bar{\boldsymbol{y}}^{(t)} = \boldsymbol{f}(\bar{\boldsymbol{x}}^{(t)})$. Let $\bar{\mathcal{D}} = \{(\bar{\boldsymbol{x}}^{(t)}, \bar{\boldsymbol{y}}^{(t)})\}_{t=1}^{n}$ be n observed points and $\bar{X} = \{\bar{\boldsymbol{x}}^{(t)}\}_{t=1}^{n}$ be the set of all the observed decision vectors in $\bar{\mathcal{D}}$. For any point $\boldsymbol{x} \in \mathcal{X}$, we define the single-point improvement (SPI) as:

$$I_{SP}(\boldsymbol{f}(\boldsymbol{x}); g^*, \boldsymbol{z}^*, \bar{\mathcal{D}}) = \max\Big(0,\, g^* - \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^*\|\Big), \tag{1}$$

where $g^* = \min_{\boldsymbol{x} \in \bar{X}} g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{z}^*)$.

Expected Single-Point Improvement (ESPI). We now present ESPI to account for the posterior distribution $p(\boldsymbol{f}|\bar{\mathcal{D}})$. Suppose that we independently model each objective f_i as a Gaussian process based on $\bar{\mathcal{D}}$. Then, the posterior of each f_i at a new location $\boldsymbol{x}$ is a Gaussian random variable, i.e., $f_i(\boldsymbol{x})|\bar{\mathcal{D}} \sim \mathcal{N}\big(\mu_i(\boldsymbol{x}), \sigma_i^2(\boldsymbol{x})\big)$, where $f_1, \ldots, f_m$ are mutually independent Gaussians. Let $\eta_i := f_i - z_i^*$. Then we obtain $\eta_i \sim \mathcal{N}\big(\mu_i(\boldsymbol{x}) - z_i^*, \sigma_i^2(\boldsymbol{x})\big)$ which is a Gaussian as well. The proposed ESPI is defined as:

$$\begin{aligned} \alpha_{\text{ESPI}}(\boldsymbol{x}) &= \mathbb{E}\big[I_{SP}(\boldsymbol{f}(\boldsymbol{x}); g^*, \boldsymbol{z}^*, \bar{\mathcal{D}})\big] \\ &= \mathbb{E}_{p(\boldsymbol{\eta})}\big[\max(0,\, g^* - \|\boldsymbol{\eta}\|)\big] \end{aligned} \tag{2}$$

An illustration of ESPI in a bi-objective case is given in Figure 6 of Appendix B for aiding understanding. Note that the integral in Eq. 2 cannot be solved analytically as it involves the distribution of $\|\boldsymbol{\eta}\|$, whose PDF and CDF have no closed-form expressions and are typically computed via numerical methods (Imhof, 1961; Ruben, 1962; Das, 2025). Hence, we use the MC integration with samples from the posterior $\tilde{\boldsymbol{f}}_t(\boldsymbol{x}) \sim p(\boldsymbol{f}(\boldsymbol{x})|\bar{\mathcal{D}})$ for $t = 1, \ldots, N$ to estimate Eq. 2:

$$\alpha_{\text{ESPI}}(\boldsymbol{x}) \approx \hat{\alpha}_{\text{ESPI}}(\boldsymbol{x}) = \frac{1}{N} \sum_{t=1}^{N} I_{SP}(\tilde{\boldsymbol{f}}_t(\boldsymbol{x}) | g^*, \boldsymbol{z}^*, \bar{\mathcal{D}}) \tag{3}$$

Noisy Expected Single-Point Improvement (NESPI). In the real world, it is not uncommon to encounter an optimisation problem with noises: $\boldsymbol{y}^{(t)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}) + \boldsymbol{\zeta}^{(t)}$, where $\boldsymbol{\zeta}^{(t)} \sim \mathcal{N}(\boldsymbol{0}, \text{diag}(\boldsymbol{\sigma}_\zeta^2))$. In noisy cases, simply using the observed best distance g^* may adversely affect the optimisation performance. Here, we present an extension of ESPI, i.e., noisy ESPI (NESPI). Let $\mathcal{D} = \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^n$ be n observed points and $X = \{\boldsymbol{x}^{(t)}\}_{t=1}^n$ be the set of all the observed decision vectors in $\mathcal{D}$. By considering the uncertainty in the function values at X, the proposed NESPI is defined as:

$$\alpha_{\text{NESPI}}(\boldsymbol{x}) = \int \alpha_{\text{ESPI}}(\boldsymbol{x}|\hat{g^*}) p(\boldsymbol{f}|\mathcal{D}) d\boldsymbol{f} \tag{4}$$

where $\hat{g^*}$ denotes the smallest distance to the utopian point over $\boldsymbol{f}(X)$. Note that in noiseless cases, NESPI is equivalent to ESPI. Additionally, ESPI and NESPI can be naturally extended to the parallel (batch) setting by using sequential greedy approximation (Balandat et al., 2020).

Like in the noiseless case, the integral in Eq. 4 is also analytically intractable but can be approximated using the MC integration. Let $\tilde{\boldsymbol{f}}_t(\boldsymbol{x}) \sim p(\boldsymbol{f}(\boldsymbol{x})|\mathcal{D})$ for $t = 1, \ldots, N$ be samples from the posterior, and let $\hat{g^*} = \min_{\boldsymbol{x} \in X} \|\tilde{\boldsymbol{f}}_t(\boldsymbol{x}) - \boldsymbol{z}^*\|$ be the smallest distance to utopian point over the previously evaluated points under the sampled function $\tilde{\boldsymbol{f}^{(t)}}(\boldsymbol{x})$. Then, $\alpha_{\text{NESPI}} \approx \frac{1}{N} \sum_{t=1}^{N} \alpha_{\text{ESPI}}(\boldsymbol{x}|\hat{g^*}, \boldsymbol{z}^*, \mathcal{D})$. Using the MC integration, the inner expectation in α_{NESPI} can be computed simultaneously using samples from the joint posterior $\tilde{\boldsymbol{f}}_t(\boldsymbol{x}, X) \sim p(f(\boldsymbol{x}, X)|\mathcal{D})$ over $\boldsymbol{x}$ and X:

$$\alpha_{\text{NESPI}}(\boldsymbol{x}) \approx \hat{\alpha}_{\text{NESPI}}(\boldsymbol{x}) = \frac{1}{N} \sum_{t=1}^{N} I_{SP}(\tilde{\boldsymbol{f}}_t; \hat{g^*}, \boldsymbol{z}^*, \mathcal{D}) \tag{5}$$

4 Optimising ESPI and NESPI

Having presented the MC estimators of ESPI and NESPI, we are now ready to optimise them.

Differentiability. The MC estimators of ESPI and NESPI ($\hat{\alpha}_{\text{ESPI}}(\boldsymbol{x})$ in Eq. 3 and $\hat{\alpha}_{\text{NESPI}}(\boldsymbol{x})$ in Eq. 5) are differentiable with respect to $\boldsymbol{x}$. We are able to automatically compute exact gradients of the MC estimators of ESPI and NESPI ($\nabla_{\boldsymbol{x}} \hat{\alpha}_{\text{ESPI}}(\boldsymbol{x})$ and $\nabla_{\boldsymbol{x}} \hat{\alpha}_{\text{NESPI}}(\boldsymbol{x})$) by leveraging the auto-differentiation in modern computational frameworks. This facilitates efficient gradient-based optimisation of ESPI and NESPI.

SAA Convergence Results. The sample average approximation (SAA) approach (Kleywegt et al., 2002), which addresses stochastic optimisation problems by using the MC simulation, has gained increasing popularity and has become a standard technique in BO for optimising MC-based acquisition functions (Balandat et al., 2020). By fixing the base samples, the SAA yields a deterministic acquisition function which enables using (quasi-) higher-order optimisation algorithms to obtain fast convergence rates for acquisition optimisation. We now give the theoretical convergence guarantees on acquisition optimisation under the SAA.

Theorem 4.1. *Suppose that $\mathcal{X}$ is compact and $\boldsymbol{f}$ has a multi-output GP prior whose mean and covariance functions are continuously differentiable. Let $\alpha^*_{ESPI} := \max_{\boldsymbol{x}\in\mathcal{X}} \alpha_{ESPI}(\boldsymbol{x})$ denote the maximum of ESPI, $S^* := \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_{ESPI}(\boldsymbol{x})$ denote the set of maximisers of α_{ESPI}, $\hat{\alpha}^N_{ESPI}(\boldsymbol{x})$ denote the deterministic function via the base samples $\{\epsilon^{(t)}\}_{t=1}^N \sim \mathcal{N}(0, I_m)$. Suppose $\hat{\boldsymbol{x}}^*_N \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \hat{\alpha}^N_{ESPI}(\boldsymbol{x})$, then*

(1) $\hat{\alpha}^N_{ESPI}(\hat{\boldsymbol{x}}^*_N) \to \alpha^*_{ESPI}$ a.s.

(2) $d(\hat{\boldsymbol{x}}^*_N, S^*) \to 0$ a.s., where $d(\hat{\boldsymbol{x}}^*_N, S^*) := \inf_{\boldsymbol{x}\in S^*} \|\hat{\boldsymbol{x}}^*_N - \boldsymbol{x}\|$.

The proof of the theorem is given in Appendix D.1. This theorem indicates that one is able to optimise the MC estimator of the acquisition function ESPI to obtain a solution that converges almost surely to the optimal solution of the original acquisition function. For the noisy case NESPI, the theorem of the theoretical convergence guarantees on acquisition optimisation under the SAA (Theorem D.1), together with its proof, is provided in Appendix D.2.

5 Experimental Design

Compared Methods. To evaluate the proposed SPMO, we consider eight MOBO methods. These include one baseline method (Sobol (Sobol, 1967)), five methods that aim to approximate the entire Pareto front, and two methods that aim at the trade-off region of the Pareto front. The five methods consist of three scalarisation-based methods, ParEGO (Knowles, 2006) (along with its noisy variant NParEGO (Daulton et al., 2021)), TS-TCH (Paria et al., 2020), and PSL (Lin et al., 2022), and two Pareto-based methods, EHVI (Daulton et al., 2020) (along with its noisy variant NEHVI (Daulton et al., 2021)), and JES (Tu et al., 2022). For methods that do not aim at the entire Pareto front, we consider TS-TCH (Paria et al., 2020) with fixed weights, denoted as TS-TCH$^+$, and C-EHVI (Gaudrie et al., 2018; 2020). TS-TCH$^+$ uses equal weights to search for a good trade-off solution, and C-EHVI prefers the central region of the Pareto front. We would like to see whether these two methods are competitive with our method for identifying a well-balanced solution. For all EI-based methods, i.e., ParEGO, NParEGO, EHVI, NEHVI, C-EHVI and SPMO, we use the log version as suggested by Ament et al. (2023). Note that we only consider NESPI in this work, as it is equivalent to ESPI under noiseless cases.[1]

Benchmarks and Real-World Problems. For benchmark problems, we first choose two most widely scalable functions, DTLZ1 and DTLZ2 (Deb et al., 2005). However, their Pareto fronts are rather homogeneous, i.e., with a simplex shape. We then include their inverted versions, i.e., inverted DTLZ1 and inverted DTLZ2 (Deb & Jain, 2013). These problems do not include the one with convex Pareto fronts nor different objective scales. We thus add convex DTLZ2 and scaled DTLZ2 (Jain & Deb, 2013a). We also give the results of other DTLZ problems which have different features (e.g., degenerate and disconnected Pareto fronts) (Deb et al., 2005; Cheng et al., 2017) in Appendix F.1; those problems are widely used in many-objective optimisation (Deb & Jain, 2013; Li et al., 2014b;a; 2015b). Each problem is considered with 3, 5 and 10 objectives, following the practice in Deb & Jain (2013); Jain & Deb (2013a); Li et al. (2014a), and tested under both noiseless and noisy cases. We also consider three well-studied real-world problems (Tanabe & Ishibuchi, 2020), i.e., car side impact design (CSI) (Jain & Deb, 2013a), water resource planning (WRP) (Ray et al., 2001), and car cab design (Deb & Jain, 2013). CSI and WRP are noise-free problems with four and six objectives, respectively. The car cab design problem has nine objectives and involves four stochastic variables (out of the total seven variables) that introduce noise into the optimisation process, making it a natural noisy optimisation problem.

[1]Although NEHVI is equivalent to EHVI under noiseless cases, the wall time of NEHVI is much higher than EHVI (see Table 28). Hence we consider EHVI and NEHVI under noiseless and noisy cases, respectively.

For the other problems without noise, to make their noise cases, we use the additive zero-mean Gaussian noise with a standard deviation of 0.1, as suggested in Hernandez-Lobato et al. (2016); Jiang & Li (2025b). The details of the problem formulations are given in Appendix E.3.

Performance Metrics. The proposed method aims to find a single trade-off solution between objectives which has a high chance of being favoured by the decision-maker, and therefore requires the use of appropriate metrics to fairly evaluate our method (Li & Yao, 2019; Li et al., 2022). As such, we first consider two single-point-based metrics, i.e., the distance-based metric used in the proposed method (reported as log-distance for better visualisation) and the HV-based metric (Zitzler & Thiele, 1999), which measures the HV contribution of a solution. Note that the distance-based metric is used in the normalised objective space. We expect that our method performs well on the distance-based metric as we directly optimise it. Notably, we are not certain whether our method performs best on the single-point HV since we do not directly optimise it. On the other hand, since most of the compared methods aim to achieve a good approximation of the entire Pareto front, we also consider the HV of the whole (nondominated) solution set obtained. It is expected that our method performs poorly, compared to other methods since we only optimise one point, rather than maximising the HV of the whole set (see Figure 1). For the reference point of the two HV metrics, we followed the practice in Ishibuchi et al. (2018); Balandat et al. (2020); Chugh (2020); Daulton et al. (2020) (see detailed settings in Appendix E.2).

Budget and Statistical Validation. For all the methods, we allow a maximum of 200 evaluations, following the practice in Daulton et al. (2020; 2021); Konakovic Lukovic et al. (2020). To enable statistical comparisons, each optimisation was repeated 30 times. We use the Wilcoxon rank-sum test (Wilcoxon, 1992) at a significance level of $\alpha = 0.05$ and Holm-Bonferroni correction (Holm, 1979) to see if our method differs significantly from each peer method.

6 Experimental Results

We first report the results under noiseless cases, then under noisy cases and under batch settings. Next, we perform the sensitivity analysis of the utopian point used. Then, we compare the proposed framework working with different single-point metrics, e.g., the distance-based, weighted sum-based, Tchebycheff-based, and HV-based. We further investigate the performance of these metrics when preference information is available. Lastly, we give the acquisition optimisation wall time for all the algorithms.

Noiseless Cases. We begin our evaluation by considering the distance metric, for which we expect good performance obtained by our SPMO. Table 1 shows the results of SPMO and the peer methods on the benchmark problems with five objectives and the two real-world problems. Unsurprisingly, as can be seen from the table, our method significantly outperforms the other methods on all the problems. In addition, to understand the anytime performance, Figure 2 presents the trajectories of the distance-based metric obtained by the nine methods. As seen, SPMO demonstrates a clear advantage over the other methods, obtaining a better convergence rate from the very beginning.

We now compare the methods by the HV metric of a single solution. Table 2 shows the results of the best solution (in terms of its HV value) obtained by SPMO and the peer methods. As can be seen, SPMO is still very competitive, performing best on all the problems except DTLZ2, where EHVI obtains the best HV. In addition, to evaluate the anytime behaviour, Figure 3 presents the trajectories of the HV of the best solution obtained by the nine methods. The results indicate that SPMO generally outperforms the other methods and converges faster from the early stages of the optimisation process. To get a sense of what such a best-HV solution looks like, we use a spider chart to plot the solution on the five-objective inverted DTLZ1 problem in Figure 4. As seen, the solution of SPMO has the largest area, and it actually Pareto dominates the solutions of the other methods (i.e., better or at least equal on all five objectives).

Lastly, we consider the HV results of all the solutions obtained by the compared methods (Table 3). Interestingly, although SPMO does not aim to approximate the entire Pareto front, it still gets fairly good results, outperforming the other methods on at least 3 out of the 7 problems. One explanation for this is that within very tight budget, searching for improving convergence of solutions may play a bigger part than searching for improving diversity, thus contributing more to the HV value.

Table 1: Results of the distance-based metric (log distance) obtained by the SPMO and the peer methods on the benchmark problems with 5 objectives and the two real-world problems on 30 runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	WRP Mean (Std)	Sum up +/∼/−
Sobol	3.7e+0 (3.0e−1)$^+$	2.4e−1 (4.6e−2)$^+$	4.8e+0 (3.1e−1)$^+$	6.0e−1 (5.7e−2)$^+$	-3.1e−1 (2.5e−1)$^+$	2.3e−1 (5.1e−2)$^+$	-1.9e−1 (2.5e−2)$^+$	-6.0e−1 (4.1e−2)$^+$	**8/ 0/ 0**
ParEGO	3.4e+0 (1.9e−1)$^+$	6.7e−2 (8.1e−2)$^+$	3.2e+0 (5.4e−1)$^+$	2.5e−1 (1.3e−2)$^+$	-1.7e+0 (2.0e−1)$^+$	1.3e−1 (8.9e−2)$^+$	-3.3e−1 (6.0e−3)$^+$	-7.0e−1 (2.3e−3)$^\sim$	**7/ 1/ 0**
TS-TCH	3.9e+0 (2.2e−1)$^+$	2.0e−1 (4.1e−2)$^+$	4.8e+0 (3.2e−1)$^+$	4.6e−1 (1.2e−2)$^+$	-7.1e−1 (2.1e−1)$^+$	2.6e−1 (3.8e−2)$^+$	-2.7e−1 (1.5e−2)$^+$	-6.7e−1 (2.9e−2)$^+$	**8/ 0/ 0**
TS-TCH$^+$	4.0e+0 (2.9e−1)$^+$	8.5e−2 (2.1e−2)$^+$	4.8e+0 (2.9e−1)$^+$	3.8e−1 (2.9e−2)$^+$	-9.2e−1 (3.3e−1)$^+$	8.0e−2 (2.3e−2)$^+$	-2.8e−1 (1.5e−2)$^+$	-6.9e−1 (6.5e−3)$^+$	**8/ 0/ 0**
EHVI	3.5e+0 (9.6e−2)$^+$	9.3e−3 (3.5e−3)$^+$	4.0e+0 (4.4e−1)$^+$	2.3e−1 (5.5e−3)$^+$	-1.4e+0 (2.1e−1)$^+$	3.1e−1 (6.0e−2)$^+$	-3.3e−1 (1.0e−2)$^+$	-6.9e−1 (4.1e−3)$^+$	**8/ 0/ 0**
C-EHVI	3.6e+0 (1.4e−1)$^+$	3.3e−3 (3.8e−3)$^+$	3.9e+0 (4.6e−1)$^+$	2.5e−1 (1.8e−2)$^+$	-3.5e−1 (2.9e−1)$^+$	7.6e−2 (8.0e−2)$^+$	-3.3e−1 (1.0e−2)$^+$	-6.5e−1 (5.9e−2)$^+$	**8/ 0/ 0**
JES	3.4e+0 (1.3e−1)$^+$	1.1e−1 (8.0e−2)$^+$	4.5e+0 (1.7e−1)$^+$	2.6e−1 (2.0e−2)$^+$	-1.0e+0 (4.5e−1)$^+$	8.8e−2 (9.3e−2)$^+$	-3.3e−1 (8.3e−3)$^+$	-6.9e−1 (6.7e−3)$^+$	**8/ 0/ 0**
PSL	3.4e+0 (1.7e−1)$^+$	1.1e−2 (3.3e−2)$^+$	3.0e+0 (4.6e−1)$^\sim$	2.6e−1 (1.7e−2)$^+$	-6.3e−1 (2.1e−1)$^+$	1.1e−1 (9.4e−2)$^+$	-3.2e−1 (1.1e−2)$^+$	-7.0e−1 (2.7e−3)$^\sim$	**6/ 2/ 0**
SPMO	**3.1e+0 (3.0e−1)**	**9.0e−4 (8.3e−4)**	**2.9e+0 (4.9e−1)**	**2.1e−1 (5.0e−5)**	**-2.1e+0 (7.7e−3)**	**1.7e−4 (1.2e−4)**	**-3.4e−1 (2.3e−6)**	**-7.0e−1 (2.6e−3)**	

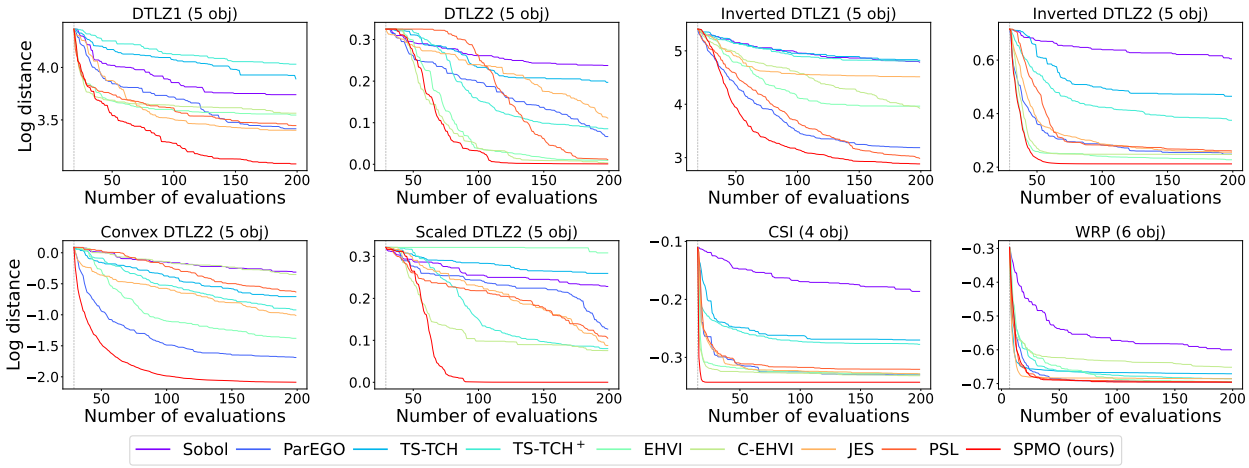


Figure 2: Trajectories of the distance-based metric (log distance) obtained by the nine methods on the benchmark problems with 5 objectives and the two real-world problems. Each coloured line represents the mean metric value on 30 independent runs (after the initial samples, represented by the dashed grey line).

For simplicity, we here only show the results of the benchmark problems with five objectives. Results on 3- and 10-objective problems can be found in Appendix F.1. A general pattern is that as the number of objectives increases, the advantages of SPMO become more pronounced. On the 3-objective problems, the differences between SPMO and the peer methods in terms of the two single-point metrics are relatively small, whereas on the 10-objective problems, the gaps in both metrics become substantially larger (see Figures 7 and 8 in the Appendix). Regarding the HV of all evaluated solutions, SPMO statistically outperforms the peer methods on at least 2 and 5 out of the 6 problems on the 3-objective and 10-objective cases, respectively.

Noisy Cases. We compare the proposed SPMO with the peer methods on the seven noisy benchmark and real-world problems. The results (the distance-based metric, two HV metrics, and convergence trajectories) are given in Appendix F.2. Like in the noiseless setting, SPMO significantly outperforms the peer methods on all the problems with respect to the single-point metrics (distance and HV). Regarding the HV of all the (nondominated) solutions obtained, SPMO achieves the best performance on the majority of the problems. Figure 5 shows the spider chart of the best solution (with respect to its HV) obtained by each algorithm in a typical run on the 9-objective car cab design problem. The violin plot of their HV values in 30 independent runs is given in the left panel for reference. As can be seen, the solution of SPMO is the best or close to the best on most of the objectives (except the first objective), thus having a clearly larger area.

Batch Setting. Previously, we considered the case in the sequential setting, where solutions are evaluated sequentially. Now we want to see if the proposed method works in the batch setting. Here, the batch size q is set to 5, a commonly used value (Lin et al., 2022; Jiang & Li, 2025b). As can be seen in Tables 14–16 and Figures 17–19 (Appendix F.3), similar to the results in the sequential setting, SPMO generally performs best.

Table 2: The HV of the best solution (in terms of its HV value) obtained by the SPMO and the peer methods on the benchmark problems with 5 objectives and the two real-world problems on 30 runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	WRP Mean (Std)	Sum up +/∼/−
Sobol	8.7e+12 (3.7e+11)$^+$	3.5e−2 (1.5e−2)$^+$	5.1e+12 (1.1e+12)$^+$	8.7e−4 (1.5e−3)$^+$	3.8e−1 (1.7e−1)$^+$	3.8e−2 (2.1e−2)$^+$	2.6e−1 (1.8e−2)$^+$	5.4e−1 (2.7e−2)$^+$	**8/ 0/ 0**
ParEGO	9.1e+12 (1.8e+11)$^+$	1.3e−1 (5.6e−2)$^+$	8.8e+12 (7.0e+11)$^+$	4.0e−2 (2.9e−3)$^+$	1.1e+0 (6.2e−2)$^+$	8.7e−2 (5.3e−2)$^+$	3.6e−1 (2.6e−3)$^+$	6.0e−1 (1.1e−3)$^\sim$	**7/ 1/ 0**
TS-TCH	8.4e+12 (3.8e+11)$^+$	3.2e−2 (1.3e−2)$^+$	4.9e+12 (1.2e+12)$^+$	7.1e−3 (1.1e−3)$^+$	6.5e−1 (1.4e−1)$^+$	2.8e−2 (1.5e−2)$^+$	3.1e−1 (9.3e−3)$^+$	5.9e−1 (1.7e−2)$^+$	**8/ 0/ 0**
TS-TCH$^+$	8.4e+12 (4.9e+11)$^+$	1.1e−1 (1.7e−2)$^+$	5.0e+12 (1.1e+12)$^+$	1.8e−2 (4.2e−3)$^+$	7.9e−1 (1.8e−1)$^+$	1.1e−1 (1.9e−2)$^+$	3.1e−1 (1.3e−2)$^+$	6.0e−1 (4.3e−3)$^+$	**8/ 0/ 0**
EHVI	9.0e+12 (9.5e+10)$^+$	**1.9e−1 (7.2e−3)**$^\sim$	7.5e+12 (9.7e+11)$^+$	4.5e−2 (1.4e−3)$^+$	1.0e+0 (9.7e−2)$^+$	1.5e−2 (1.4e−2)$^+$	3.6e−1 (6.8e−3)$^+$	6.0e−1 (2.1e−3)$^+$	**7/ 1/ 0**
C-EHVI	9.0e+12 (1.0e+11)$^+$	1.6e−1 (1.9e−2)$^+$	7.7e+12 (7.8e+11)$^+$	4.1e−2 (4.1e−3)$^+$	4.3e−1 (2.1e−1)$^+$	1.2e−1 (5.1e−2)$^+$	3.6e−1 (4.4e−3)$^\sim$	5.8e−1 (3.4e−2)$^+$	**7/ 1/ 0**
JES	9.0e+12 (1.2e+11)$^+$	1.0e−1 (3.9e−2)$^+$	6.2e+12 (5.0e+11)$^+$	3.9e−2 (4.5e−3)$^+$	8.5e−1 (2.4e−1)$^+$	1.1e−1 (5.5e−2)$^+$	3.6e−1 (5.1e−3)$^+$	6.0e−1 (3.6e−3)$^+$	**8/ 0/ 0**
PSL	9.2e+12 (1.9e+11)$^\sim$	1.9e−1 (3.0e−2)$^\sim$	9.1e+12 (4.3e+11)$^\sim$	3.8e−2 (3.9e−3)$^+$	6.3e−1 (1.4e−1)$^+$	9.7e−2 (5.3e−2)$^+$	3.6e−1 (5.5e−3)$^+$	6.0e−1 (1.4e−3)$^\sim$	**4/ 4/ 0**
SPMO	**9.3e+12 (2.8e+11)**	1.9e−1 (1.4e−2)	**9.2e+12 (4.8e+11)**	**4.9e−2 (1.2e−5)**	**1.3e+0 (5.0e−3)**	**1.6e−1 (1.6e−2)**	**3.6e−1 (3.2e−3)**	**6.0e−1 (1.7e−3)**	

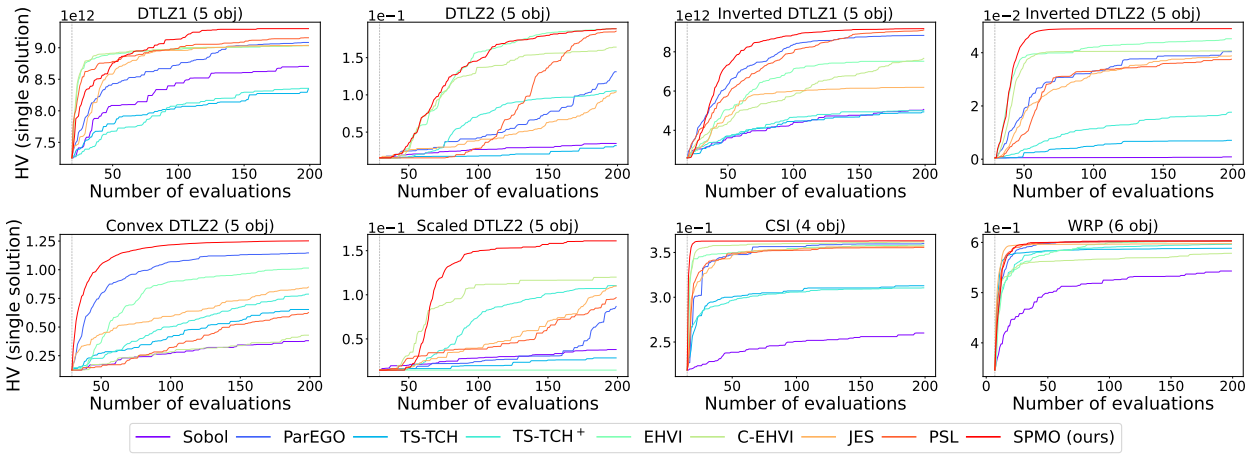


Figure 3: Trajectories of the HV of the best solution (in terms of its HV value) obtained by the SPMO and the peer methods on the benchmark problems with 5 objectives and the two real-world problems. Each coloured line represents the mean metric value on 30 independent runs.

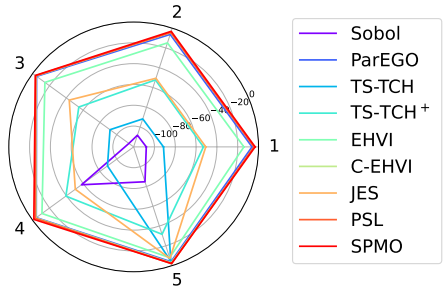


Figure 4: Spider chart of the best solution (in terms of its HV) obtained by the nine methods on the inverted DTLZ1 problem with 5 objectives in a typical run. Each axis in the spider chart represents one objective. Here, the objective values are multiplied by −1 in this minimisation problem, such that a solution with a larger area indicates better quality.

On the single-point metrics, it achieves the smallest distance on all the problems and the highest HV on 5 out of the 6 problems (except on DTLZ2). As for the HV of all evaluated solutions, SPMO obtains the best HV on two problems and takes the second or third places on the remaining ones. This indicates that prioritising convergence is also very useful for many-objective problems in batch settings.

Sensitivity Analysis. A parameter needed in the proposed method is the utopian point. In our experiment, we set it to be the vector consisting of the best value on each objective (i.e., the problem's ideal point). However, in real life, the ideal point is usually unknown before the optimisation. Hence, we would like to investigate how much different utopian points affect the performance. In this context, we consider three other settings based on the problem's true ideal point. The first one is slightly better than the ideal point, i.e., with a difference of 0.01, the second is fairly better than the ideal point (i.e. 0.1), and the last one is significantly better than the ideal point (i.e. 1.0). The results are given in Appendix F.4.1. As can be seen, interestingly, SPMO with the three lower utopian points with different levels performs better than or at least equivalently to SPMO with our current setting. This indicates that 1) using the ideal point in the proposed method

Table 3: The HV of all the solutions obtained by the SPMO and the peer methods on the benchmark problems with five objectives and the two real-world problems on 30 independent runs. The method with the best mean HV is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	WRP Mean (Std)	Sum up +/∼/−
Sobol	1.0e+13 (3.1e+10)$^+$	7.9e−2 (2.5e−2)$^+$	5.6e+12 (8.7e+11)$^+$	9.3e−4 (1.5e−3)$^+$	5.8e−1 (2.1e−1)$^+$	8.6e−2 (3.1e−2)$^+$	5.2e−1 (1.0e−2)$^+$	1.0e+0 (9.3e−3)$^\sim$	**7/ 1/ 0**
ParEGO	1.0e+13 (3.8e+10)$^-$	3.6e−1 (2.1e−1)$^+$	9.1e+12 (6.2e+11)$^+$	1.4e−1 (1.0e−2)$^-$	**1.5e+0 (2.8e−2)**$^-$	2.2e−1 (1.6e−1)$^+$	7.4e−1 (1.4e−2)$^-$	1.2e+0 (1.1e−2)$^-$	**3/ 0/ 5**
TS-TCH	1.0e+13 (2.6e+10)$^+$	5.9e−2 (2.7e−2)$^+$	5.4e+12 (1.0e+12)$^+$	2.0e−2 (3.6e−3)$^+$	1.0e+0 (1.5e−1)$^+$	5.1e−2 (2.7e−2)$^+$	6.5e−1 (1.1e−2)$^-$	8.1e−1 (3.3e−2)$^+$	**7/ 0/ 1**
TS-TCH$^+$	9.8e+12 (2.0e+11)$^+$	2.6e−1 (3.3e−2)$^+$	5.4e+12 (9.7e+11)$^+$	4.3e−2 (7.0e−3)$^+$	1.1e+0 (2.3e−1)$^+$	2.7e−1 (3.0e−2)$^\sim$	4.4e−1 (1.4e−2)$^+$	1.1e+0 (7.9e−3)$^-$	**6/ 1/ 1**
EHVI	**1.0e+13 (3.3e+8)**$^-$	**8.3e−1 (6.2e−2)**$^-$	8.2e+12 (6.6e+11)$^+$	**2.1e−1 (1.9e−3)**$^-$	1.5e+0 (6.9e−2)$^+$	2.0e−2 (1.7e−2)$^+$	7.4e−1 (9.6e−3)$^-$	**1.2e+0 (9.5e−3)**$^-$	**3/ 0/ 5**
C-EHVI	1.0e+13 (1.6e+11)$^\sim$	3.6e−1 (8.5e−2)$^+$	8.0e+12 (6.9e+11)$^+$	7.2e−2 (9.2e−3)$^-$	5.8e−1 (2.7e−1)$^+$	2.1e−1 (1.1e−1)$^+$	5.9e−1 (3.0e−2)$^-$	8.2e−1 (6.7e−2)$^+$	**5/ 1/ 2**
JES	1.0e+13 (9.0e+9)$^-$	2.8e−1 (1.5e−1)$^+$	8.6e+12 (3.5e+11)$^+$	1.4e−1 (9.4e−3)$^-$	1.3e+0 (2.6e−1)$^+$	**3.1e−1 (2.0e−1)**$^\sim$	7.4e−1 (1.3e−2)$^-$	1.2e+0 (1.1e−2)$^-$	**3/ 1/ 4**
PSL	1.0e+13 (1.1e+11)$^+$	5.7e−1 (1.4e−1)$^-$	9.5e+12 (3.3e+11)$^\sim$	1.6e−1 (9.3e−3)$^-$	1.0e+0 (1.7e−1)$^+$	2.4e−1 (2.0e−1)$^+$	**7.5e−1 (2.3e−2)**$^-$	1.2e+0 (9.2e−3)$^-$	**3/ 1/ 4**
SPMO	1.0e+13 (8.1e+10)	5.0e−1 (8.2e−2)	**9.6e+12 (2.9e+11)**	6.4e−2 (2.8e−3)	1.5e+0 (3.2e−2)	3.0e−1 (1.1e−1)	5.3e−1 (3.1e−2)	9.8e−1 (1.3e−1)	

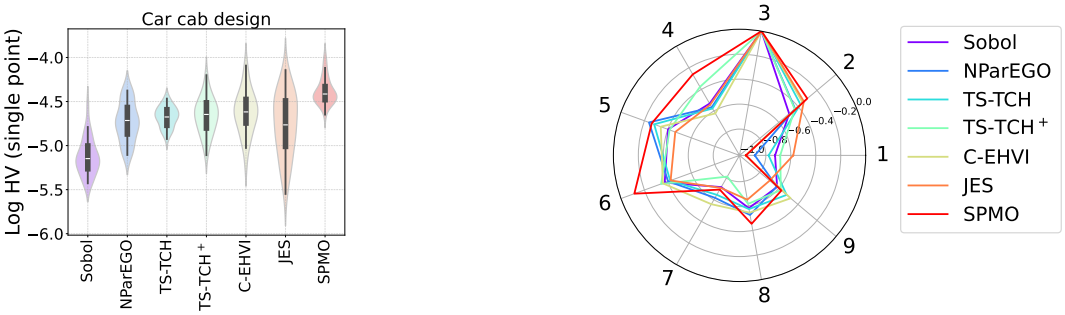


Figure 5: *Left:* Violin plots of the HV values of the best solution (with respect to its HV) obtained by all the methods on the car cab design problem in 30 independent runs. *Right:* The objective values (normalised and multiplied by −1) of the best solution (with the highest HV) obtained by each method on the cab design problem in a typical run.

may not be the best choice (though it performs better than the other MOBO methods), and 2) SPMO's performance is robust to the choice of the utopian point and a liberal estimate is sufficient to achieve good results, e.g., considering zero cost in real-world cases.

Previously, we investigated SPMO using different utopian points based on the problem's true ideal point. However, in real-world applications, one may have very little knowledge about the optimisation problem. To further examine a more practical setting, we consider dynamically setting the utopian point based only on the observed solutions during optimisation. We consider four settings, where the observed objective values are first normalised and the utopian point is then set to $(-0.1, \ldots, -0.1)$, $(-0.2, \ldots, -0.2)$, $(-0.5, \ldots, -0.5)$, and $(-1.0, \ldots, -1.0)$, respectively. The results are reported in Appendix F.4.2. Overall, the dynamic settings achieve comparable performance to the setting using the problem's true ideal point. Moreover, using a more optimistic dynamic utopian point, e.g., $(-1.0, \ldots, -1.0)$, can even lead to better performance.

Comparison of Single-Point Metrics within SPMO. In the proposed SPMO framework, we employ a distance metric (i.e., the distance of a solution to the utopian point). It is worth noting that different metrics can be adopted provided that they can reflect the quality of a solution in achieving a good trade-off between objectives. We now consider three other well-known metrics, weighted sum, Tchebycheff scalarisation, and single-point HV, denoted by SPMO$_{ws}$, SPMO$_{Tch}$, and SPMO$_{hv}$, respectively. The equal weights $(\frac{1}{m}, \ldots, \frac{1}{m})$ are used in SPMO$_{ws}$ and SPMO$_{Tch}$. The reference point of SPMO$_hv$ is set to 1.1 times the observed Pareto front, since using a fixed reference point (e.g., $(1.1, \ldots, 1.1)$ for DTLZ2) can lead to zero acquisition values when no candidate solution dominates the reference point. We compare these four variants of SPMO. The results (given in Appendix F.5) show that SPMO (using the distance metric, denoted as SPMO$_{dist}$) performs in general better than SPMO$_{ws}$, SPMO$_{Tch}$, and SPMO$_{hv}$. It achieves the best statistical results on all the problems in terms of the HV of the best solution. It is interesting to see that SPMO$_hv$, which optimises the single-point hypervolume, performs relatively poorly. One possible reason for this is that the reference point constructed during the search may not be reliable when the optimisation focuses on only one solution rather

than the whole solution set. As for the HV of all the solutions, SPMO$_{dist}$ performs best on inverted DTLZ1 and convex DTLZ2, but performs relatively poorly on inverted DTLZ2 and CSI. A possible explanation is that other variants (e.g., SPMO$_{hv}$) have slower convergence and can be better in exploring different solutions, and thus perform better on relatively easy-to-converge problems.

Incorporating Preference Information into SPMO. Previously, we focused on the preference-free setting, where no prior knowledge of the decision maker's preferences is available. However, SPMO can naturally incorporate preference information into its single-point metrics. We extend the four metrics introduced above, i.e., SPMO$_{dist}$, SPMO$_{ws}$, SPMO$_{Tch}$, and SPMO$_{hv}$, to the preference-based setting. Given a weight vector $\boldsymbol{w}$, a preference-based distance metric can be defined as $\|\boldsymbol{w} \odot (\boldsymbol{y} - \boldsymbol{z})\|_2$, where $\odot$ denotes element-wise multiplication. Weighted-sum and Tchebycheff scalarisation can incorporate preference information directly through a weight vector. Regarding the single-point HV metric, given a reference point $\boldsymbol{r}$, preference-based single-point HV can be defined as $\prod_{i=1}^{m} \max(w_i(y_i - r_i), 0)$.

We consider a very different weight vector from the previous equal weights; that is $\boldsymbol{w} = (0.1, 0.2, \ldots, 0.1m)$ for an m-objective problem. Note that these weights will be normalised by their sum, i.e., $\boldsymbol{w} \leftarrow \boldsymbol{w}/\sum_{i=1}^{m} w_i$, to ensure that their sum is equal to 1.0. Similar to the previous comparison, we use the preference-based distance metric and preference-based single-point HV as the performance metrics. The results are given in Appendix F.6. As shown in the results, SPMO$_{dist}$ generally performs better than the other variants. It obtains the best results on 6 out of the 7 problems in terms of the preference-based distance metric. Regarding preference-based single-point HV, SPMO$_{dist}$ obtains the best results on 4 out of the 7 problems. Overall, the distance-based metric remains a robust choice.

Acquisition Optimisation Wall Time. Lastly, we present the wall time for optimising the acquisition function (i.e., determining a solution to be evaluated). The results (Appendix F.7) show that our method is among the fastest algorithms. When the number of objectives is 3 or 5, the time of all the methods is acceptable with a maximum of 98 seconds. As the number of objectives increases to 10, hypervolume-based methods become very expensive. For example, EHVI and NEHVI take about half an hour and more than 3 hours, respectively.[2] Note that although PSL takes only several minutes based on the initial samples, the hypervolume computation can take up to one day for a single evaluation when the number of non-dominated solutions becomes large in the 10-objective case. The proposed SPMO method shows high computational efficiency, achieving the lowest time requirement in four out of the six instances.

7 Discussion

7.1 Beyond Approximating the Entire Pareto Front in Many-Objective Optimisation

In general, multi-objective optimisation algorithms aim to generate a set of solutions that well approximates the entire Pareto front. However, the required number of solutions to well approximate the entire Pareto front could be exponentially large with respect to the number of objectives. This difficulty is particularly acute when the evaluation budget is very limited. Hence this work proposes the SPMO framework to find a single high-quality trade-off solution without approximating the entire Pareto front. Along the same lines, some methods in literature target a specific region of the front, such as the central region (Gaudrie et al., 2018; 2020; Binois et al., 2020), though they do not aim for a single trade-off solution.

Another related direction is to identify a small set of representative solutions that collectively cover many objectives (Ding et al., 2024; Liu et al., 2024; Li et al., 2025b; Lin et al., 2025; Liu et al., 2025; Maus et al., 2025). Note that most of these methods are developed for general many-objective optimisation rather than Bayesian optimisation, but they share the similar motivation of avoiding full Pareto-front approximation. These representative solutions complement one another so that each objective is well represented by at least one solution in the set, e.g., using sum-of-minimum (Ding et al., 2024), Tchebycheff set scalarisation (Lin et al., 2025), and optimal transport between objectives and solutions (Li et al., 2025b). Liu et al. (2025) refines and extends this idea as a conditional set-optimization problem, i.e., the few-for-many problem, and Maus et al.

[2]Wall time is measured based on the initial samples. Due to the exponentially increasing computational complexity with objectives, EHVI, NEHVI, and PSL are evaluated only for problems with objectives $m \leq 6$.

(2025) further extend this idea to the Bayesian optimisation setting. Together with our work, these studies suggest a trend in many-objective optimisation: moving beyond full Pareto-front approximation towards identifying a single trade-off solution or a small set of representative solutions. Note that this differs from the archiving process in evolutionary multi-objective optimisation, where an archive is maintained to preserve a small representative set of Pareto optimal solutions for presentation to the decision-maker (though the algorithm may generate many Pareto optimal solutions) (Li et al., 2024b). Instead, the objective here is to generate only a small number of solutions given the expensiveness of the optimisation problem.

7.2 The Distance-to-Utopian Metric

In the proposed SPMO framework, we employ a distance metric (i.e., the distance of a solution to the utopian point). One reason for using the distance-to-utopian metric is that it naturally favours a single well-balanced trade-off solution. Minimising this distance favours a solution that is jointly close to the best attainable value of each objective, rather than being extremely good in one objective but poor in others. Such a distance-based metric is also widely used as a utility function in multi-criteria decision making, e.g., TOPSIS (Hwang & Yoon, 1981) and compromise programming (Zeleny, 1973).

Note that we have considered three other well-known metrics in the experiments, i.e., weighted sum, Tchebycheff scalarisation, and single-point HV. The results show that SPMO with the distance-to-utopian metric generally performs better than these alternatives. This suggests the effectiveness of the distance-based method under the SPMO framework. In addition, it is worth noting that we also reported the single-point HV as an additional evaluation metric, alongside the distance metric, to validate the effectiveness of the considered distance-based SPMO. The results show a consistent trend across these two single-point metrics: methods that perform well under the distance-to-utopian metric also tend to perform well under single-point HV (see Tables 1 and 2).

8 Conclusion

This work presented a multi-objective BO framework that aims to find a single trade-off solution of the highest possible quality with respect to multiple objectives, rather than seeking to explore their entire Pareto front. We theoretically proved the convergence guarantees on acquisition optimisation under the SAA and empirically verified the proposed framework through extensive experiments, including on noiseless/noisy and sequential/batch cases, by sensitivity analysis, with different metrics for the acquisition function, and on a range of benchmark and real-world problems. A noticeable limitation of the proposed framework is that it focuses on finding a single trade-off point, thus failing to capture the information about the entire Pareto front; it thus may be less useful for certain applications where such information is valuable (e.g., the ranges and nadir point of the Pareto front). A detailed discussion of its applicability is provided in Appendix G. However, notably, the proposed framework showed its competitiveness against existing state-of-the-arts with respect to even the quality of the whole solution set (through HV of all solutions, see Table 3). Future work includes studying and enhancing the scalability of the proposed methods (i.e., in higher-dimensional search space) and extending their applicability to other scenarios, e.g., multi-fidelity optimisation (see Appendix H for more details).

References

Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Multi-objective Bayesian optimisation with preferences over objectives. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Alaleh Ahmadianshalchi, Syrine Belakaria, and Janardhan Rao Doppa. Pareto front-diverse batch multi-objective Bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10): 10784–10794, 2024.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20577–20612. Curran Associates, Inc., 2023.

Kirill Antonov, Elena Raponi, Hao Wang, and Carola Doerr. High dimensional Bayesian optimization with kernel principal component analysis. In *International Conference on Parallel Problem Solving from Nature*, pp. 118–131. Springer, 2022.

Raul Astudillo and Peter Frazier. Multi-attribute Bayesian optimization with interactive preference learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 4496–4507. PMLR, 2020.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538. Curran Associates, Inc., 2020.

Dianne Carrol Bautista. *A sequential design for approximating the Pareto front using the expected Pareto improvement function*. The Ohio State University, 2009.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. *Proceedings of the AAAI Conference on artificial intelligence*, 34(06):10035–10043, 2020a.

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective Bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10044–10052, 2020b.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Output space entropy search framework for multi-objective Bayesian optimization. *Journal of artificial intelligence research*, 72:667–715, 2021.

Shriya Bhatija, Paul-David Zuercher, Jakob Thumm, and Thomas Bohné. Multi-objective causal bayesian optimization. *arXiv preprint arXiv:2502.14755*, 2025.

Mickael Binois and Nathan Wycoff. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

Mickael Binois, Victor Picheny, Patrick Taillandier, and Abderrahmane Habbal. The kalai-smorodinsky solution for many-objective Bayesian optimization. *Journal of Machine Learning Research*, 21(150):1–42, 2020.

Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński (eds.). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252. Springer, 2008.

Ali Cheaitou and Pierre Cariou. Greening of maritime transportation: a multi-objective optimization approach. *Annals of Operations Research*, 273(1):501–525, 2019.

Taicai Chen, Yue Duan, Dong Li, Lei Qi, Yinghuan Shi, and Yang Gao. PG-LBO: enhancing high-dimensional Bayesian optimization with pseudo-label and Gaussian process guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11381–11389, 2024.

Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3:67–81, 2017.

Tinkle Chugh. Scalarizing functions in Bayesian multiobjective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60(3):575–594, 2014.

Abhranil Das. New methods to compute the generalized chi-square distribution. *Journal of Statistical Computation and Simulation*, pp. 1–35, 2025.

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

Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Hypervolume knowledge gradient: A lookahead approach for multi-objective Bayesian optimization with partial information. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7167–7204. PMLR, 2023.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9851–9864. Curran Associates, Inc., 2020.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective Bayesian optimization under input noise. In *International Conference on Machine Learning*, pp. 4831–4866. PMLR, 2022a.

Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective Bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pp. 507–517. PMLR, 2022b.

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

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications*, pp. 105–145. Springer, 2005.

Kalyanmoy Deb, Shubham Gupta, David Daum, Jürgen Branke, Abhishek Kumar Mall, and Dhanesh Padmanabhan. Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 13(5):1054–1074, 2009.

Ian Delbridge, David Bindel, and Andrew Gordon Wilson. Randomly projected additive Gaussian processes for regression. In *International Conference on Machine Learning*, pp. 2453–2463. PMLR, 2020.

Jingda Deng, Jianyong Sun, Qingfu Zhang, and Hui Li. Expected hypervolume improvement is a particular hypervolume improvement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):16217–16225, 2025.

Lisang Ding, Ziang Chen, Xinshang Wang, and Wotao Yin. Efficient algorithms for sum-of-minimum optimization. *arXiv preprint arXiv:2402.07070*, 2024.

Youssef Diouane, Victor Picheny, Rodolophe Le Riche, and Alexandre Scotto Di Perrotolo. Trego: a trust-region framework for efficient global optimization. *Journal of Global Optimization*, 86(1):1–23, 2023.

John H Dunlap, Jeffrey G Ethier, Amelia A Putnam-Neeb, Sanjay Iyer, Shao-Xiong Lennon Luo, Haosheng Feng, Jose Antonio Garrido Torres, Abigail G Doyle, Timothy M Swager, Richard A Vaia, et al. Continuous flow synthesis of pyridinium salts accelerated by multi-objective Bayesian optimization with active learning. *Chemical Science*, 14(30):8061–8069, 2023.

Michael TM Emmerich and André H Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17:585–609, 2018.

Michael TM Emmerich, Kyriakos C Giannakoglou, and Boris Naujoks. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pp. 493–503. PMLR, 2021.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5496–5507. Curran Associates, Inc., 2019.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

Eduardo C Garrido-Merchán, Daniel Fernández-Sánchez, and Daniel Hernández-Lobato. Parallel predictive entropy search for multi-objective Bayesian optimization with constraints applied to the tuning of machine learning algorithms. *Expert Systems with Applications*, 215:119328, 2023.

Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Predictive entropy search for multi-objective Bayesian optimization with constraints. *Neurocomputing*, 361:50–68, 2019.

David Gaudrie, Rodolphe Le Riche, Victor Picheny, Benoit Enaux, and Vincent Herbert. Budgeted multi-objective optimization with a focus on the central part of the Pareto front–extended version. *arXiv preprint arXiv:1809.10482*, 2018.

David Gaudrie, Rodolphe Le Riche, Victor Picheny, Benoit Enaux, and Vincent Herbert. Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence*, 88(1):187–212, 2020.

Miguel González-Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Sø ren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140478–140508. Curran Associates, Inc., 2024.

Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional Bayesian optimization via tree-structured additive models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7630–7638, 2021.

Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective Bayesian optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1492–1501, New York, USA, 2016. PMLR.

Robert M Hierons, Miqing Li, Xiaohui Liu, Jose Antonio Parejo, Sergio Segura, and Xin Yao. Many-objective test suite generation for software product lines. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 29(1):1–46, 2020.

Vu Viet Hoang, Hung The Tran, Sunil Gupta, and Vu Nguyen. High dimensional Bayesian optimization using lasso variable selection. *arXiv preprint arXiv:2504.01743*, 2025.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, pp. 65–70, 1979.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 20793–20817. PMLR, 21–27 Jul 2024.

Ching-Lai Hwang and Kwangsun Yoon. *Multiple attribute decision making and applications*. Springer, 1981.

Jean-Pierre Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3/4): 419–426, 1961.

Joshua Hang Sai Ip, Ankush Chakrabarty, Ali Mesbah, and Diego Romeres. User preference meets Pareto-optimality in multi-objective Bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20246–20254, 2025.

Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pp. 2419–2426. IEEE, 2008.

Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation*, 26(3):411–440, 2018.

Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2013a.

Himanshu Jain and Kalyanmoy Deb. An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 307–321. Springer, 2013b.

Shinkyu Jeong and Shigeru Obayashi. Efficient global optimization (ego) for multi-objective problem and data mining. In *2005 IEEE congress on evolutionary computation*, volume 3, pp. 2138–2145. IEEE, 2005.

Chao Jiang and Miqing Li. Multi-objectivising acquisition functions in Bayesian optimisation. *ACM Transactions on Evolutionary Learning and Optimization*, 5(2), 2025a.

Chao Jiang and Miqing Li. Trading off quality and uncertainty through multi-objective optimisation in batch Bayesian optimisation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):27027–27035, 2025b.

Chao Jiang, Zimin Liang, and Miqing Li. When to use which? An investigation of search methods on expensive black-box optimisation problems, 2026. URL https://openreview.net/forum?id=vsKYK3KpUR.

Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian optimisation with continuous approximations. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1799–1808. PMLR, 2017.

Andy J Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-guided multi-objective Bayesian optimization with batch evaluations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17708–17720. Curran Associates, Inc., 2020.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1546–1558. Curran Associates, Inc., 2020.

Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):1–35, 2015a.

Bingdong Li, Zixiang Di, Yongfan Lu, Hong Qian, Feng Wang, Peng Yang, Ke Tang, and Aimin Zhou. Expensive multi-objective Bayesian optimization based on diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):27063–27071, 2025a.

Diantong Li, Fengxue Zhang, Chong Liu, and Yuxin Chen. Constrained multi-objective Bayesian optimization through optimistic constraints estimation. *arXiv preprint arXiv:2411.03641*, 2024a.

Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE transactions on Evolutionary Computation*, 19 (5):694–716, 2014a.

Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–38, 2019.

Miqing Li, Shengxiang Yang, and Xiaohui Liu. Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365, 2014b.

Miqing Li, Shengxiang Yang, and Xiaohui Liu. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228:45–65, 2015b.

Miqing Li, Tao Chen, and Xin Yao. How to evaluate solutions in pareto-based search-based software engineering: A critical review and methodological guidance. *IEEE Transactions on Software Engineering*, 48(5):1771–1799, 2022.

Miqing Li, Manuel López-Ibáñez, and Xin Yao. Multi-objective archiving. *IEEE Transactions on evolutionary computation*, 28(3):696–717, 2024b.

Shibo Li, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-fidelity Bayesian optimization via deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8521–8531. Curran Associates, Inc., 2020.

Ziyue Li, Tian Li, Virginia Smith, Jeff Bilmes, and Tianyi Zhou. Many-objective multi-solution transport. In *International Conference on Learning Representations*, volume 2025, pp. 56276–56301, 2025b.

Jing Liang, Yahang Lou, Mingyuan Yu, Ying Bi, and Kunjie Yu. A survey of surrogate-assisted evolutionary algorithms for expensive optimization. *Journal of Membrane Computing*, pp. 1–20, 2024.

Qiaohao Liang, Aldair E Gongora, Zekun Ren, Armi Tiihonen, Zhe Liu, Shijing Sun, James R Deneault, Daniil Bash, Flore Mekki-Berrada, Saif A Khan, et al. Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, 2021.

Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 19231–19247. Curran Associates, Inc., 2022.

Xi Lin, Yilu Liu, Xiaoyuan Zhang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Few for many: Tchebycheff set scalarization for many-objective optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yilu Liu, Chengyu Lu, Xi Lin, and Qingfu Zhang. Many-objective cover problem: Discovering few solutions to cover many objectives. In *International Conference on Parallel Problem Solving from Nature*, pp. 68–82. Springer, 2024.

Yilu Liu, Xi Lin, Liang Zhao, and Qingfu Zhang. Few for many: Towards efficient and flexible many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2025.

Andre KY Low, Flore Mekki-Berrada, Abhishek Gupta, Aleksandr Ostudin, Jiaxun Xie, Eleonore Vissol-Gaudin, Yee-Fun Lim, Qianxiao Li, Yew Soon Ong, Saif A Khan, et al. Evolution-guided Bayesian optimization for constrained multi-objective optimization in self-driving labs. *npj Computational Materials*, 10(1):104, 2024.

Gustavo Malkomes, Bolong Cheng, Eric H Lee, and Mike Mccourt. Beyond the Pareto efficient frontier: Constraint active search for multiobjective experimental design. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 7423–7434. PMLR, 2021.

Evgenii S Matrosov, Ivana Huskova, Joseph R Kasprzyk, Julien J Harou, Chris Lambert, and Patrick M Reed. Many-objective optimization and visual analytics reveal key trade-offs for london's water supply. *Journal of Hydrology*, 531:1040–1053, 2015.

Natalie Maus, Kyurae Kim, Yimeng Zeng, Haydn Thomas Jones, Fangping Wan, M Torres, C de la Fuente-Nunez, and J Gardner. Multi-objective coverage Bayesian optimization (MOCOBO). *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 1999.

Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based Bayesian optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021.

Nobuo Namura, Koji Shimoyama, and Shigeru Obayashi. Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(6):898–913, 2017.

Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pp. 4752–4761. PMLR, 2019.

Lam Ngo, Huong Ha, Jeffrey Chan, and Hongyu Zhang. Mobo-osd: Batch multi-objective Bayesian optimization via orthogonal search directions. *arXiv preprint arXiv:2510.20872*, 2025.

Ryota Ozaki, Kazuki Ishikawa, Youhei Kanzaki, Shion Takeno, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-objective Bayesian optimization with active preference learning. *Proceedings of the AAAI conference on artificial intelligence*, 38(13):14490–14498, 2024.

Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: Reliable high-dimensional Bayesian optimization for combinatorial and mixed spaces. In *Advances in Neural Information Processing Systems*, volume 36, pp. 1764–1793. Curran Associates, Inc., 2023.

Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional Bayesian optimization. *arXiv preprint arXiv:2502.09198*, 2025.

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 766–776. PMLR, 2020.

Ji Won Park, Nataša Tagasovska, Michael Maser, Stephen Ra, and Kyunghyun Cho. Botied: Multi-objective bayesian optimization with tied multivariate ranks. *arXiv preprint arXiv:2306.00344*, 2023.

Seongeon Park, Jonggeol Na, Minjun Kim, and Jong Min Lee. Multi-objective Bayesian optimization of chemical reactor design using computational fluid dynamics. *Computers & Chemical Engineering*, 119: 25–37, 2018.

James Parr. *Improvement criteria for constraint handling and multiobjective optimization.* PhD thesis, University of Southampton, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Peng Peng, Yi Peng, Fuguo Liu, Shuai Long, Cheng Zhang, Aitao Tang, Jia She, Jianyue Zhang, and Fusheng Pan. Bayesian optimization and explainable machine learning for high-dimensional multi-objective optimization of biodegradable magnesium alloys. *Journal of Materials Science & Technology*, 2025.

Victor Picheny, Mickael Binois, and Abderrahmane Habbal. A Bayesian optimization approach to find nash equilibria. *Journal of Global Optimization*, 73:171–192, 2019.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *Parallel Problem Solving from Nature – PPSN X*, pp. 784–794. Springer, 2008.

Jixiang Qing, Ivo Couckuyt, and Tom Dhaene. A robust multi-objective Bayesian optimization framework considering input uncertainty. *Journal of Global Optimization*, 86(3):693–711, 2023.

Elena Raponi, Hao Wang, Mariusz Bujny, Simonetta Boria, and Carola Doerr. High dimensional Bayesian optimization assisted by principal component analysis. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I 16*, pp. 169–183. Springer, 2020.

Tapabrata Ray, Kang Tai, and Kin Chye Seow. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424, 2001.

Ashwin Renganathan and Kade Carlson. *q*`POTS`: Efficient batch multiobjective Bayesian optimization via pareto optimal Thompson sampling. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4051–4059. PMLR, 03–05 May 2025.

Harold Ruben. Probability content of regions under spherical normal distributions, iv: The distribution of homogeneous and non-homogeneous quadratic functions of normal variables. *The Annals of Mathematical Statistics*, 33(2):542–570, 1962.

Maria Laura Santoni, Elena Raponi, Renato De Leone, and Carola Doerr. Comparison of high-dimensional Bayesian optimization algorithms on BBOB. *ACM Transactions on Evolutionary Learning and Optimisation*, 4(3):1–33, 2024.

Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20, 2020.

Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

Ilya M Sobol. The distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity Bayesian optimization with Gaussian processes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3158–3167. PMLR, 2019.

Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective Bayesian optimization using Pareto-frontier entropy. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9279–9288. PMLR, 2020.

Joshua Svenson. *Computer experiments: Multiobjective optimization and sensitivity analysis*. PhD thesis, The Ohio State University, 2011.

Joshua Svenson and Thomas Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94:250–264, 2016.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9334–9345. PMLR, 2020.

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

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9922–9938. Curran Associates, Inc., 2022.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 745–754. PMLR, 2018.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, pp. 196–202. Springer, 1992.

Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 788–798. PMLR, 22–25 Jul 2020.

Zhitong Xu, Haitao Wang, Jeff M Phillips, and Shandian Zhe. Standard Gaussian process is all you need for high-dimensional Bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1): 3–34, 2019a.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945–956, 2019b.

Byeng D Youn, KK Choi, R-J Yang, and Lei Gu. Reliability-based design optimization for crashworthiness of vehicle side impact. *Structural and Multidisciplinary Optimization*, 26:272–283, 2004.

Milan Zeleny. Compromise programming. *Multiple Criteria Decision Making*, 1973.

Dawei Zhan, Yuansheng Cheng, and Jun Liu. Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(6):956–975, 2017.

Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11096–11105. PMLR, 2020.

Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-based multi-fidelity Bayesian optimization. In *NIPS workshop on Bayesian optimization*, volume 49. Journal of Machine Learning Research JMLR. org Cambridge, MA, 2017.

Liang Zhao and Qingfu Zhang. Exact formulas for the computation of expected tchebycheff improvement. In *2023 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2023.

Ruihao Zheng and Zhenkun Wang. Boundary decomposition for nadir objective vector estimation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 14349–14385. Curran Associates, Inc., 2024.

Juliusz Krzysztof Ziomek and Haitham Bou Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In *International Conference on Machine Learning*, pp. 43347–43368. PMLR, 2023.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *International conference on machine learning*, pp. 462–470. PMLR, 2013.

Marcela Zuluaga, Andreas Krause, and Markus Püschel. ϵ-pal: An active learning approach to the multi-objective optimization problem. *Journal of Machine Learning Research*, 17(104):1–32, 2016. URL http://jmlr.org/papers/v17/15-047.html.

Appendix to:

Do We Really Need to Approach the Entire Pareto Front in Many-Objective Bayesian Optimisation?

A Multi-objective Bayesian Optimisation (MOBO)

MOBO consists of two main steps, i.e., training m Gaussian process models based on the observed solutions and optimising an acquisition function $\alpha(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ to select a solution for evaluation. In this work, we model each objective with an independent Gaussian process $f_i \sim \mathcal{GP}(m_i(\boldsymbol{x}), k_i(\boldsymbol{x}, \boldsymbol{x}'))$, where $m_i(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is the ith mean function, and $k_i(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the ith covariance function. We use the notation $K(\mathrm{A}, \mathrm{B})$ to represent the covariance matrix at all pairs of solutions in set A and in set B. Given n observed solutions $\mathcal{D} = \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1}^{n}$ where $\boldsymbol{y}^{(t)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}) + \boldsymbol{\zeta}^{(t)}$ and the noise $\boldsymbol{\zeta}^{(t)} \sim \mathcal{N}(0, \mathrm{diag}(\boldsymbol{\sigma}_{\zeta}^2))$, the posterior distribution of ith objective at a new location $\boldsymbol{x}$ is a Gaussian distribution:

$$p(f_i(\boldsymbol{x})|\mathcal{D}) \sim \mathcal{N}(\mu_i(\boldsymbol{x}), \sigma_i^2(\boldsymbol{x})) \tag{6}$$

$$\mu_i(\boldsymbol{x}) = K(\boldsymbol{x}, X)(K(X, X) + \sigma_{\zeta_i}^2 \mathbf{I})^{-1} Y_i \tag{7}$$

$$\sigma_i^2(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}) - K(\boldsymbol{x}, X)((K(X, X) + \sigma_{\zeta_i}^2 \mathbf{I})^{-1} K(X, \boldsymbol{x}) \tag{8}$$

where $\mu_i(\boldsymbol{x})$ and $\sigma_i^2(\boldsymbol{x})$ are the mean and variance at $\boldsymbol{x}$, respectively; $X = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}) \in \mathbb{R}^{n \times d}$ and $Y_i = (y_i^{(1)}, \ldots, y_i^{(n)}) \in \mathbb{R}^n$ are the matrix of evaluated solutions and the corresponding vector of y values, respectively; $\sigma_{\zeta_i}^2$ is the variance of the observation noise $\zeta_i \sim \mathcal{N}(0, \sigma_{\zeta_i}^2)$, and corresponds to the i-th diagonal entry of the noise covariance matrix $\mathrm{diag}(\boldsymbol{\sigma}_{\zeta}^2)$;

B Illustrative Example of ESPI

To help understand the proposed ESPI, an illustration of ESPI in a bi-objective case is shown in Figure 6. In this example, the utopian point is denoted by the black star, while the red dots represent the nondominated solutions from the current dataset. A new candidate point, whose objective values are yet to be observed, is shown as the blue dot. The red dotted line indicates the shortest Euclidean distance g^* from the utopian point to the existing nondominated set. In contrast, the blue dotted line represents the distance from the new point to the utopian point. When a new point lies within the shaded blue region and is closer to the utopian point, the improvement $I_{SP}(\cdot)$ is higher.

C Extended Related Work

Over the last two decades, a variety of BO methods have been proposed to tackle expensive multi-objective optimisation problems (Jeong & Obayashi, 2005; Bautista, 2009; Svenson, 2011; Svenson & Santner, 2016; Zuluaga et al., 2016; Zhan et al., 2017; Picheny et al., 2019; Belakaria et al., 2020b; Malkomes et al., 2021; Park et al., 2023; Ngo et al., 2025). Most of them aim to identify a good approximation of the entire Pareto front (Keane, 2006; Svenson, 2011; Parr, 2013; Zuluaga et al., 2013; Ahmadianshalchi et al., 2024). To do so, some studies convert a multi-objective problem into multiple single-objective problems by a scalarisation function (e.g., random augmented Tchebycheff scalarisation). They then optimise acquisition functions from single-objective BO to determine the next evaluation point(s) (Knowles, 2006; Zhao & Zhang, 2023). In these studies, different acquisition functions are employed, such as expected improvement (EI) (Jones et al., 1998) in Knowles (2006); Zhang et al. (2010); Namura et al. (2017); Chugh (2020), Thompson

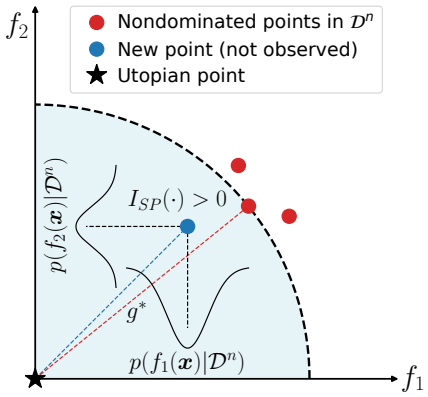


Figure 6: Illustration of the proposed expected single-point improvement (ESPI) in a bi-objective space. The figure shows the utopian point (black star), nondominated points in the current dataset $\mathcal{D}^n$ (red dots), and a new candidate point whose true objective values are not yet observed (blue dot). The dashed arc represents the current best Euclidean distance from the utopian point to the nondominated set, denoted as g^* (red dotted line). The blue dotted line represents the distance from the new candidate point to the utopian point $\boldsymbol{z}^*$, which may improve upon the current best distance g^*. When a new point lies within the shaded blue region and is closer to the utopian point, the improvement $I_{SP}(\cdot)$ is higher.

sampling (TS) (Thompson, 1933) in Paria et al. (2020); Zhang & Golovin (2020), and upper confidence bound (UCB) (Lai & Robbins, 1985) in Paria et al. (2020); Zhang & Golovin (2020); Li et al. (2024a).

The remaining studies directly optimise multi-objective problems by considering the definition of optimality in multi-objective optimisation, i.e., the Pareto dominance relation. A representative approach is to use HV since maximising the HV value is equivalent to finding the entire Pareto front (Ponweiser et al., 2008; Couckuyt et al., 2014; Daulton et al., 2020; 2021; Renganathan & Carlson, 2025; Bhatija et al., 2025). Along this line, expected hypervolume improvement (EHVI) is widely considered (Emmerich et al., 2006; Daulton et al., 2023; Qing et al., 2023; Yang et al., 2019b;a; Deng et al., 2025) as it is a natural extension of the EI for multi-objective optimisation. Another idea is to leverage information theory to guide exploration toward regions likely contributing to the Pareto front. Such methods focus on improving the posterior of optimal inputs (i.e., the approximated Pareto set) (Garrido-Merchán et al., 2023; Garrido-Merchán & Hernández-Lobato, 2019; Hernandez-Lobato et al., 2016), optimal outputs (i.e., the approximated Pareto front) (Belakaria et al., 2019; 2021; Suzuki et al., 2020), or both of them (Tu et al., 2022).

That said, there do exist a few studies that do not aim to identify the entire Pareto front. Among them, some attempt to use decision-maker preferences to guide the search towards specific region(s) (Abdolshah et al., 2019; Astudillo & Frazier, 2020; Ozaki et al., 2024; Ip et al., 2025). Such methods typically adjust the target region by eliciting or updating decision-maker preferences during optimisation. Another attempt is to directly target certain region(s) of the Pareto front, without an assumption that decision-maker preferences can be available or elicited (Gaudrie et al., 2018; 2020; Binois et al., 2020). For example, Gaudrie et al. (2018; 2020) propose the Centred Expected Hypervolume Improvement (C-EHVI) by dynamically adjusting the reference point using the Kalai-Smorodinsky equilibrium (also used in Binois et al. (2020)) to approach the central part of the Pareto front, regardless of whether the central region contains best trade-off solutions, even if the central region is of relatively low quality (e.g., on a rather concave front). In contrast, our proposed approach searches for a good trade-off solution, which may not lie in the intermediate region, but can often be interpreted as a "knee" point on the Pareto front. C-EHVI is more relevant to our study, and we have thus included it in our experimental comparison.

D Theoretical Results

D.1 Proof of Theorem 4.1.

We consider the setting from Balandat et al. (2020, Section D.5). Let $\epsilon^{(t)} \sim \mathcal{N}(0, I_m)$.[3] Using the reparameterisation trick, we can write the posterior at $\boldsymbol{x}$ as

$$\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) = \bar{\mu}(\boldsymbol{x}) + \bar{L}(\boldsymbol{x})\epsilon^{(t)}$$

[3]Theorem 4.1 can be extended to handle non-iid base samples from a family of quasi-Monte Carlo methods as in Balandat et al. (2020).

where $\bar{\mu}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^m$ is the multi-output GP's posterior mean; $\bar{L}(\boldsymbol{x}) \in \mathbb{R}^{m\times m}$ is a root decomposition (often a Cholesky decomposition) of the multi-output GP's posterior covariance $\bar{K} \in \mathbb{R}^{m\times m}$; and $\epsilon^{(t)} \in \mathbb{R}^m$. Let

$$\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) = \max\Big(0,\, g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big)$$

where $g^* = \min_{\boldsymbol{x}\in\bar{X}} g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{z}^*)$ and $\boldsymbol{z}^* = (z_1^*, z_2^*, \ldots, z_m^*)$ is the utopian point. Following Balandat et al. (2020, Theorem 3), we need to show that there exists an integrable function $\ell : \mathbb{R}^m \mapsto \mathbb{R}$ such that for almost every $\epsilon^{(t)}$ and all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \subset \mathbb{R}^d$,

$$|\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) - \bar{A}(\boldsymbol{y}, \epsilon^{(t)})| \le \ell(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{9}$$

Let

$$\begin{aligned}\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) &= \max\Big(0,\, g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big)\\ &= \frac{1}{2}\Big(g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\| + \big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big).\end{aligned}$$

Hence we obtain,

$$\begin{aligned}&\big|\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) - \bar{A}(\boldsymbol{y}, \epsilon^{(t)})\big|\\ =&\Big|\frac{1}{2}\Big(\|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big) + \frac{1}{2}\Big(\big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big)\Big|.\end{aligned}$$

Let $I_1 := \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|$ and $I_2 := \big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|$. Hence

$$\big|\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) - \bar{A}(\boldsymbol{y}, \epsilon^{(t)})\big| \le \frac{1}{2}|I_1| + \frac{1}{2}|I_2|.$$

We observe that

$$\begin{aligned}|I_1| &= \big|\|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\\ &\le \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)})\|\\ &= \|\bar{\mu}(\boldsymbol{y}) + \bar{L}(\boldsymbol{y})\epsilon^{(t)} - (\bar{\mu}(\boldsymbol{x}) + \bar{L}(\boldsymbol{x})\epsilon^{(t)})\|\\ &\le \|\bar{\mu}(\boldsymbol{y}) - \bar{\mu}(\boldsymbol{x})\| + \|(\bar{L}(\boldsymbol{y}) - \bar{L}(\boldsymbol{x}))\epsilon^{(t)}\|.\end{aligned}$$

Since $\mathcal{X}$ is compact, and $\bar{\mu}$ and $\bar{L}$ have uniformly bounded gradients, they are Lipschitz. There exist $C_{\mu_1}, C_{L_1} < \infty$ such that

$$\begin{aligned}|I_1| &\le \|\bar{\mu}(\boldsymbol{y}) - \bar{\mu}(\boldsymbol{x})\| + \|(\bar{L}(\boldsymbol{y}) - \bar{L}(\boldsymbol{x}))\epsilon^{(t)}\|\\ &\le \ell_{I_1}(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\|\end{aligned}$$

where $\ell_{I_1}(\epsilon^{(t)}) := C_{\mu_1} + C_{L_1}\|\epsilon^{(t)}\|$. Furthermore,

$$\begin{aligned}|I_2| &= \Big|\big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|g^* - \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big|\\ &\le \big|\|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\\ &\le \|\bar{\boldsymbol{f}}_t(\boldsymbol{x}, \epsilon^{(t)}) - \bar{\boldsymbol{f}}_t(\boldsymbol{y}, \epsilon^{(t)})\|\\ &= \|\bar{\mu}(\boldsymbol{x}) + \bar{L}(\boldsymbol{x})\epsilon^{(t)} - (\bar{\mu}(\boldsymbol{y}) + \bar{L}(\boldsymbol{y})\epsilon^{(t)})\|\\ &\le \|\bar{\mu}(\boldsymbol{x}) - \bar{\mu}(\boldsymbol{y})\| + \|(\bar{L}(\boldsymbol{x}) - \bar{L}(\boldsymbol{y}))\epsilon^{(t)}\|.\end{aligned}$$

Since $\mathcal{X}$ is compact, and $\bar{\mu}$ and $\bar{L}$ have uniformly bounded gradients, they are Lipschitz. There exist $C_{\mu_2}, C_{L_2} < \infty$ such that

$$|I_2| \leq \ell_{I_2}(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\|$$

where $\ell_{I_2}(\epsilon^{(t)}) := C_{\mu_2} + C_{L_2}\|\epsilon^{(t)}\|$. Hence

$$\begin{aligned}\left|\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) - \bar{A}(\boldsymbol{y}, \epsilon^{(t)})\right| &\leq \frac{1}{2}|I_1| + \frac{1}{2}|I_2| \\ &\leq \frac{1}{2}\ell_{I_1}(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\| + \frac{1}{2}\ell_{I_2}(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\| \\ &= \frac{1}{2}(\ell_{I_1}(\epsilon^{(t)}) + \ell_{I_2}(\epsilon^{(t)}))\|\boldsymbol{x} - \boldsymbol{y}\|.\end{aligned}$$

Hence

$$\left|\bar{A}(\boldsymbol{x}, \epsilon^{(t)}) - \bar{A}(\boldsymbol{y}, \epsilon^{(t)})\right| \leq \ell(\epsilon^{(t)})\|\boldsymbol{x} - \boldsymbol{y}\|$$

where $\ell(\epsilon^{(t)}) := (C_{\mu_1} + C_{\mu_2}) + (C_{L_1} + C_{L_2})\|\epsilon^{(t)}\|$. Note that $\ell(\epsilon^{(t)})$ is integrable because all absolute moments exist for the Gaussian distribution. Since this satisfies the criteria for Theorem 3 in Balandat et al. (2020), the theorem holds for ESPI.

D.2 Theorem D.1 and Its Proof

Theorem D.1. *Suppose that $\mathcal{X}$ is compact and that $\boldsymbol{f}$ has a multi-output GP prior with continuously differentiable mean and covariance functions. Let $X = \{\boldsymbol{x}^{(t)}\}_{t=1}^{n}$ be the set of observed decision vectors in $\mathcal{D}$, $\alpha^*_{NESPI} := \max_{\boldsymbol{x}\in\mathcal{X}} \alpha_{NESPI}(\boldsymbol{x})$ denote the maximum of NESPI, $S^* := \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_{NESPI}(\boldsymbol{x})$ denote the set of maximisers of α_{NESPI}, $\hat{\alpha}^N_{NESPI}(\boldsymbol{x})$ denote the deterministic acquisition function via the base samples $\{\epsilon^{(t)}\}_{t=1}^{N} \sim \mathcal{N}(0, I_{(n+1)m})$. Suppose that $\hat{\boldsymbol{x}}^*_N \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \hat{\alpha}^N_{NESPI}(\boldsymbol{x})$, then*

(1) $\hat{\alpha}^N_{NESPI}(\hat{\boldsymbol{x}}^*_N) \to \alpha^*_{NESPI}$ a.s.,

(2) $d(\hat{\boldsymbol{x}}^*_N, S^*) \to 0$ a.s., where $d(\hat{\boldsymbol{x}}^*_N, S^*) := \inf_{\boldsymbol{x}\in S^*}\|\hat{\boldsymbol{x}}^*_N - \boldsymbol{x}\|$.

Proof of Theorem D.1. Let $X := [(\boldsymbol{x}^{(1)})^T, \ldots, (\boldsymbol{x}^{(n)})^T]^T \in \mathbb{R}^{nd}$ be the n observed points; $\boldsymbol{x}^{(n+1)} \in \mathcal{X} \subset \mathbb{R}^d$ be a candidate point; and $\epsilon^{(t)} \in \mathbb{R}^{(n+1)m}$ with $\epsilon^{(t)} \sim \mathcal{N}(0, I_{(n+1)m})$. Let $\tilde{\boldsymbol{f}}_t(X, \boldsymbol{x}^{(n+1)}) := [\tilde{\boldsymbol{f}}_t(\boldsymbol{x}^{(1)}), \ldots, \tilde{\boldsymbol{f}}_t(\boldsymbol{x}^{(n)}), \tilde{\boldsymbol{f}}_t(\boldsymbol{x}^{(n+1)})]$ denote the tth sample of the corresponding objectives, so that we can write the posterior via the reparameterisation trick as:

$$\boldsymbol{f}_t(X, \boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) = \mu(X, \boldsymbol{x}^{(n+1)}) + L(X, \boldsymbol{x}^{(n+1)})\epsilon^{(t)}$$

where $\mu(X, \boldsymbol{x}^{(n+1)}) : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)m}$ is the multi-output GP's posterior mean; $L(X, \boldsymbol{x}^{(n+1)}) \in \mathbb{R}^{(n+1)m\times(n+1)m}$ is a root decomposition of the multi-output GP's posterior covariance $K \in \mathbb{R}^{(n+1)m\times(n+1)m}$. Let $\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(i)}, \epsilon^{(t)}) := S_i\Big(\mu(X, \boldsymbol{x}^{(n+1)}) + L(X, \boldsymbol{x}^{(n+1)})\epsilon^{(t)}\Big)$, where $\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(i)}, \epsilon^{(t)}) \in \mathbb{R}^m$ represents the posterior at point $\boldsymbol{x}^{(i)}$ and $S_i \in \mathbb{R}^{m\times(n+1)m}, i = 1, \ldots, n+1$ is the selector matrix used to extract the corresponding element for the $\boldsymbol{x}^{(i)}$. Let

$$A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) = \max\Big(0,\ \hat{g^*_t}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big).$$

Let $\hat{g^*_t}(X) := \min_{\boldsymbol{x}_{obs}\in\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}} \|\boldsymbol{f}^{(m)}_t(\boldsymbol{x}_{obs}, \epsilon^{(t)}) - \boldsymbol{z}^*\|$, hence we obtain:

$$A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) = \max\Big(0, \min_{\boldsymbol{x}_{obs}\in\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}} \|\boldsymbol{f}^{(m)}(\boldsymbol{x}_{obs}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big).$$

Following Balandat et al. (2020, Theorem 3), we need to show that there exists an integrable function $\ell : \mathbb{R}^{(n+1)m} \mapsto \mathbb{R}$ such that for almost every $\epsilon^{(t)}$ and all $\boldsymbol{x}^{(n+1)}, \boldsymbol{y}^{(n+1)} \in \mathcal{X} \subset \mathbb{R}^d$,

$$|A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) - A(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}; X)| \leq \ell(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\|. \tag{10}$$

Let

$$\begin{aligned} A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) &= \max\Big(0, \hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big) \\ &= \frac{1}{2}\Big(\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\| + \big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big). \end{aligned}$$

Hence we obtain

$$\begin{aligned} &\big|A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) - A(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}; X)\big| \\ =& \Big|\frac{1}{2}\Big(\|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\Big) \\ &+ \frac{1}{2}\Big(\big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big)\Big|. \end{aligned}$$

Let $I_1' := \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|$ and $I_2' := \big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|$. Hence

$$\big|A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) - A(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}; X)\big| \leq \frac{1}{2}|I_1'| + \frac{1}{2}|I_2'|.$$

We observe that

$$\begin{aligned} |I_1'| &= \big|\|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| \\ &\leq \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)})\| \\ &= \|\mu^{(m)}(\boldsymbol{y}^{(n+1)}) + L^{(m)}(\boldsymbol{y}^{(n+1)})\epsilon^{(t)} - (\mu^{(m)}(\boldsymbol{x}^{(n+1)}) + L^{(m)}(\boldsymbol{x}^{(n+1)})\epsilon^{(t)})\| \\ &\leq \|\mu^{(m)}(\boldsymbol{y}^{(n+1)}) - \mu^{(m)}(\boldsymbol{x}^{(n+1)})\| + \|(L^{(m)}(\boldsymbol{y}^{(n+1)}) - L^{(m)}(\boldsymbol{x}^{(n+1)}))\epsilon^{(t)}\| \end{aligned}$$

Since $\mathcal{X}$ is compact, and $\mu^{(m)}$ and $L^{(m)}$ have uniformly bounded gradients, they are Lipschitz. There exist $C'_{\mu_1}, C'_{L_1} < \infty$ such that

$$\begin{aligned} |I_1'| &\leq \|\mu^{(m)}(\boldsymbol{y}^{(n+1)}) - \mu^{(m)}(\boldsymbol{x}^{(n+1)})\| + \|(L^{(m)}(\boldsymbol{y}^{(n+1)}) - L^{(m)}(\boldsymbol{x}^{(n+1)}))\epsilon^{(t)}\| \\ &\leq \ell_{I_1'}(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\|. \end{aligned}$$

where $\ell_{I_1'}(\epsilon^{(t)}) := C'_{\mu_1} + C'_{L_1}\|\epsilon^{(t)}\|$. Furthermore,

$$\begin{aligned} |I_2'| &= \Big|\big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| - \big|\hat{g_t^*}(X) - \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big|\Big| \\ &\leq \big|\|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\| - \|\boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{z}^*\|\big| \\ &\leq \|\boldsymbol{f}^{(m)}(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - \boldsymbol{f}^{(m)}(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)})\| \\ &= \|\mu^{(m)}(\boldsymbol{x}^{(n+1)}) + L^{(m)}(\boldsymbol{x}^{(n+1)})\epsilon^{(t)} - (\mu^{(m)}(\boldsymbol{y}^{(n+1)}) + L^{(m)}(\boldsymbol{y}^{(n+1)})\epsilon^{(t)})\| \\ &\leq \|\mu^{(m)}(\boldsymbol{x}^{(n+1)}) - \mu^{(m)}(\boldsymbol{y}^{(n+1)})\| + \|(L^{(m)}(\boldsymbol{x}^{(n+1)}) - L^{(m)}(\boldsymbol{y}^{(n+1)}))\epsilon^{(t)}\| \end{aligned}$$

Since $\mathcal{X}$ is compact, and $\mu^{(m)}$ and $L^{(m)}$ have uniformly bounded gradients, they are Lipschitz. There exist $C'_{\mu_2}, C'_{L_2} < \infty$ such that

$$|I_2'| \leq \ell_{I_2'}(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\|$$

where $\ell_{I_2'}(\epsilon^{(t)}) := C'_{\mu_2} + C'_{L_2}\|\epsilon^{(t)}\|$. Hence

$$\begin{aligned}\left|A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}; X) - A(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)}; X)\right| &\leq \frac{1}{2}|I_1'| + \frac{1}{2}|I_2'| \\ &\leq \frac{1}{2}\ell_{I_1'}(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\| + \frac{1}{2}\ell_{I_2'}(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\| \\ &= \frac{1}{2}(\ell_{I_1'}(\epsilon^{(t)}) + \ell_{I_2'}(\epsilon^{(t)}))\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\|\end{aligned}$$

Hence

$$\left|A(\boldsymbol{x}^{(n+1)}, \epsilon^{(t)}) - A(\boldsymbol{y}^{(n+1)}, \epsilon^{(t)})\right| \leq \ell(\epsilon^{(t)})\|\boldsymbol{x}^{(n+1)} - \boldsymbol{y}^{(n+1)}\|$$

where $\ell(\epsilon^{(t)}) := (C'_{\mu_1} + C'_{\mu_2}) + (C'_{L_1} + C'_{L_2})\|\epsilon^{(t)}\|$. Note that $\ell(\epsilon^{(t)})$ is integrable because all absolute moments exist for the Gaussian distribution. Since this satisfies the criteria for Theorem 3 in Balandat et al. (2020), the theorem holds for NESPI. It is worth noting that Theorems 4.1 and D.1 readily extend to the batch setting.

It can be shown that the gradient of $\hat{\alpha}_{\text{ESPI}}(\boldsymbol{x})$ or $\hat{\alpha}_{\text{NESPI}}(\boldsymbol{x})$ is an unbiased estimator of the true gradient of α_{ESPI} or α_{NESPI}, though it is not necessary for the SAA approach (Daulton et al., 2021).

E Experiment Settings

E.1 Implementation details

All experiments were conducted using Python 3.12, with all methods developed on the open-source Python framework BoTorch (Balandat et al., 2020), which builds on GPyTorch (Gardner et al., 2018) for Gaussian process modelling and PyTorch (Paszke et al., 2019) for automatic differentiation. The computational studies were performed on a Red Hat Enterprise Linux 8.8 system, operating on a 64-bit x86 CPU architecture. The computing cluster utilised Intel Xeon Platinum 8360Y processors running at 2.40 GHz.

The code for ParEGO, NParEGO, TS-TCH, EHVI, NEHVI, and JES is available at `https://github.com/pytorch/botorch`. Specifically, we implement ParEGO and NParEGO based on the implementation provided by BoTorch.[4] We implement TS-TCH based on the implementation provided by BoTorch.[5] We implement EHVI and NEHVI based on the implementation provided by BoTorch.[6] We implement JES based on the implementation provided by BoTorch.[7]

The data and code are available at `https://github.com/ChaoJiang52/SPMO`.

E.2 Method Details

For all the methods, we set the same $2(d+1)$ points from a scrambled Sobol sequence and allow a maximum of 200 evaluations, following the practice in Daulton et al. (2020; 2021); Konakovic Lukovic et al. (2020). All the methods use $N = 128$ Monte Carlo samples.

For ParEGO (Knowles, 2006) and its noisy variant NParEGO (Daulton et al., 2021), we employ random scalarisations, whereby a weight vector $\boldsymbol{w} \in \mathbb{R}^m$ is generated from the unit simplex. The augmented Tchebycheff scalarisation function is applied, defined as $g(\boldsymbol{y}) = \max_i(w_i y_i) + \alpha \sum_i (w_i y_i)$. Log expected improvement and noisy log expected improvement are used as the acquisition functions in ParEGO and NParEGO, respectively, as recommended in Ament et al. (2023). In the batch setting, q distinct weight vectors are sampled, and the acquisition function is optimised sequentially for each.

For TS-TCH (Paria et al., 2020), similarly to ParEGO, we employ random scalarisations by using the augmented Tchebycheff scalarisation function. After converting to single-objective optimisation problem,

[4] EI, NEI, and the BoTorch multi-objective tutorial.
[5] TS and BoTorch multi-objective tutorial
[6] EHVI, NEHVI, and BoTorch multi-objective tutorial
[7] Botorch JES implementation

Table 4: Reference points of the six benchmark problems used for hypervolume computation in EHVI and NEHVI, as well as for performance evaluation. Note that the referents points of the two real-world problems, i.e., car side impact design and car cab design, are set to $(1.1, ..., 1.1) \in \mathbb{R}^m$ in the normalised objective space following the practice in Tanabe & Ishibuchi (2020) the utopian and nadir points are available at `https://github.com/ryojitanabe/reproblems/tree/master/ideal_nadir_points`).

Problem	Reference point	Suggested by
DTLZ1	$(400.0, ..., 400.0) \in \mathbb{R}^m$	Balandat et al. (2020); Chugh (2020)
DTLZ2	$(1.1, \dots, 1.1) \in \mathbb{R}^m$	Balandat et al. (2020); Daulton et al. (2020); Ishibuchi et al. (2018)
Inverted DTLZ1	$(400.0, \dots, 400.0) \in \mathbb{R}^m$	Chugh (2020); Ishibuchi et al. (2018)
Inverted DTLZ2	$(1.1, \dots, 1.1) \in \mathbb{R}^m$	Ishibuchi et al. (2018)
Convex DTLZ2	$(1.1, \dots, 1.1) \in \mathbb{R}^m$	Ishibuchi et al. (2018)
Scaled DTLZ2	$(1.1 * 2^0, \dots, 1.1 * 2^m) \in \mathbb{R}^m$	Ishibuchi et al. (2018)
DTLZ3	$(10000.0, \dots, 10000.0) \in \mathbb{R}^m$	Balandat et al. (2020)
DTLZ4	$(1.1, \dots, 1.1) \in \mathbb{R}^m$	Balandat et al. (2020)
DTLZ5	$(10.0, \dots, 10.0) \in \mathbb{R}^m$	Balandat et al. (2020)
DTLZ6	$(10.0, \dots, 10.0) \in \mathbb{R}^m$	Balandat et al. (2020)
DTLZ7	$(15.0, \dots, 15.0) \in \mathbb{R}^m$	Balandat et al. (2020)

Thompson sampling is used as the acquisition function. We draw a sample from the joint posterior over a discrete set of $1000d$ points sampled from a scrambled Sobol sequence, suggested by Daulton et al. (2020). In the batch setting, q distinct weight vectors are sampled, and the acquisition function is optimised sequentially for each.

For EHVI (Daulton et al., 2020) and its noisy variant NEHVI (Daulton et al., 2021), the reference point is predefined, following the practice in Daulton et al. (2020); Yang et al. (2019b). Table 4 lists the reference points used for each problem in the experimental evaluation. The logarithmic variants of both acquisition functions are employed, as recommended in Ament et al. (2023). In the batch setting, the sequential greedy optimisation strategy is adopted. It is worth noting that in this study, EHVI and NEHVI are evaluated only on problems with 3 and 5 objectives, as the acquisition optimisation wall time becomes prohibitively high when the number of objectives increases to 10 (see Table 28).

For C-EHVI (Gaudrie et al., 2018; 2020), the Kalai-Smorodinsky equilibrium is used to determine the reference point of HV. The logarithmic variant of the acquisition functions is employed, as recommended in Ament et al. (2023).

For joint entropy search (JES), we use $S = 10$ Monte Carlo samples and $p = 10$ number of Pareto optimal points, according to Tu et al. (2022). In the batch setting, the sequential greedy optimisation strategy is adopted.

For all the problems, we normalise the input variables and standardise the objective values before training Gaussian processes. We assume an independent surrogate model for each objective, using a constant mean function and a Matérn 5/2 ARD kernel. We optimise all acquisition functions by using the L-BFGS-B, with up to 200 iterations.

E.3 Problem Details

The details of the benchmark problems and real-world problems are given in the following.

DTLZ1. DTLZ1 is a scalable benchmark problem from Deb et al. (2005), which is defined as:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= \frac{1}{2}x_1x_2\cdots x_{m-1}(1+g(\boldsymbol{x}_m)),\\
f_2(\boldsymbol{x}) &= \frac{1}{2}x_1x_2\cdots(1-x_{m-1})(1+g(\boldsymbol{x}_m)),\\
&\vdots\\
f_{m-1}(\boldsymbol{x}) &= \frac{1}{2}x_1(1-x_2)(1+g(\boldsymbol{x}_m)),\\
f_m(\boldsymbol{x}) &= \frac{1}{2}(1-x_1)(1+g(\boldsymbol{x}_m)),\\
&\text{s.t.}\quad 0\le x_i\le 1,\quad \text{for } i=1,2,\ldots,d.
\end{aligned}
$$

where $g(\boldsymbol{x}_m) = 100\left(|\boldsymbol{x}_m| + \sum_{x_i\in\boldsymbol{x}_m}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right)$ and $\boldsymbol{x}_m$ is the last $d-m+1$ variables.

DTLZ2. DTLZ2 is a scalable benchmark problem from Deb et al. (2005), which is defined as:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= (1+g(\boldsymbol{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2),\\
f_2(\boldsymbol{x}) &= (1+g(\boldsymbol{x}_m))\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2),\\
f_3(\boldsymbol{x}) &= (1+g(\boldsymbol{x}_m))\cos(x_1\pi/2)\cdots\sin(x_{m-2}\pi/2),\\
&\vdots\\
f_m(\boldsymbol{x}) &= (1+g(\boldsymbol{x}_m))\sin(x_1\pi/2),\\
&s.t.\quad 0\le x_i\le 1,\quad \text{for } i=1,2,\ldots,d.
\end{aligned}
$$

where $g(\boldsymbol{x}_m) = \sum_{x_i\in\boldsymbol{x}_m}(x_i-0.5)^2$ and $\boldsymbol{x}_m$ is the last $d-m+1$ variables.

Inverted DTLZ1. Inverted DTLZ1 is a variant of DTLZ1 (Jain & Deb, 2013a), which is defined as:

$$f_i(\boldsymbol{x}) = 0.5\cdot(1+g(\boldsymbol{x}_m)) - f_i^{\text{DTLZ1}}(\boldsymbol{x}),\quad i=1,\ldots,m$$

where $g(\boldsymbol{x}_m)$ is the same function as used in DTLZ1, $f_i^{\text{DTLZ1}}(\boldsymbol{x})$ denotes the ith objective of the original DTLZ1 formulation.

Inverted DTLZ2. Inverted DTLZ2 is a variant of DTLZ2 (Jain & Deb, 2013b), which is defined as:

$$f_i(\boldsymbol{x}) = 1+g(\boldsymbol{x}_m) - f_i^{\text{DTLZ2}}(\boldsymbol{x}),\quad i=1,\ldots,m$$

where $g(\boldsymbol{x}_m)$ is the same function as used in DTLZ2, $f_i^{\text{DTLZ2}}(\boldsymbol{x})$ denotes the ith objective of the original DTLZ2 formulation.

Convex DTLZ2. Convex DTLZ2 is a variant of DTLZ2 (Deb & Jain, 2013), which is defined as:

$$
\begin{aligned}
f_i(\boldsymbol{x}) &= (f_i^{\text{DTLZ2}}(\boldsymbol{x}))^4,\quad i=1,\ldots,m-1\\
f_m(\boldsymbol{x}) &= (f_m^{\text{DTLZ2}}(\boldsymbol{x}))^2
\end{aligned}
$$

where $f_i^{\text{DTLZ2}}(\boldsymbol{x})$ denotes the ith objective of the original DTLZ2 formulation. This problem convert the original concave problem to convex problem.

Scaled DTLZ2. Scaled DTLZ2 is a variant of DTLZ2, which is defined as:

$$f_i(\boldsymbol{x}) = 2^{i-1}\cdot f_i^{\text{DTLZ2}}(\boldsymbol{x}),\quad i=1,\ldots,m$$

where $f_i^{\text{DTLZ2}}(\boldsymbol{x})$ denotes the ith objective of the original DTLZ2 formulation. This benchmark problem is used to see whether an algorithm can deal with problems with different scales of different objectives.

DTLZ3–DTLZ7. DTLZ3–DTLZ7 are scalable multi-objective benchmark problems. Their mathematical formulations are provided in Deb et al. (2005).

According to the original paper (Deb et al., 2005), the dimensionality d of DTLZ1 and its variant (i.e., inverted DTLZ1) is $m+4$, the dimensionality d of DTLZ2-6 and their variants (i.e., inverted DTLZ2, convex DTLZ2, and scaled DTLZ2) is $m+9$, and the dimensionality d of DTLZ7 is $m+19$.

Car Side Impact Design (CSI). The car side-impact problem aims to minimise vehicle weight while satisfying safety constraints related to occupant injury and structural response (Jain & Deb, 2013a). It involves $m=4$ objectives with $d=7$ variables, which are based on a surrogate model that is fit to data collected from a simulator. The mathematical formulations are given as follows:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 10^{-5}x_6 + 2.73x_7 \\
f_2(\boldsymbol{x}) &= 4.72 - 0.5x_4 - 0.19x_2x_3 \\
f_3(\boldsymbol{x}) &= 0.5\left(V_{\mathrm{MBP}}(\boldsymbol{x}) + V_{\mathrm{FD}}(\boldsymbol{x})\right) \\
f_4(\boldsymbol{x}) &= -\sum_{i=1}^{10} \max\left(g_i(\boldsymbol{x}), 0\right)
\end{aligned}
$$

where the constraint functions $g_i(\boldsymbol{x})$ are defined as:

$$
\begin{aligned}
g_1(\boldsymbol{x}) &= 1 - 1.16 + 0.3717x_2x_4 + 0.0092928x_3 \\
g_2(\boldsymbol{x}) &= 0.32 - 0.261 + 0.0159x_1x_2 + 0.06486x_1 + 0.019x_2x_7 - 0.0144x_3x_5 - 0.0154464x_6 \\
g_3(\boldsymbol{x}) &= 0.32 - 0.214 - 0.00817x_5 + 0.045195x_1 + 0.0135168x_1 - 0.03099x_2x_6 \\
&\quad + 0.018x_2x_7 - 0.007176x_3 - 0.023223x_3 + 0.00364x_5x_6 + 0.018x_2^2 \\
g_4(\boldsymbol{x}) &= 0.32 - 0.74 + 0.61x_2 + 0.031296x_3 + 0.031872x_7 - 0.227x_2^2 \\
g_5(\boldsymbol{x}) &= 32 - 28.98 - 3.818x_3 + 4.2x_1x_2 - 1.27296x_6 + 2.68065x_7 \\
g_6(\boldsymbol{x}) &= 32 - 33.86 - 2.95x_3 + 5.057x_1x_2 + 3.795x_2 + 3.4431x_7 - 1.45728 \\
g_7(\boldsymbol{x}) &= 32 - 46.36 + 9.9x_2 + 4.4505x_1 \\
g_8(\boldsymbol{x}) &= 4 - f_2(\boldsymbol{x}) \\
g_9(\boldsymbol{x}) &= 9.9 - V_{\mathrm{MBP}}(\boldsymbol{x}) \\
g_{10}(\boldsymbol{x}) &= 15.7 - V_{\mathrm{FD}}(\boldsymbol{x})
\end{aligned}
$$

with volume terms defined as:

$$
\begin{aligned}
V_{\mathrm{MBP}}(\boldsymbol{x}) &= 10.58 - 0.674x_1x_2 - 0.67275x_2 \\
V_{\mathrm{FD}}(\boldsymbol{x}) &= 16.45 - 0.489x_3x_7 - 0.8435x_6x_7.
\end{aligned}
$$

The search space is defined as:

$$
\begin{aligned}
x_1 &\in [0.5, 1.5], \\
x_2 &\in [0.45, 1.35], \\
x_3, x_4 &\in [0.5, 1.5], \\
x_5 &\in [0.875, 2.625], \\
x_6, x_7 &\in [0.4, 1.2].
\end{aligned}
$$

Water Resource Planning (WRP). The water resource planning problem (Ray et al., 2001) aims to minimise the costs and expected losses associated with an urban storm drainage system. It involves $m=6$ objectives with $d=3$ variables, where the first five objectives correspond to the drainage network cost, storage facility cost, treatment facility cost, expected flood damage cost, and expected economic loss due to flooding,

respectively. The sixth objective is defined as the total constraint violation. The mathematical formulations are given as follows:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= 106780.37(x_2 + x_3) + 61704.67 \\
f_2(\boldsymbol{x}) &= 3000x_1 \\
f_3(\boldsymbol{x}) &= \frac{305700 \times 2289x_2}{(0.06 \times 2289)^{0.65}} \\
f_4(\boldsymbol{x}) &= 250 \times 2289 \exp(-39.75x_2 + 9.9x_3 + 2.74) \\
f_5(\boldsymbol{x}) &= 25 \left(\frac{1.39}{x_1 x_2} + 4940x_3 - 80 \right) \\
f_6(\boldsymbol{x}) &= \sum_{i=1}^{7} \max\left(g_i(\boldsymbol{x}), 0\right).
\end{aligned}
$$

where the constraint functions $g_i(\boldsymbol{x})$ are defined as:

$$
\begin{aligned}
g_1(\boldsymbol{x}) &= \frac{0.00139}{x_1 x_2} + 4.94x_3 - 0.08 - 1 \\
g_2(\boldsymbol{x}) &= \frac{0.000306}{x_1 x_2} + 1.082x_3 - 0.0986 - 1 \\
g_3(\boldsymbol{x}) &= \frac{12.307}{x_1 x_2} + 49408.24x_3 + 4051.02 - 50000 \\
g_4(\boldsymbol{x}) &= \frac{2.098}{x_1 x_2} + 8046.33x_3 - 696.71 - 16000 \\
g_5(\boldsymbol{x}) &= \frac{2.138}{x_1 x_2} + 7883.39x_3 - 705.04 - 10000 \\
g_6(\boldsymbol{x}) &= 0.417x_1 x_2 + 1721.26x_3 - 136.54 - 2000 \\
g_7(\boldsymbol{x}) &= \frac{0.164}{x_1 x_2} + 631.13x_3 - 54.48 - 550.
\end{aligned}
$$

The search space is defined as:

$$
\begin{aligned}
x_1 &\in [0.01, 0.45], \\
x_2, x_3 &\in [0.01, 0.10].
\end{aligned}
$$

Car Cab Design. This vehicle performance optimisation problem involves $m = 9$ objectives with $d = 7$ variables, relating to aspects such as car roominess, fuel economy, acceleration time, and road noise at various speeds (Deb & Jain, 2013). The problem includes 7 decision variables and 4 stochastic variables, which are based on a surrogate model that is fit to data collected from a simulator, defined as:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.75x_5 + 10^{-5}x_6 + 2.73x_7 \\
f_2(\boldsymbol{x}) &= [1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10}]_+ \\
f_3(\boldsymbol{x}) &= \left[\frac{1}{0.32}\begin{pmatrix} 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} \\ + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} \end{pmatrix}\right]_+ \\
f_4(\boldsymbol{x}) &= \left[\frac{1}{0.32}\begin{pmatrix} 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 \\ + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} \\ - 0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} - 0.018x_2^2 \end{pmatrix}\right]_+ \\
f_5(\boldsymbol{x}) &= \left[\frac{0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2}{0.32}\right]_+ \\
f_6(\boldsymbol{x}) &= \left[\frac{1}{32}\cdot\frac{1}{3}\begin{pmatrix} 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.77x_7x_8 + 0.32x_9x_{10} \\ + 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 \\ + 22x_8x_9 + 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} \end{pmatrix}\right]_+ \\
f_7(\boldsymbol{x}) &= \left[\frac{4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2}{4.0}\right]_+ \\
f_8(\boldsymbol{x}) &= \left[\frac{10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10}}{9.9}\right]_+ \\
f_9(\boldsymbol{x}) &= \left[\frac{16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2}{15.7}\right]_+
\end{aligned}
$$

where $[\cdot]_+$ denotes $\max(0, \cdot)$ and the search space is defined as:

$$
\begin{aligned}
x_1 &\in [0.5,\ 1.5], \\
x_2 &\in [0.45,\ 1.35], \\
x_3,\ x_4 &\in [0.5,\ 1.5], \\
x_5 &\in [0.875,\ 2.625], \\
x_6,\ x_7 &\in [0.4,\ 1.2].
\end{aligned}
$$

The four stochastic variables are defined as:

$$
\begin{aligned}
x_8 &\sim \mathcal{N}(0.345,\ 0.006^2), \\
x_9 &\sim \mathcal{N}(0.192,\ 0.006^2), \\
x_{10},\ x_{11} &\sim \mathcal{N}(0,\ 10^2).
\end{aligned}
$$

F Additional Experimental Results

F.1 Noiseless Cases

In this section, we present the results on the noiseless problems, including DTLZ1 and DTLZ2, their four variants with 3 and 10 objectives, and the two real-world problems. Tables 5, 6 and 7 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value) and the HV of all evaluated solutions obtained by the SPMO and the peer methods on the six noiseless problems, respectively. Figures 7, 8 and 9 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 5, 6, and 7, respectively. In addition, Figure 10 presents the trajectories of the distance metric obtained by each method on the noiseless problems with 3 and 10 objectives. Figure 11 presents the trajectories of the HV of the best solution obtained by each method on the noiseless problems with 3 and 10 objectives. Figure 12 presents the trajectories of the HV of all evaluated solutions obtained by each method on the noiseless problems. Note that the results of the HV of all evaluated solutions are reported every 10 evaluations due to the high computational cost of HV calculation. We also give the results of the problems DTLZ3–DLTZ7 with 5 objectives, shown in Tables 8, 9 and 10.

Table 5: Results of the distance-based metric (log distance) obtained by the SPMO and the peer methods on the noiseless problems with 3 objectives (**top**) and 10 objectives (**bottom**) on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

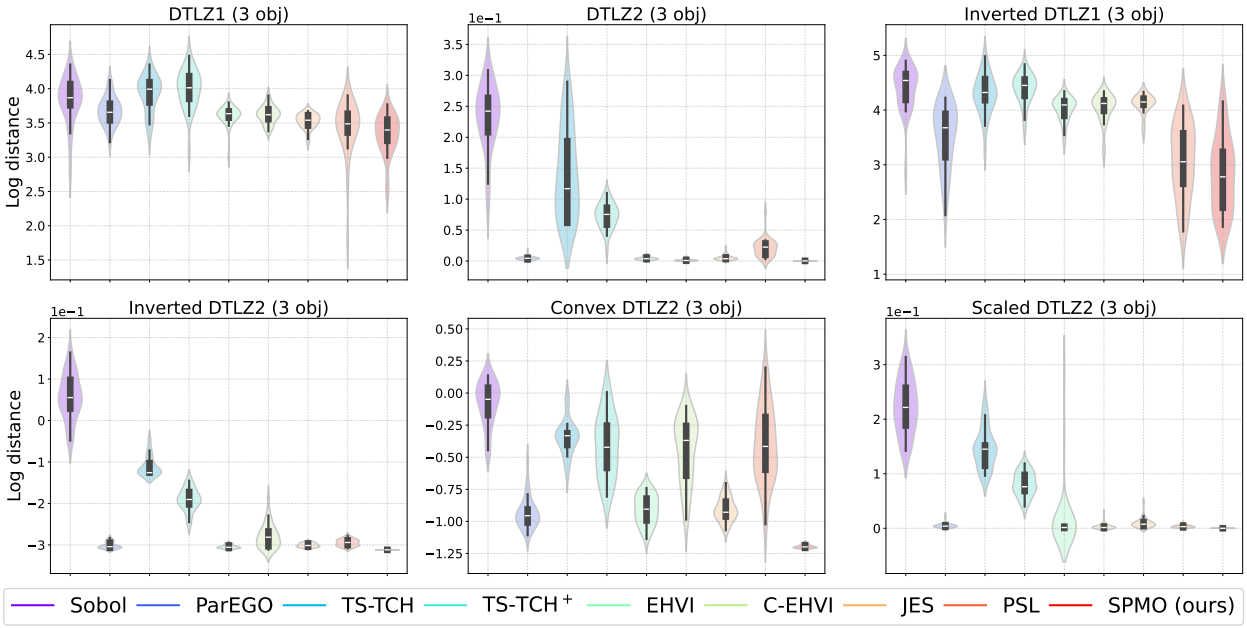
Method	**DTLZ1 (3)** Mean (Std)	**DTLZ2 (3)** Mean (Std)	**Inverted DTLZ1 (3)** Mean (Std)	**Inverted DTLZ2 (3)** Mean (Std)	**Convex DTLZ2 (3)** Mean (Std)	**Scaled DTLZ2 (3)** Mean (Std)	**Sum up** +/∼/−
Sobol	3.8e+0 (3.3e–1)$^+$	2.3e–1 (5.1e–2)$^+$	4.4e+0 (4.0e–1)$^+$	5.7e–2 (5.3e–2)$^+$	-7.7e–2 (1.7e–1)$^+$	2.2e–1 (4.8e–2)$^+$	**6/ 0/ 0**
ParEGO	3.7e+0 (2.0e–1)$^+$	4.7e–3 (3.1e–3)$^+$	3.5e+0 (5.7e–1)$^+$	-3.0e–1 (7.7e–3)$^+$	-9.3e–1 (1.2e–1)$^+$	5.2e–3 (4.7e–3)$^+$	**6/ 0/ 0**
TS-TCH	3.9e+0 (2.6e–1)$^+$	1.4e–1 (7.1e–2)$^+$	4.3e+0 (3.6e–1)$^+$	-1.1e–1 (2.4e–2)$^+$	-3.4e–1 (1.3e–1)$^+$	1.5e–1 (3.6e–2)$^+$	**6/ 0/ 0**
TS-TCH$^+$	4.0e+0 (2.8e–1)$^+$	7.3e–2 (2.0e–2)$^+$	4.4e+0 (2.6e–1)$^+$	-1.9e–1 (2.6e–2)$^+$	-4.1e–1 (2.4e–1)$^+$	8.1e–2 (2.1e–2)$^+$	**6/ 0/ 0**
EHVI	3.6e+0 (1.4e–1)$^+$	4.6e–3 (2.5e–3)$^+$	4.0e+0 (2.1e–1)$^+$	-3.0e–1 (4.3e–3)$^+$	-9.1e–1 (1.1e–1)$^+$	1.8e–2 (6.0e–2)$^+$	**6/ 0/ 0**
C-EHVI	3.6e+0 (1.2e–1)$^+$	1.6e–3 (1.7e–3)$^+$	4.0e+0 (2.6e–1)$^+$	-2.8e–1 (2.9e–2)$^+$	-4.5e–1 (2.4e–1)$^+$	3.2e–3 (5.5e–3)$^+$	**6/ 0/ 0**
JES	3.5e+0 (1.1e–1)$^+$	5.5e–3 (4.2e–3)$^+$	4.1e+0 (1.7e–1)$^+$	-3.0e–1 (4.8e–3)$^+$	-9.1e–1 (9.0e–2)$^+$	9.6e–3 (8.7e–3)$^+$	**6/ 0/ 0**
PSL	3.4e+0 (4.0e–1)$^\sim$	2.1e–2 (1.5e–2)$^+$	3.1e+0 (6.5e–1)$^\sim$	-2.9e–1 (8.1e–3)$^+$	-4.0e–1 (2.8e–1)$^+$	3.5e–3 (2.4e–3)$^+$	**4/ 2/ 0**
SPMO	**3.3e+0 (2.9e–1)**	**2.2e–4 (1.2e–4)**	**2.8e+0 (6.6e–1)**	**-3.1e–1 (1.3e–5)**	**-1.2e+0 (1.4e–2)**	**3.8e–5 (1.6e–5)**	
Method	**DTLZ1 (10)** Mean (Std)	**DTLZ2 (10)** Mean (Std)	**Inverted DTLZ1 (10)** Mean (Std)	**Inverted DTLZ2 (10)** Mean (Std)	**Convex DTLZ2 (10)** Mean (Std)	**Scaled DTLZ2 (10)** Mean (Std)	**Sum up** +/∼/−
Sobol	3.8e+0 (2.4e–1)$^+$	2.4e–1 (4.1e–2)$^+$	5.2e+0 (2.5e–1)$^+$	1.2e+0 (5.4e–2)$^+$	-4.5e–1 (2.2e–1)$^+$	2.4e–1 (5.1e–2)$^+$	**6/ 0/ 0**
ParEGO	3.7e+0 (2.6e–1)$^+$	1.2e–1 (8.3e–2)$^+$	3.5e+0 (4.7e–1)$^\sim$	8.6e–1 (1.9e–2)$^+$	-1.8e+0 (2.3e–1)$^+$	1.1e–1 (6.3e–2)$^+$	**5/ 1/ 0**
TS-TCH	3.8e+0 (3.9e–1)$^+$	2.1e–1 (2.3e–2)$^+$	5.3e+0 (3.7e–1)$^+$	1.0e+0 (1.4e–2)$^+$	-6.7e–1 (2.5e–1)$^+$	2.1e–1 (3.6e–2)$^+$	**6/ 0/ 0**
TS-TCH$^+$	3.9e+0 (2.1e–1)$^+$	8.6e–2 (2.0e–2)$^+$	5.3e+0 (3.0e–1)$^+$	9.9e–1 (3.6e–2)$^+$	-8.3e–1 (2.7e–1)$^+$	8.9e–2 (1.8e–2)$^+$	**6/ 0/ 0**
C-EHVI	3.7e+0 (1.5e–1)$^+$	6.4e–3 (5.1e–3)$^+$	3.9e+0 (5.1e–1)$^+$	8.4e–1 (1.6e–2)$^+$	-5.5e–1 (3.2e–1)$^+$	5.5e–2 (7.5e–2)$^+$	**6/ 0/ 0**
JES	3.4e+0 (2.1e–1)$^+$	1.5e–1 (6.6e–2)$^+$	4.5e+0 (5.3e–1)$^+$	8.6e–1 (2.0e–2)$^+$	-1.7e+0 (3.0e–1)$^+$	1.2e–1 (8.5e–2)$^+$	**6/ 0/ 0**
SPMO	**2.8e+0 (5.5e–1)**	**1.3e–3 (2.6e–3)**	**3.4e+0 (5.1e–1)**	**7.8e–1 (2.4e–2)**	**-3.2e+0 (1.5e–1)**	**1.7e–2 (3.7e–2)**	

Table 6: The HV of the best solution (in terms of its HV value) obtained by the proposed SPMO and the peer methods on the noiseless problems with 3 objectives (**top**) and 10 objectives (**bottom**) on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

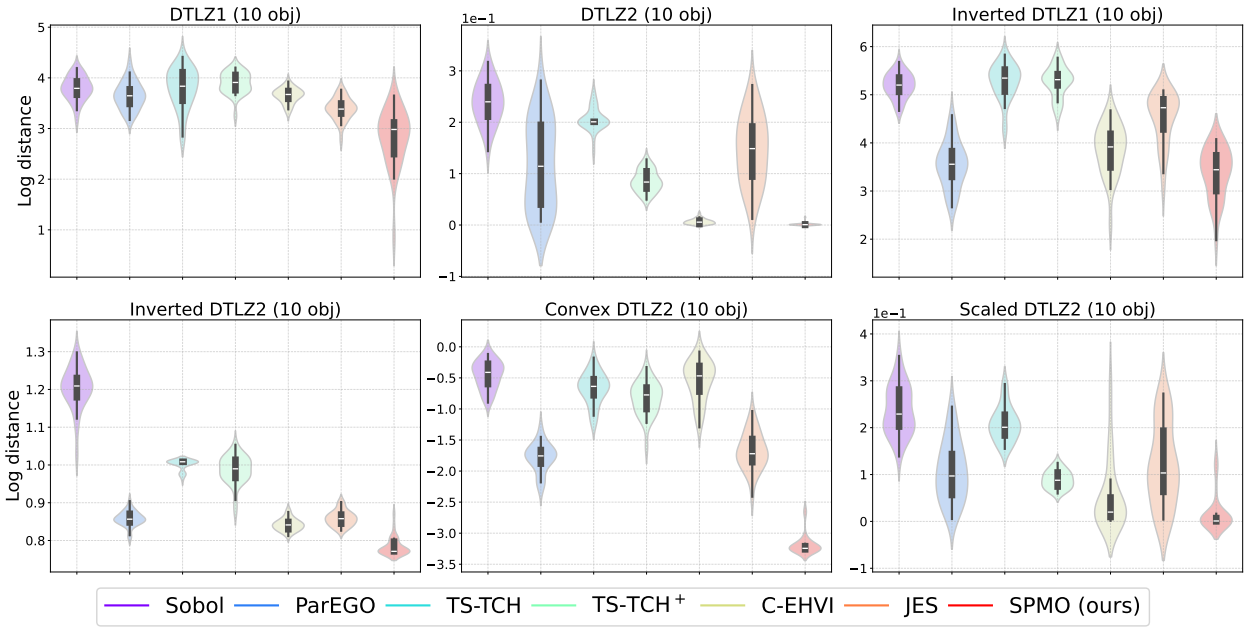
Method	DTLZ1 (3) Mean (Std)	DTLZ2 (3) Mean (Std)	Inverted DTLZ1 (3) Mean (Std)	Inverted DTLZ2 (3) Mean (Std)	Convex DTLZ2 (3) Mean (Std)	Scaled DTLZ2 (3) Mean (Std)	Sum up +/∼/−
Sobol	5.3e+7 (3.3e+6)$^{+}$	3.3e–2 (2.1e–2)$^{+}$	4.4e+7 (5.7e+6)$^{+}$	1.1e–1 (2.5e–2)$^{+}$	1.9e–1 (1.1e–1)$^{+}$	3.7e–2 (1.8e–2)$^{+}$	**6/ 0/ 0**
ParEGO	5.6e+7 (1.5e+6)$^{+}$	1.6e–1 (7.0e–3)$^{+}$	5.5e+7 (4.2e+6)$^{+}$	3.1e–1 (3.7e–3)$^{+}$	7.2e–1 (5.1e–2)$^{+}$	1.6e–1 (9.5e–3)$^{-}$	**5/ 0/ 1**
TS-TCH	5.3e+7 (3.1e+6)$^{+}$	5.0e–2 (3.1e–2)$^{+}$	4.6e+7 (5.2e+6)$^{+}$	2.0e–1 (1.2e–2)$^{+}$	3.7e–1 (8.4e–2)$^{+}$	4.8e–2 (2.5e–2)$^{+}$	**6/ 0/ 0**
TS-TCH^{+}	5.3e+7 (2.4e+6)$^{+}$	1.1e–1 (9.7e–3)$^{+}$	4.5e+7 (4.3e+6)$^{+}$	2.4e–1 (1.4e–2)$^{+}$	4.1e–1 (1.5e–1)$^{+}$	1.0e–1 (9.0e–3)$^{+}$	**6/ 0/ 0**
EHVI	5.7e+7 (7.7e+5)$^{+}$	1.6e–1 (4.6e–3)$^{+}$	5.1e+7 (2.1e+6)$^{+}$	3.1e–1 (2.2e–3)$^{+}$	6.8e–1 (4.6e–2)$^{+}$	1.3e–1 (3.4e–2)$^{-}$	**5/ 0/ 1**
C-EHVI	5.7e+7 (8.3e+5)$^{+}$	1.3e–1 (1.3e–2)$^{+}$	5.1e+7 (2.4e+6)$^{+}$	3.0e–1 (1.4e–2)$^{+}$	4.5e–1 (1.4e–1)$^{+}$	1.4e–1 (1.7e–2)$^{-}$	**5/ 0/ 1**
JES	5.6e+7 (8.2e+5)$^{+}$	1.5e–1 (8.3e–3)$^{+}$	5.0e+7 (2.0e+6)$^{+}$	3.0e–1 (2.4e–3)$^{+}$	7.1e–1 (3.6e–2)$^{+}$	1.5e–1 (8.1e–3)$^{-}$	**5/ 0/ 1**
PSL	5.7e+7 (1.6e+6)$^{\sim}$	1.2e–1 (2.9e–2)$^{+}$	5.8e+7 (3.5e+6)$^{\sim}$	3.0e–1 (3.9e–3)$^{+}$	4.1e–1 (1.7e–1)$^{+}$	**1.6e–1 (5.8e–3)**$^{-}$	**3/ 2/ 1**
SPMO	**5.8e+7 (1.7e+6)**	**1.7e–1 (5.3e–3)**	**5.9e+7 (3.5e+6)**	**3.1e–1 (6.3e–6)**	**7.9e–1 (5.1e–3)**	1.2e–1 (7.5e–3)	
Method	**DTLZ1 (10)** Mean (Std)	**DTLZ2 (10)** Mean (Std)	**Inverted DTLZ1 (10)** Mean (Std)	**Inverted DTLZ2 (10)** Mean (Std)	**Convex DTLZ2 (10)** Mean (Std)	**Scaled DTLZ2 (10)** Mean (Std)	**Sum up** +/∼/−
Sobol	8.7e+25 (3.8e+24)$^{+}$	4.7e–2 (2.5e–2)$^{+}$	2.4e+25 (9.4e+24)$^{+}$	5.9e-11 (3.2e-10)$^{+}$	7.7e–1 (2.4e–1)$^{+}$	5.0e–2 (2.1e–2)$^{+}$	**6/ 0/ 0**
ParEGO	9.2e+25 (3.3e+24)$^{+}$	1.2e–1 (9.6e–2)$^{+}$	7.8e+25 (1.1e+25)$^{\sim}$	2.2e–5 (1.4e–5)$^{+}$	1.9e+0 (1.0e–1)$^{+}$	1.3e–1 (8.3e–2)$^{+}$	**5/ 1/ 0**
TS-TCH	8.8e+25 (5.1e+24)$^{+}$	4.3e–2 (4.0e–2)$^{+}$	2.2e+25 (1.3e+25)$^{+}$	1.6e–8 (2.6e–8)$^{+}$	1.0e+0 (2.5e–1)$^{+}$	4.1e–2 (2.5e–2)$^{+}$	**6/ 0/ 0**
TS-TCH^{+}	8.8e+25 (2.5e+24)$^{+}$	1.7e–1 (2.0e–2)$^{+}$	2.2e+25 (1.1e+25)$^{+}$	8.0e–7 (2.5e–6)$^{+}$	1.2e+0 (2.2e–1)$^{+}$	1.7e–1 (2.7e–2)$^{+}$	**6/ 0/ 0**
C-EHVI	9.2e+25 (1.0e+24)$^{+}$	2.6e–1 (3.2e–2)$^{+}$	7.0e+25 (1.3e+25)$^{+}$	3.4e–5 (1.4e–5)$^{+}$	9.1e–1 (3.3e–1)$^{+}$	2.0e–1 (9.0e–2)$^{+}$	**6/ 0/ 0**
JES	9.3e+25 (1.9e+24)$^{+}$	9.9e–2 (7.2e–2)$^{+}$	4.9e+25 (1.8e+25)$^{+}$	2.1e–5 (1.1e–5)$^{+}$	1.8e+0 (1.3e–1)$^{+}$	1.3e–1 (9.7e–2)$^{+}$	**6/ 0/ 0**
SPMO	**9.7e+25 (3.6e+24)**	**2.9e–1 (3.4e–2)**	**8.2e+25 (9.3e+24)**	**1.3e–4 (5.1e–5)**	**2.3e+0 (3.5e–2)**	**2.6e–1 (4.6e–2)**	

Table 7: The HV of all the solutions obtained by the proposed SPMO and the peer methods on the noiseless problems with 3 objectives (**top**) and 10 objectives (**bottom**) on 30 independent runs, respectively. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO, respectively.

Method	DTLZ1 (3) Mean (Std)	DTLZ2 (3) Mean (Std)	Inverted DTLZ1 (3) Mean (Std)	Inverted DTLZ2 (3) Mean (Std)	Convex DTLZ2 (3) Mean (Std)	Scaled DTLZ2 (3) Mean (Std)	Sum up +/∼/−
Sobol	6.3e+7 (2.8e+5)$^{\sim}$	4.9e–2 (2.7e–2)$^{+}$	5.2e+7 (2.9e+6)$^{+}$	2.1e–1 (2.6e–2)$^{+}$	2.5e–1 (1.3e–1)$^{+}$	6.2e–2 (2.5e–2)$^{+}$	**5/ 1/ 0**
ParEGO	6.3e+7 (1.1e+6)$^{-}$	5.4e–1 (5.2e–2)$^{-}$	5.8e+7 (2.6e+6)$^{+}$	6.7e–1 (1.1e–2)$^{-}$	1.0e+0 (9.0e–2)$^{\sim}$	5.4e–1 (7.5e–2)$^{-}$	**1/ 1/ 4**
TS-TCH	6.2e+7 (7.8e+5)$^{\sim}$	7.5e–2 (4.2e–2)$^{+}$	5.2e+7 (3.3e+6)$^{+}$	4.1e–1 (1.2e–2)$^{-}$	5.4e–1 (1.1e–1)$^{+}$	8.8e–2 (4.6e–2)$^{+}$	**4/ 1/ 1**
TS-TCH^{+}	6.1e+7 (1.4e+6)$^{+}$	2.1e–1 (1.8e–2)$^{+}$	4.9e+7 (3.4e+6)$^{+}$	3.5e–1 (1.8e–2)$^{+}$	5.1e–1 (2.0e–1)$^{+}$	2.2e–1 (2.2e–2)$^{-}$	**5/ 0/ 1**
EHVI	6.4e+7 (1.0e+5)$^{-}$	**6.4e–1 (2.2e–2)**$^{-}$	5.8e+7 (2.4e+6)$^{+}$	**7.0e–1 (3.9e–3)**$^{-}$	**1.0e+0 (6.9e–2)**$^{\sim}$	2.6e–1 (7.7e–2)$^{-}$	**1/ 1/ 4**
C-EHVI	6.3e+7 (6.8e+5)$^{-}$	3.3e–1 (5.4e–2)$^{+}$	5.7e+7 (3.2e+6)$^{+}$	3.2e–1 (3.5e–2)$^{+}$	5.7e–1 (2.0e–1)$^{+}$	3.3e–1 (6.5e–2)$^{-}$	**4/ 0/ 2**
JES	**6.4e+7 (5.0e+4)**$^{-}$	5.6e–1 (5.3e–2)$^{-}$	6.1e+7 (1.6e+6)$^{\sim}$	6.6e–1 (1.1e–2)$^{-}$	1.0e+0 (6.3e–2)$^{\sim}$	5.0e–1 (8.5e–2)$^{-}$	**0/ 2/ 4**
PSL	6.3e+7 (7.8e+5)$^{-}$	2.6e–1 (1.0e–1)$^{+}$	6.0e+7 (2.1e+6)$^{+}$	6.8e–1 (1.2e–2)$^{-}$	5.9e–1 (2.4e–1)$^{+}$	**5.8e–1 (5.9e–2)**$^{-}$	**3/ 0/ 3**
SPMO	6.2e+7 (1.0e+6)	4.6e–1 (6.3e–2)	**6.1e+7 (1.5e+6)**	3.7e–1 (1.5e–2)	1.0e+0 (4.2e–2)	1.6e–1 (5.2e–2)	
Method	**DTLZ1 (10)** Mean (Std)	**DTLZ2 (10)** Mean (Std)	**Inverted DTLZ1 (10)** Mean (Std)	**Inverted DTLZ2 (10)** Mean (Std)	**Convex DTLZ2 (10)** Mean (Std)	**Scaled DTLZ2 (10)** Mean (Std)	**Sum up** +/∼/−
Sobol	2.4e+28 (1.3e+28)$^{-}$	3.6e–1 (1.7e–1)$^{+}$	5.3e+25 (5.9e+25)$^{+}$	5.9e-11 (3.2e-10)$^{+}$	5.2e+0 (3.2e+0)$^{+}$	3.6e–1 (1.6e–1)$^{+}$	**5/ 0/ 1**
ParEGO	9.0e+26 (9.8e+26)$^{+}$	1.7e+0 (3.5e+0)$^{+}$	3.6e+27 (1.3e+28)$^{+}$	7.2e–4 (4.9e–4)$^{+}$	2.7e+2 (2.2e+2)$^{+}$	6.8e–1 (7.6e–1)$^{+}$	**6/ 0/ 0**
TS-TCH	**2.9e+28 (2.5e+28)**$^{-}$	1.8e–1 (2.0e–1)$^{+}$	4.7e+25 (4.4e+25)$^{+}$	1.9e–8 (3.3e–8)$^{+}$	2.8e+1 (2.4e+1)$^{+}$	1.5e–1 (1.0e–1)$^{+}$	**5/ 0/ 1**
TS-TCH^{+}	1.0e+26 (1.3e+24)$^{+}$	3.7e–1 (3.8e–2)$^{+}$	2.2e+25 (1.1e+25)$^{+}$	8.2e–7 (2.5e–6)$^{+}$	1.8e+0 (3.2e–1)$^{+}$	3.7e–1 (5.1e–2)$^{+}$	**6/ 0/ 0**
C-EHVI	1.0e+26 (1.6e+24)$^{+}$	5.2e–1 (1.2e–1)$^{+}$	7.1e+25 (1.3e+25)$^{+}$	6.6e–5 (2.6e–5)$^{+}$	1.3e+0 (4.0e–1)$^{+}$	3.8e–1 (2.2e–1)$^{+}$	**6/ 0/ 0**
JES	8.8e+26 (1.3e+27)$^{+}$	6.3e–1 (1.1e+0)$^{+}$	5.7e+27 (1.9e+28)$^{+}$	7.5e–4 (3.9e–4)$^{+}$	2.8e+2 (1.9e+2)$^{+}$	1.5e+0 (2.2e+0)$^{+}$	**6/ 0/ 0**
SPMO	1.2e+28 (1.5e+28)	**5.2e+1 (3.3e+1)**	**2.3e+28 (5.3e+28)**	**1.4e–1 (1.5e–1)**	**6.3e+3 (8.0e+3)**	**5.2e+1 (6.5e+1)**	

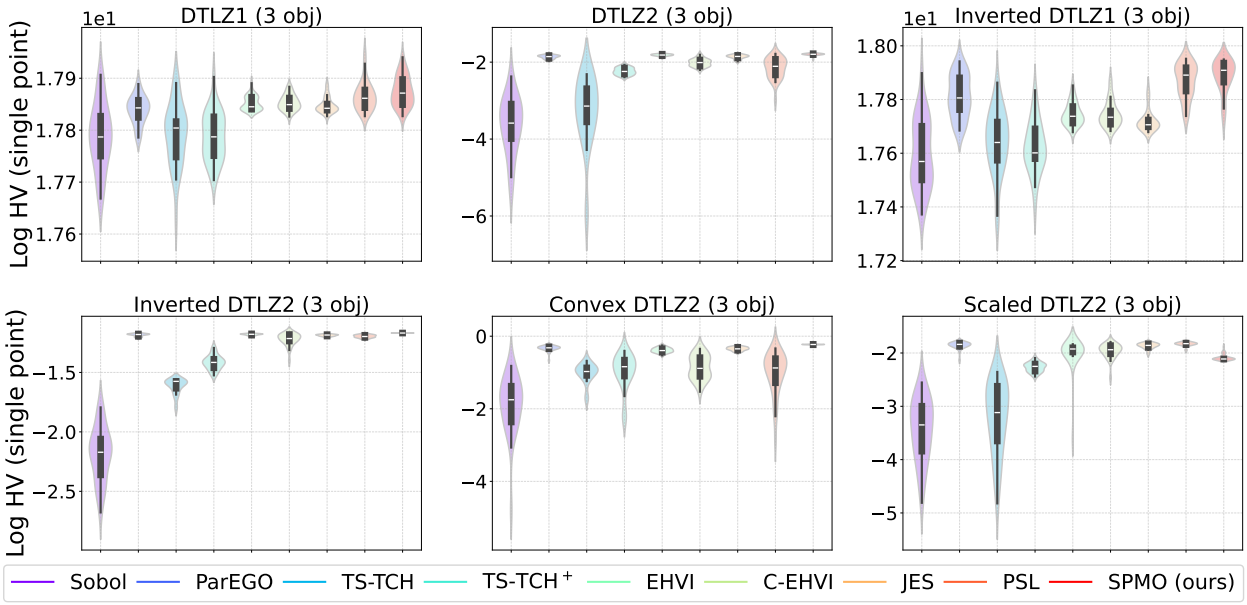


(a) Violin plots of the distance-based metric (log distance) obtained by each method on problems with 3 objectives.

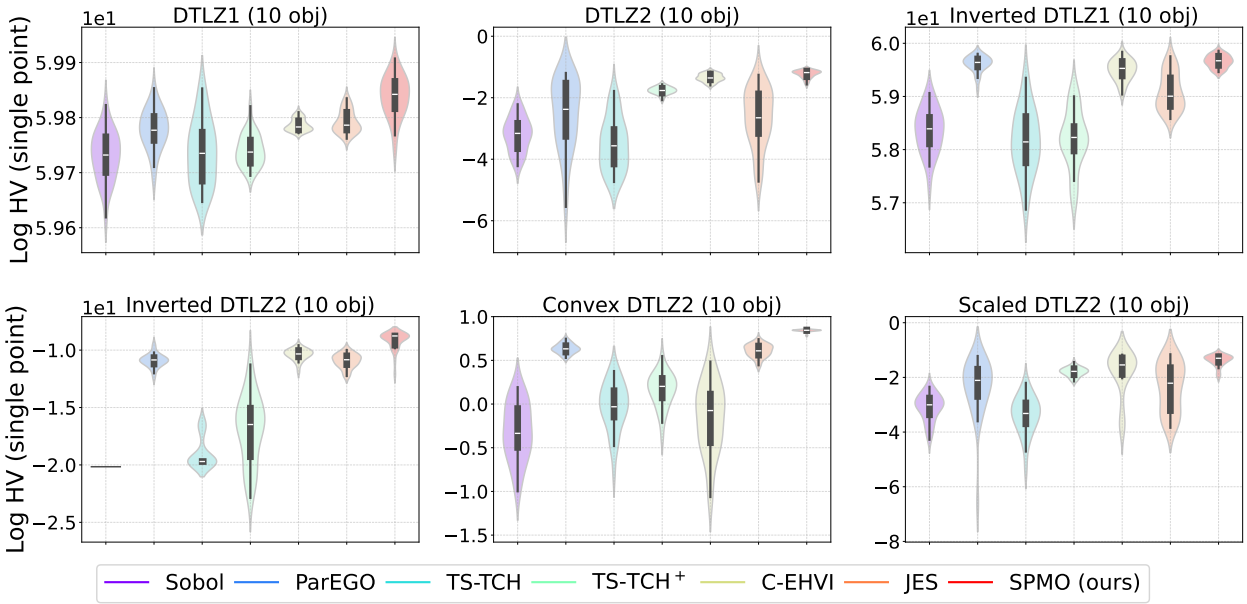


(b) Violin plots of the distance-based metric (log distance) obtained by each method on problems with 10 objectives.

Figure 7: Violin plots of the distance-based metric (log distance) obtained by the proposed SPMO and the peer methods on the noiseless problems with 3 and 10 objectives. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

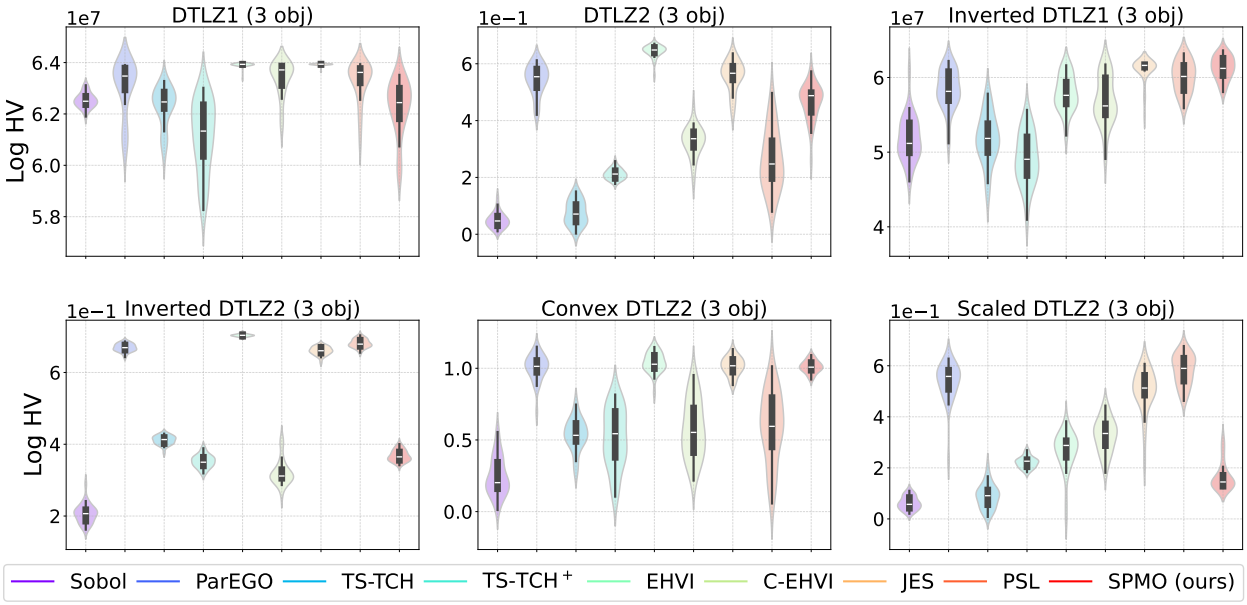


(a) Violin plots of the HV of the best solution obtained by each method on problems with 3 objectives.

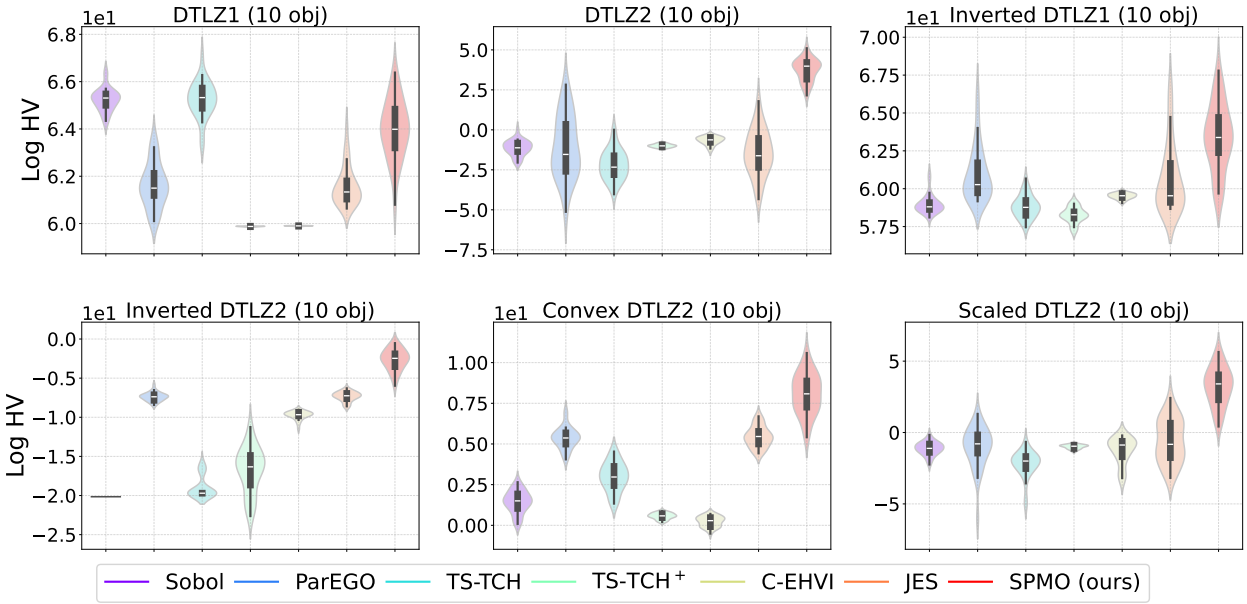


(b) Violin plots of the HV of the best solution obtained by each method on problems with 10 objectives.

Figure 8: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the proposed SPMO and the peer methods on the noiseless problems with 3 and 10 objectives. Each violin represents the distribution of HV values obtained by a method over 30 independent runs.

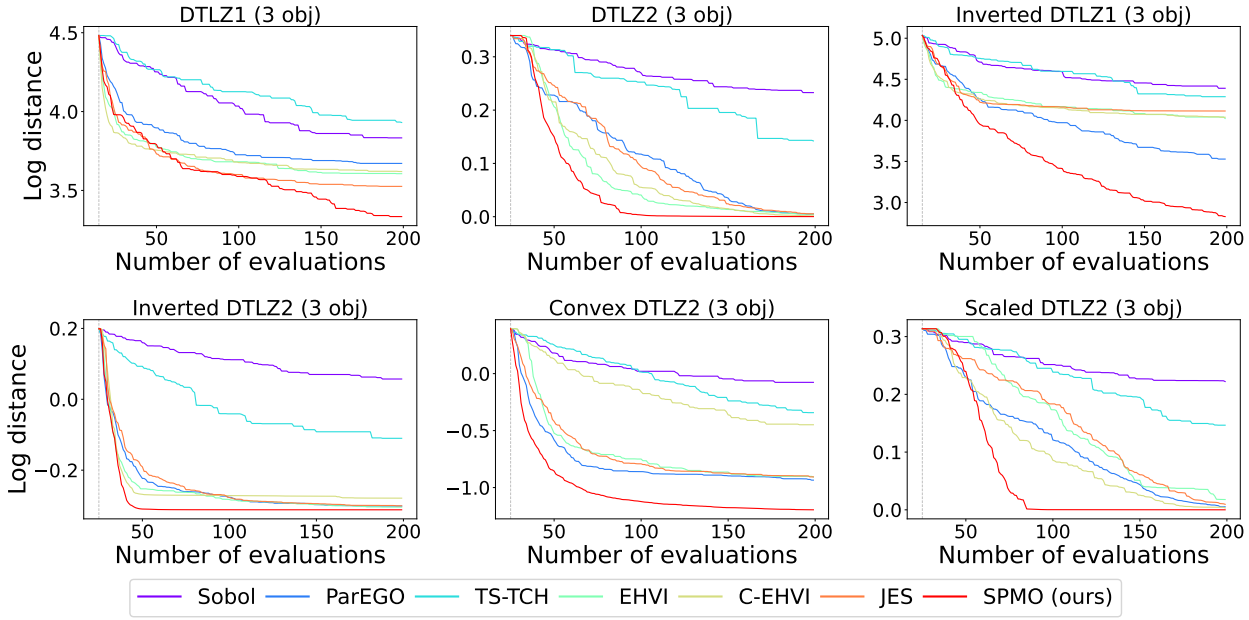


(a) Violin plots of the HV of all evaluated solutions obtained by each method on problems with 3 objectives.

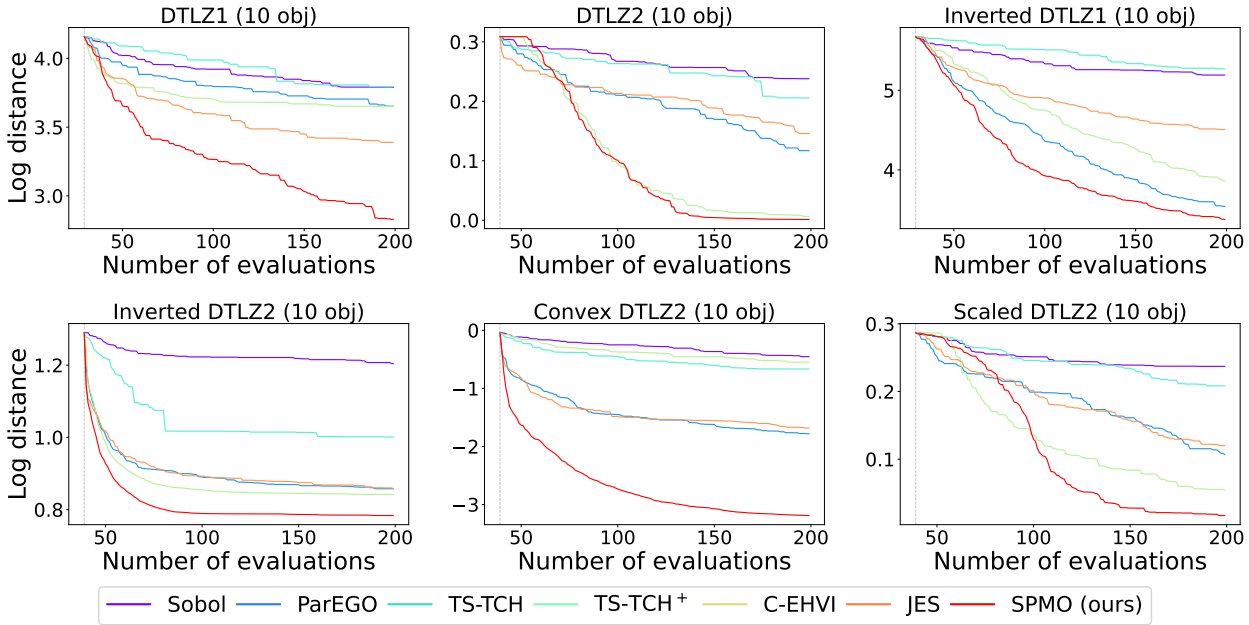


(b) Violin plots of the HV of all evaluated solutions obtained by each method on problems with 10 objectives.

Figure 9: Violin plots of the HV of all evaluated solutions obtained by the proposed SPMO and the peer methods on the noiseless problems with 3 objectives (**top**) and 10 objectives (**bottom**), respectively. Each violin represents the distribution of HV values obtained by a method over 30 independent runs.

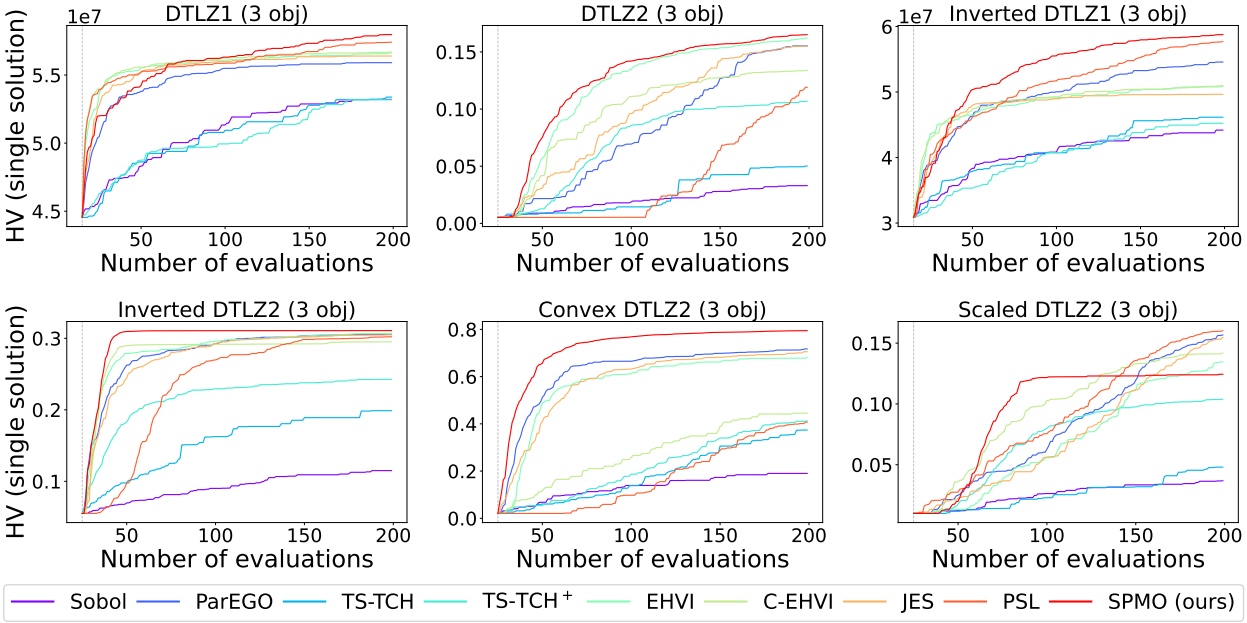


(a) Trajectories of the distance metric on the problems with 3 objectives.

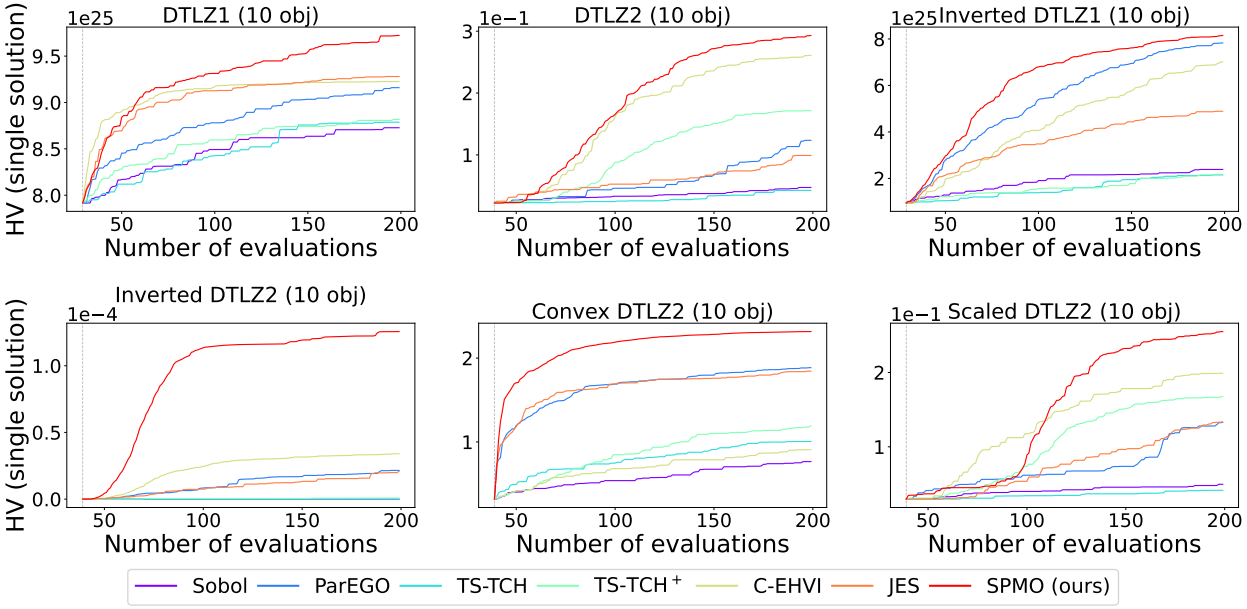


(b) Trajectories of the distance metric on the problems with 10 objectives.

Figure 10: Trajectories of the distance metric obtained by the SPMO and the peer methods on the noiseless problems with 3 and 10 objectives. Each coloured line represents the mean distance of the closest solution to the utopian point on 30 independent runs (after the initial Sobol samples, represented by the dashed grey line).

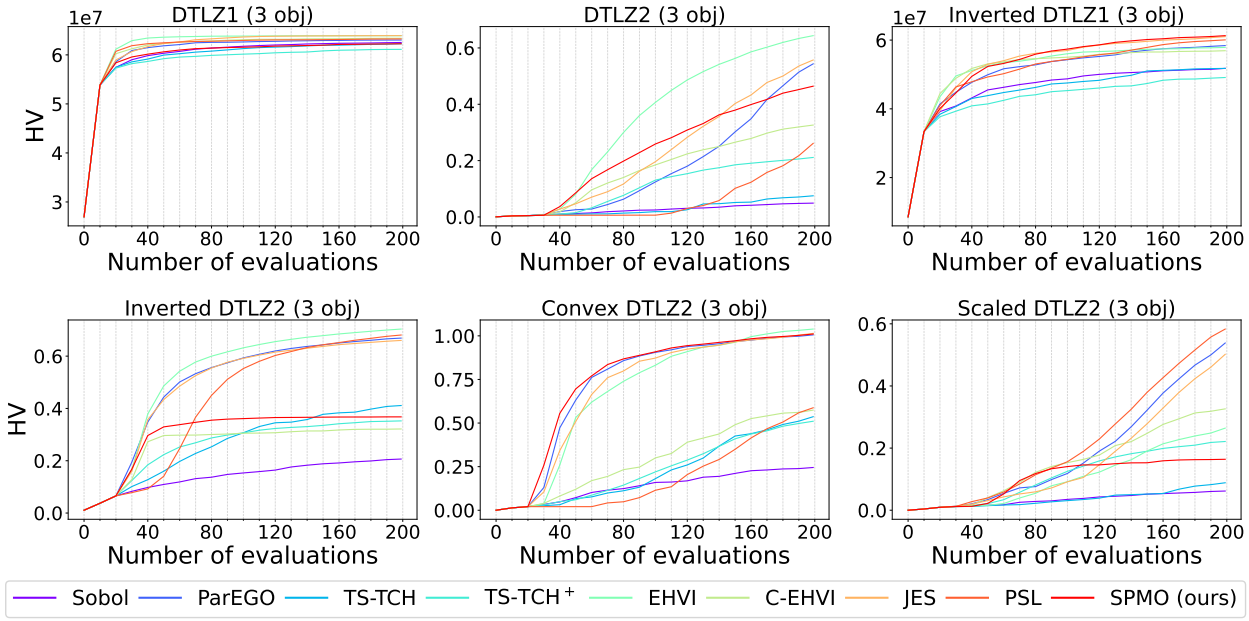


(a) Trajectories of the HV of the best solution on the problems with 3 objectives.

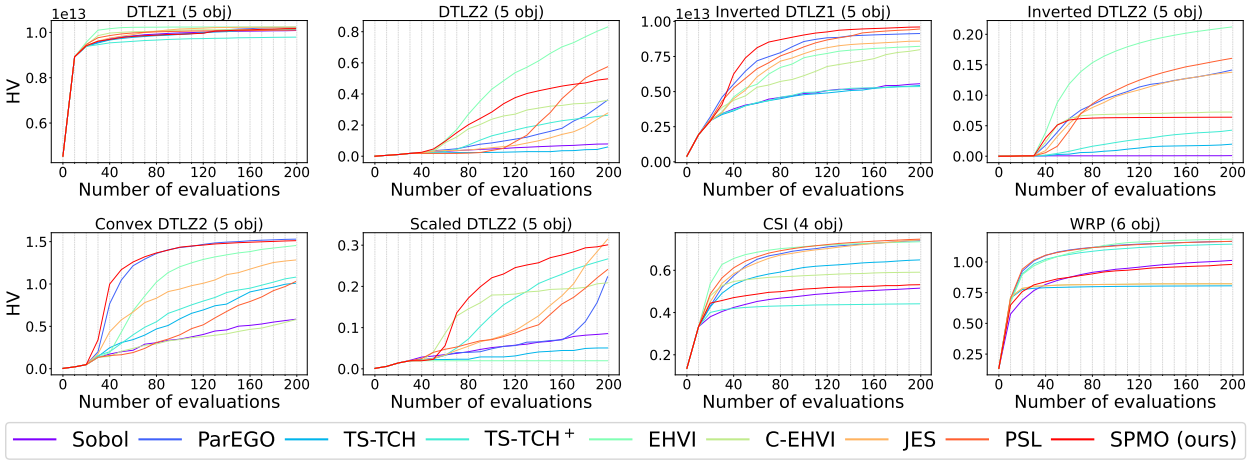


(b) Trajectories of the HV of the best solution on the problems with 10 objectives.

Figure 11: Trajectories of the HV of the best solution obtained by the SPMO and the peer methods on the benchmark problems with 3 and 10 objectives. Each coloured line represents the mean metric value on 30 independent runs (after the initial Sobol samples, represented by the dashed grey line).



(a) Trajectories of the HV of all evaluated solutions on the problems with 3 objectives.



(b) Trajectories of the HV of all evaluated solutions on the benchmark problems with 5 objectives and two real-world problems.

Figure 12: Trajectories of the HV of all evaluated solutions obtained by the SPMO and the peer methods on the benchmark problems with 3 and 5 objectives and the two real-world problems. Each coloured line represents the mean metric value on 30 independent runs. Note that the results are reported every 10 evaluations due to the high computational cost of HV calculation.

Table 8: Results of the distance-based metric (log distance) obtained by obtained by the SPMO and the peer methods on DTLZ3–DTLZ7 with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	**DTLZ3** Mean (Std)	**DTLZ4** Mean (Std)	**DTLZ5** Mean (Std)	**DTLZ6** Mean (Std)	**DTLZ7** Mean (Std)	**Sum up** +/∼/−
Sobol	6.3e+0 (1.6e–1)$^{+}$	2.7e–1 (6.0e–2)$^{+}$	2.5e–1 (4.1e–2)$^{+}$	2.2e+0 (2.1e–2)$^{+}$	3.1e+0 (5.0e–2)$^{+}$	**5/ 0/ 0**
ParEGO	5.5e+0 (1.1e–1)$^{\sim}$	2.3e–1 (7.1e–2)$^{+}$	4.0e–2 (3.1e–2)$^{+}$	4.3e–2 (1.6e–1)$^{+}$	2.2e+0 (8.0e–2)$^{+}$	**4/ 1/ 0**
TS-TCH	6.3e+0 (1.6e–1)$^{+}$	2.6e–1 (5.0e–2)$^{+}$	2.3e–1 (5.9e–2)$^{+}$	2.2e+0 (1.6e–2)$^{+}$	3.1e+0 (4.8e–2)$^{+}$	**5/ 0/ 0**
EHVI	**5.4e+0 (4.6e–2)**$^{-}$	8.7e–2 (9.4e–2)$^{+}$	3.0e–1 (6.0e–2)$^{+}$	-9.4e–7 (8.4e–7)$^{+}$	1.7e+0 (8.5e–2)$^{+}$	**4/ 0/ 1**
JES	5.5e+0 (5.0e–2)$^{\sim}$	2.3e–1 (8.1e–2)$^{+}$	5.2e–2 (5.8e–2)$^{+}$	2.1e–2 (1.1e–1)$^{+}$	2.2e+0 (5.8e–2)$^{+}$	**4/ 1/ 0**
SPMO	5.5e+0 (1.8e–1)	**1.2e–2 (1.9e–2)**	**3.2e–3 (5.0e–3)**	**-1.9e–6 (5.6e–7)**	**1.7e+0 (1.1e–1)**	

Table 9: The HV of the best solution (in terms of its HV value) obtained by SPMO and the peer methods on the DTLZ3–DTLZ7 with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	**DTLZ3** Mean (Std)	**DTLZ4** Mean (Std)	**DTLZ5** Mean (Std)	**DTLZ6** Mean (Std)	**DTLZ7** Mean (Std)	**Sum up** +/∼/−
Sobol	9.2e+19 (1.1e+18)$^{+}$	2.2e–2 (1.4e–2)$^{+}$	8.4e+4 (9.5e+2)$^{\sim}$	8.7e+3 (8.2e+2)$^{+}$	-0.0e+0 (0.0e+0)$^{+}$	**4/ 1/ 0**
ParEGO	9.7e+19 (5.2e+17)$^{\sim}$	2.5e–2 (3.0e–2)$^{+}$	**8.9e+4 (1.4e+3)**$^{-}$	8.8e+4 (3.8e+3)$^{-}$	2.8e+5 (2.7e+4)$^{+}$	**2/ 1/ 2**
TS-TCH	9.3e+19 (1.0e+18)$^{+}$	1.3e–2 (1.1e–2)$^{+}$	8.6e+4 (1.1e+3)$^{-}$	9.3e+3 (9.9e+2)$^{+}$	-0.0e+0 (0.0e+0)$^{+}$	**4/ 0/ 1**
EHVI	**9.8e+19 (1.1e+17)**$^{-}$	1.3e–1 (5.6e–2)$^{+}$	8.3e+4 (1.6e+3)$^{+}$	**9.0e+4 (1.3e–2)**$^{-}$	3.8e+5 (2.2e+4)$^{\sim}$	**2/ 1/ 2**
JES	9.8e+19 (1.4e+17)$^{-}$	3.6e–2 (3.5e–2)$^{+}$	8.9e+4 (1.5e+3)$^{-}$	8.8e+4 (3.5e+3)$^{-}$	2.8e+5 (2.0e+4)$^{+}$	**2/ 0/ 3**
SPMO	9.7e+19 (1.0e+18)	**1.6e–1 (2.5e–2)**	8.5e+4 (2.6e+3)	8.3e+4 (2.9e+3)	**3.8e+5 (2.8e+4)**	

Table 10: The HV of all evaluated solutions obtained by SPMO and the peer methods on DTLZ3–DTLZ7 with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	**DTLZ3** Mean (Std)	**DTLZ4** Mean (Std)	**DTLZ5** Mean (Std)	**DTLZ6** Mean (Std)	**DTLZ7** Mean (Std)	**Sum up** +/∼/−
Sobol	3.9e+20 (1.7e+20)$^{\sim}$	4.9e–2 (2.4e–2)$^{+}$	2.1e+5 (1.1e+5)$^{\sim}$	1.4e+5 (1.7e+4)$^{+}$	-0.0e+0 (0.0e+0)$^{+}$	**3/ 2/ 0**
ParEGO	2.1e+20 (2.1e+20)$^{\sim}$	5.3e–2 (7.4e–2)$^{+}$	2.4e+5 (8.7e+4)$^{\sim}$	3.9e+5 (1.6e+5)$^{-}$	9.1e+5 (4.1e+5)$^{+}$	**2/ 2/ 1**
TS-TCH	**5.0e+20 (3.0e+20)**$^{-}$	2.1e–2 (1.8e–2)$^{+}$	1.8e+5 (8.2e+4)$^{+}$	1.8e+5 (4.2e+4)$^{+}$	-0.0e+0 (0.0e+0)$^{+}$	**4/ 0/ 1**
EHVI	1.7e+20 (1.5e+20)$^{\sim}$	9.3e–1 (7.1e–1)$^{\sim}$	3.5e+5 (1.8e+5)$^{-}$	**4.2e+5 (1.2e+5)**$^{-}$	1.9e+6 (5.3e+5)$^{\sim}$	**0/ 3/ 2**
JES	2.7e+20 (2.7e+20)$^{\sim}$	8.8e–2 (1.3e–1)$^{+}$	**3.6e+5 (3.5e+5)**$^{\sim}$	4.0e+5 (1.6e+5)$^{-}$	1.0e+6 (4.3e+5)$^{+}$	**2/ 2/ 1**
SPMO	4.2e+20 (7.0e+20)	**1.1e+0 (9.3e–1)**	2.5e+5 (1.4e+5)	3.1e+5 (1.9e+5)	**2.3e+6 (1.3e+6)**	

F.2 Noisy Cases

In this section, we present the results on the noisy problems. Tables 11, 12 and 13 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value) and the HV of all evaluated solutions obtained by the SPMO and the peer methods, respectively. Figures 13, 14 and 15 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 11, 12 and 13, respectively. Figure 16 shows trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO and the peer methods.

Table 11: Results of the distance-based metric (log distance) obtained by the SPMO and the peer methods on the noisy problems with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	3.7e+0 (4.2e–1)$^+$	1.9e–1 (5.6e–2)$^+$	4.7e+0 (3.4e–1)$^+$	6.1e–1 (6.1e–2)$^+$	-2.9e–1 (2.2e–1)$^+$	2.2e–1 (4.4e–2)$^+$	**6/ 0/ 0**
NParEGO	3.6e+0 (3.1e–1)$^+$	1.0e–1 (8.2e–2)$^+$	3.1e+0 (3.8e–1)$^+$	1.8e–1 (5.0e–2)$^+$	-1.3e+0 (2.7e–1)$^+$	5.2e–2 (5.9e–2)$^+$	**6/ 0/ 0**
TS-TCH	3.8e+0 (3.5e–1)$^+$	2.0e–1 (4.2e–2)$^+$	4.9e+0 (3.2e–1)$^+$	4.6e–1 (5.5e–2)$^+$	-4.7e–1 (2.3e–1)$^+$	2.1e–1 (5.0e–2)$^+$	**6/ 0/ 0**
NEHVI	3.5e+0 (1.4e–1)$^+$	-1.1e–1 (8.1e–2)$^+$	4.0e+0 (5.8e–1)$^+$	1.5e–1 (3.3e–2)$^+$	-1.3e+0 (3.0e–1)$^+$	2.9e–1 (9.5e–2)$^+$	**6/ 0/ 0**
JES	3.4e+0 (1.2e–1)$^+$	1.3e–1 (1.2e–1)$^+$	4.5e+0 (1.1e–1)$^+$	1.9e–1 (6.5e–2)$^+$	-6.9e–1 (3.5e–1)$^+$	1.0e–1 (8.7e–2)$^+$	**6/ 0/ 0**
SPMO	**3.0e+0 (6.0e–1)**	**-1.8e–1 (5.8e–2)**	**2.9e+0 (4.8e–1)**	**5.8e–2 (6.8e–2)**	**-2.1e+0 (3.3e–1)**	**-1.9e–1 (1.8e–1)**	

Table 12: The HV of the best solution (in terms of its HV value) obtained by SPMO and the peer methods on the noisy problems with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	8.6e+12 (5.2e+11)$^+$	5.1e–2 (3.0e–2)$^+$	5.4e+12 (1.2e+12)$^+$	8.2e–4 (1.5e–3)$^+$	4.0e–1 (1.8e–1)$^+$	3.8e–2 (1.9e–2)$^+$	**6/ 0/ 0**
NParEGO	9.1e+12 (2.5e+11)$^+$	1.0e–1 (4.9e–2)$^+$	9.0e+12 (4.1e+11)$^+$	5.8e–2 (1.6e–2)$^+$	1.2e+0 (2.0e–1)$^+$	1.3e–1 (4.0e–2)$^+$	**6/ 0/ 0**
TS-TCH	8.5e+12 (5.1e+11)$^+$	5.0e–2 (1.4e–2)$^+$	4.8e+12 (1.2e+12)$^+$	8.1e–3 (5.4e–3)$^+$	5.3e–1 (1.7e–1)$^+$	3.1e–2 (1.6e–2)$^+$	**6/ 0/ 0**
NEHVI	9.1e+12 (1.5e+11)$^+$	2.8e–1 (7.6e–2)$^+$	7.4e+12 (1.2e+12)$^+$	6.7e–2 (1.0e–2)$^+$	1.2e+0 (2.1e–1)$^+$	1.2e–2 (1.0e–2)$^+$	**6/ 0/ 0**
JES	9.0e+12 (1.3e+11)$^+$	8.7e–2 (7.2e–2)$^+$	6.1e+12 (3.2e+11)$^+$	5.8e–2 (2.0e–2)$^+$	7.8e–1 (3.1e–1)$^+$	1.0e–1 (5.7e–2)$^+$	**6/ 0/ 0**
SPMO	**9.4e+12 (4.4e+11)**	**3.2e–1 (4.9e–2)**	**9.2e+12 (4.2e+11)**	**1.0e–1 (2.7e–2)**	**1.8e+0 (2.2e–1)**	**3.8e–1 (1.6e–1)**	

Table 13: The HV of all the solutions obtained by the proposed SPMO and the peer methods on the noisy problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	1.0e+13 (4.7e+10)$^\sim$	1.1e–1 (3.6e–2)$^+$	5.9e+12 (9.1e+11)$^+$	8.5e–4 (1.5e–3)$^+$	6.1e–1 (2.5e–1)$^+$	8.6e–2 (2.8e–2)$^+$	**5/ 1/ 0**
NParEGO	1.0e+13 (1.7e+11)$^\sim$	2.3e–1 (1.4e–1)$^+$	9.3e+12 (3.8e+11)$^+$	1.6e–1 (2.6e–2)$^+$	2.3e+0 (2.6e–1)$^+$	3.3e–1 (1.6e–1)$^+$	**5/ 1/ 0**
TS-TCH	1.0e+13 (1.4e+11)$^\sim$	1.1e–1 (2.9e–2)$^+$	5.3e+12 (1.0e+12)$^+$	1.6e–2 (7.9e–3)$^+$	9.5e–1 (2.3e–1)$^+$	5.9e–2 (3.1e–2)$^+$	**5/ 1/ 0**
NEHVI	**1.0e+13 (1.7e+9)**$^-$	**9.1e–1 (1.7e–1)**$^-$	7.7e+12 (1.1e+12)$^+$	**2.1e–1 (1.8e–2)**$^-$	2.1e+0 (3.2e–1)$^+$	1.5e–2 (1.3e–2)$^+$	**3/ 0/ 3**
JES	1.0e+13 (8.7e+9)$^-$	2.1e–1 (2.0e–1)$^+$	8.5e+12 (5.5e+11)$^+$	1.7e–1 (2.6e–2)$^\sim$	1.4e+0 (3.9e–1)$^+$	2.3e–1 (1.3e–1)$^+$	**4/ 1/ 1**
SPMO	1.0e+13 (1.5e+11)	6.1e–1 (9.9e–2)	**9.5e+12 (3.2e+11)**	1.8e–1 (1.6e–2)	**2.7e+0 (1.9e–1)**	**4.4e–1 (1.8e–1)**	

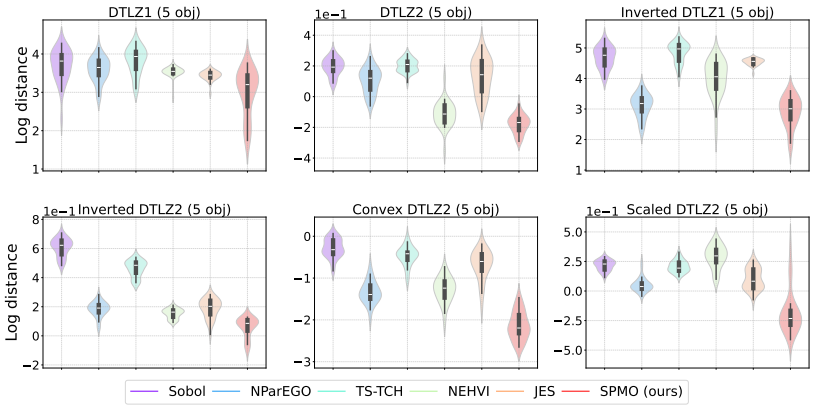


Figure 13: Violin plots of the distance-based metric (log distance) obtained by the six methods on the noisy problems with five objectives. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

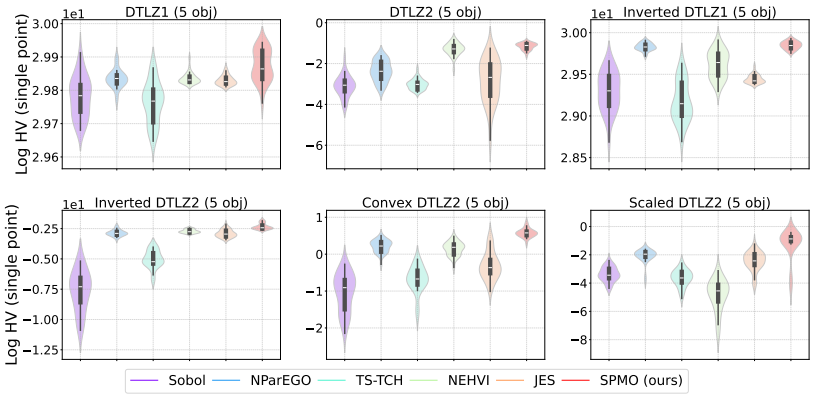


Figure 14: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the six methods on the noisy problems with five objectives. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.

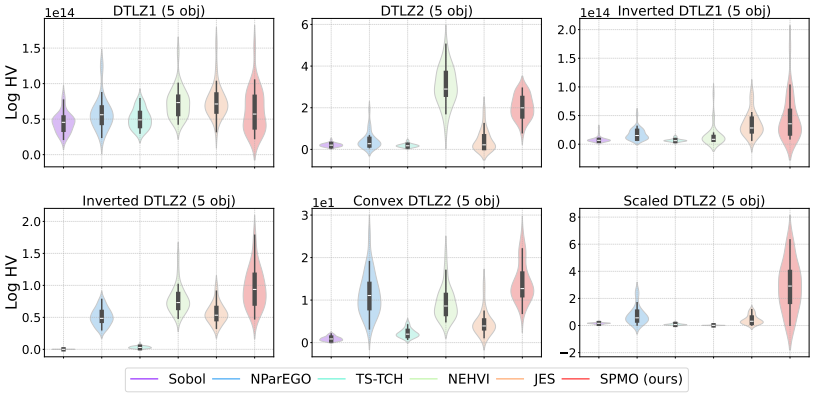
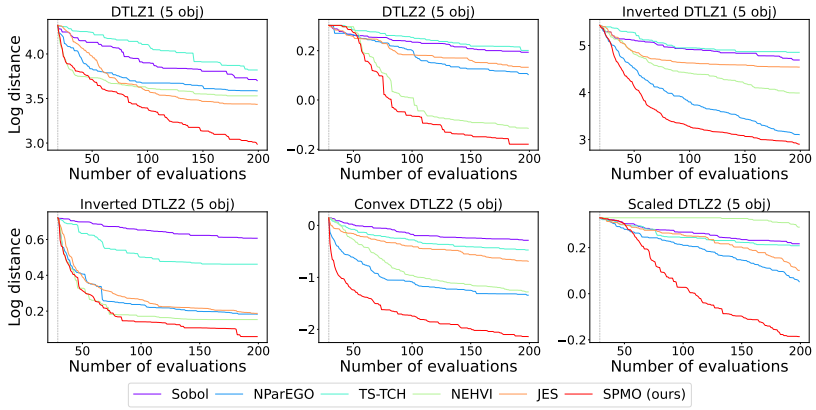
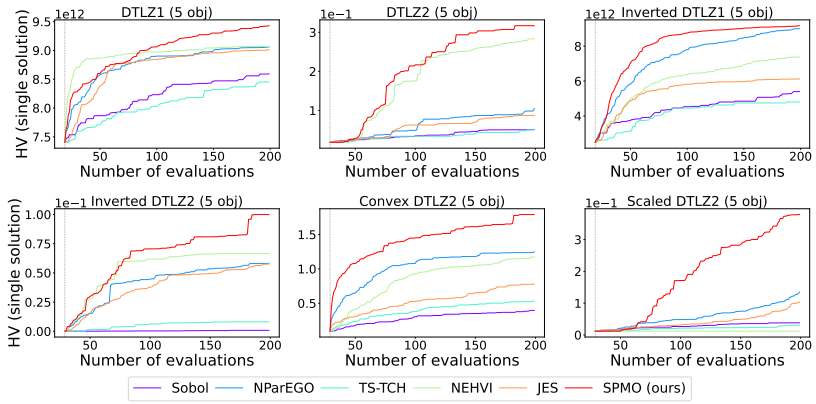


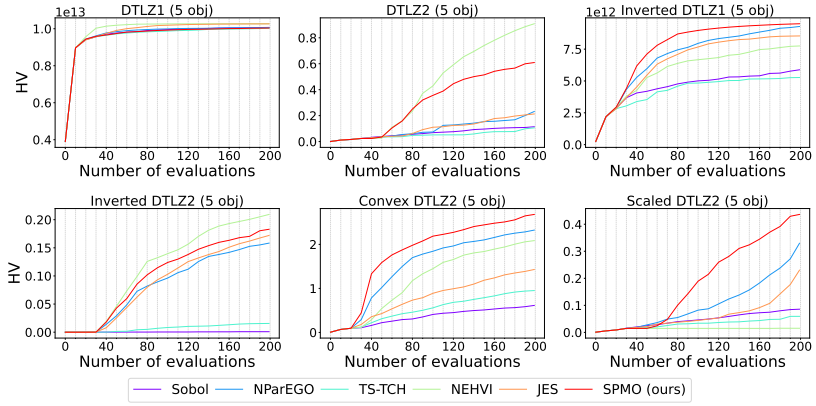
Figure 15: Violin plots of the HV of all evaluated solutions obtained by the six methods on the noisy problems with five objectives. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.



(a) Trajectories of the distance metric.



(b) Trajectories of the HV of the best solution.



(c) Trajectories of the HV of all evaluated solutions.

Figure 16: Trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO and the peer methods. Each coloured line represents the mean metric value on 30 independent runs. Note that we report the HV of all evaluated solutions every 10 evaluations due to the high computational complexity of HV calculation.

F.3 Batch Setting

We compare the proposed SPMO with the peer methods in the batch setting where the batch size q is set to 5 (a commonly used value (Lin et al., 2022)). Tables 14, 15 and 16 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value) and the HV of all evaluated solutions obtained by the SPMO and the peer methods, respectively. Figures 17, 18 and 19 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 14, 15 and 16, respectively. Figure 20 shows trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO and the peer methods.

Table 14: Results of the distance-based metric (log distance) obtained by the SPMO and the five peer methods with a batch size $q = 5$ on the problems with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	3.7e+0 (3.0e–1)$^+$	2.4e–1 (4.6e–2)$^+$	4.8e+0 (3.1e–1)$^+$	6.0e–1 (5.7e–2)$^+$	-3.1e–1 (2.5e–1)$^+$	2.3e–1 (5.1e–2)$^+$	**6/ 0/ 0**
ParEGO	3.4e+0 (1.9e–1)$^+$	6.7e–2 (8.1e–2)$^+$	3.2e+0 (5.4e–1)$^+$	2.5e–1 (1.3e–2)$^+$	-1.7e+0 (2.0e–1)$^+$	1.3e–1 (8.9e–2)$^+$	**6/ 0/ 0**
TS-TCH	3.9e+0 (2.2e–1)$^+$	2.0e–1 (4.1e–2)$^+$	4.8e+0 (3.2e–1)$^+$	4.6e–1 (1.2e–2)$^+$	-7.1e–1 (2.1e–1)$^+$	2.6e–1 (3.8e–2)$^+$	**6/ 0/ 0**
EHVI	3.5e+0 (9.6e–2)$^+$	9.3e–3 (3.5e–3)$^+$	4.0e+0 (4.4e–1)$^+$	2.3e–1 (5.5e–3)$^+$	-1.4e+0 (2.1e–1)$^+$	3.1e–1 (6.0e–2)$^+$	**6/ 0/ 0**
JES	3.4e+0 (1.3e–1)$^+$	1.1e–1 (8.0e–2)$^+$	4.5e+0 (1.7e–1)$^+$	2.6e–1 (2.0e–2)$^+$	-1.0e+0 (4.5e–1)$^+$	8.8e–2 (9.3e–2)$^+$	**6/ 0/ 0**
SPMO	**3.1e+0 (3.0e–1)**	**9.0e–4 (8.3e–4)**	**2.9e+0 (4.9e–1)**	**2.1e–1 (5.0e–5)**	**-2.1e+0 (7.7e–3)**	**1.7e–4 (1.2e–4)**	

Table 15: The HV of the best solution (in terms of its HV value) obtained by SPMO and the five peer methods with a batch size $q = 5$ on the problems with 5 objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	8.7e+12 (3.7e+11)$^+$	3.5e–2 (1.5e–2)$^+$	5.1e+12 (1.1e+12)$^+$	8.7e–4 (1.5e–3)$^+$	3.8e–1 (1.7e–1)$^+$	3.8e–2 (2.1e–2)$^+$	**6/ 0/ 0**
ParEGO	9.1e+12 (1.8e+11)$^+$	1.3e–1 (5.6e–2)$^+$	8.8e+12 (7.0e+11)$^+$	4.0e–2 (2.9e–3)$^+$	1.1e+0 (6.2e–2)$^+$	8.7e–2 (5.3e–2)$^+$	**6/ 0/ 0**
TS-TCH	8.4e+12 (3.8e+11)$^+$	3.2e–2 (1.3e–2)$^+$	4.9e+12 (1.2e+12)$^+$	7.1e–3 (1.1e–3)$^+$	6.5e–1 (1.4e–1)$^+$	2.8e–2 (1.5e–2)$^+$	**6/ 0/ 0**
EHVI	9.0e+12 (9.5e+10)$^+$	**1.9e–1 (7.2e–3)**$^\sim$	7.5e+12 (9.7e+11)$^+$	4.5e–2 (1.4e–3)$^+$	1.0e+0 (9.7e–2)$^+$	1.5e–2 (1.4e–2)$^+$	**5/ 1/ 0**
JES	9.0e+12 (1.2e+11)$^+$	1.0e–1 (3.9e–2)$^+$	6.2e+12 (5.0e+11)$^+$	3.9e–2 (4.5e–3)$^+$	8.5e–1 (2.4e–1)$^+$	1.1e–1 (5.5e–2)$^+$	**6/ 0/ 0**
SPMO	**9.3e+12 (2.8e+11)**	1.9e–1 (1.4e–2)	**9.2e+12 (4.8e+11)**	**4.9e–2 (1.2e–5)**	**1.3e+0 (5.0e–3)**	**1.6e–1 (1.6e–2)**	

Table 16: The HV of all the solutions obtained by the six methods with a batch size $q = 5$ on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO, respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
Sobol	1.0e+13 (3.7e+10)$^\sim$	8.3e–2 (2.6e–2)$^+$	5.6e+12 (7.6e+11)$^+$	1.1e–3 (1.8e–3)$^+$	5.8e–1 (1.7e–1)$^+$	7.9e–2 (2.2e–2)$^+$	**5/ 1/ 0**
ParEGO	1.0e+13 (1.6e+11)$^\sim$	2.4e–1 (1.3e–1)$^+$	7.9e+12 (7.4e+11)$^+$	1.3e–1 (8.6e–3)$^-$	1.4e+0 (1.0e–1)$^+$	2.2e–1 (1.4e–1)$^+$	**4/ 1/ 1**
TS-TCH	1.0e+13 (1.2e+11)$^\sim$	6.7e–2 (4.4e–2)$^+$	5.5e+12 (8.9e+11)$^+$	2.0e–2 (4.8e–3)$^+$	1.0e+0 (1.9e–1)$^+$	6.5e–2 (4.4e–2)$^+$	**5/ 1/ 0**
EHVI	**1.0e+13 (4.2e+9)**$^-$	**6.5e–1 (9.0e–2)**$^-$	7.0e+12 (6.8e+11)$^+$	**1.7e–1 (5.7e–3)**$^-$	1.2e+0 (2.4e–1)$^+$	4.1e–2 (3.5e–2)$^+$	**3/ 0/ 3**
JES	1.0e+13 (1.0e+10)$^-$	3.6e–1 (1.9e–1)$^\sim$	7.4e+12 (7.2e+11)$^+$	1.3e–1 (9.3e–3)$^-$	1.4e+0 (8.8e–2)$^+$	2.2e–1 (1.9e–1)$^+$	**3/ 1/ 2**
SPMO	1.0e+13 (1.2e+11)	4.0e–1 (1.1e–1)	**8.9e+12 (7.3e+11)**	1.2e–1 (7.6e–3)	**1.5e+0 (3.1e–2)**	**3.3e–1 (9.7e–2)**	

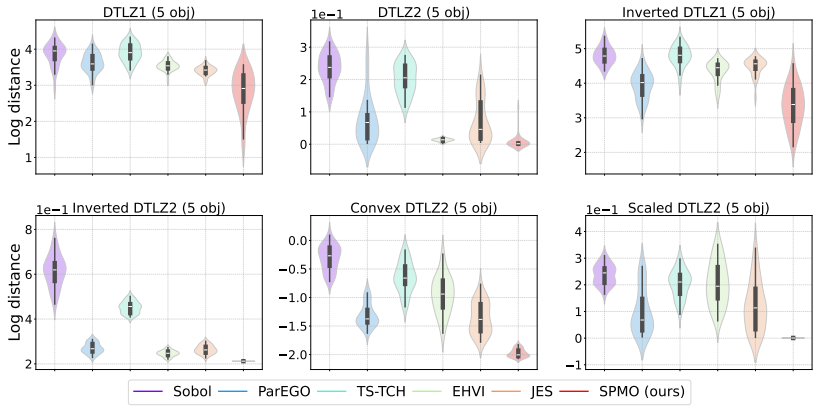


Figure 17: Violin plots of the distance-based metric (log distance) obtained by the proposed SPMO and the peer methods on the problems with five objectives. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

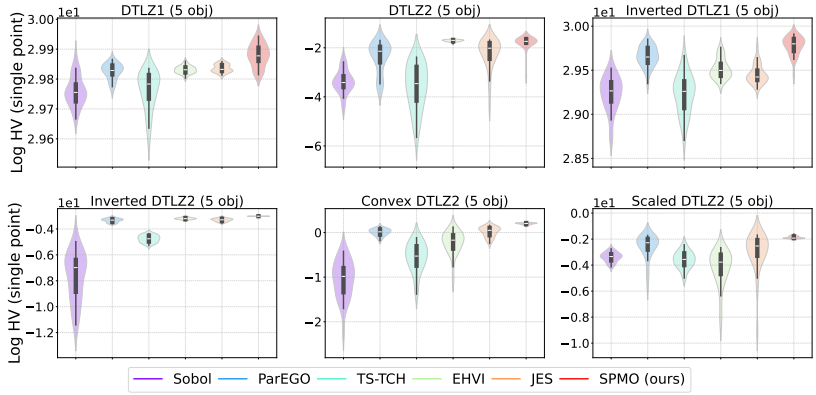


Figure 18: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the six methods with a batch size $q = 5$ on the problems with five objectives. Each violin represents the distribution of the HV of the best solution obtained by a method over 30 independent runs.

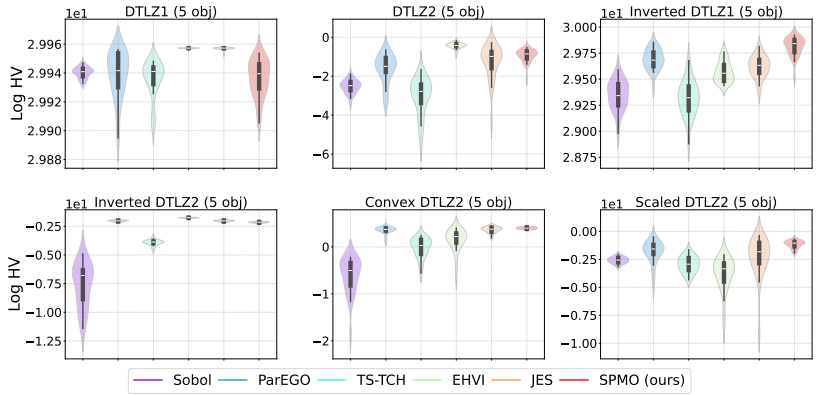
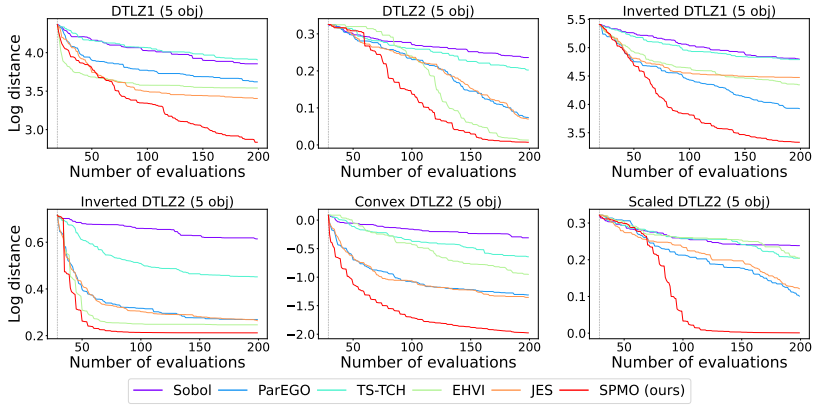
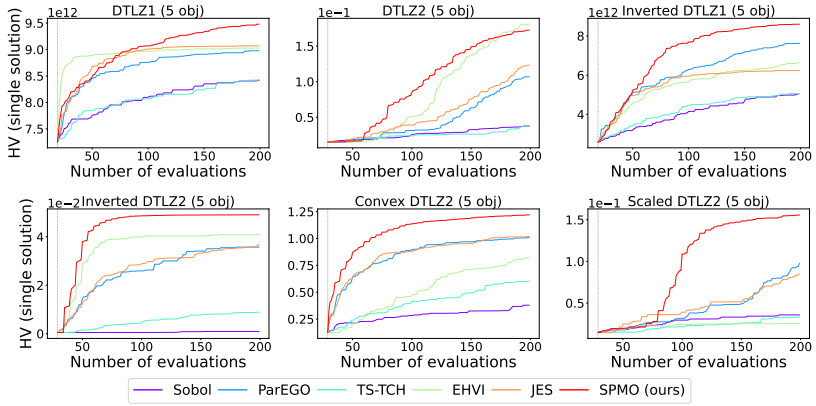


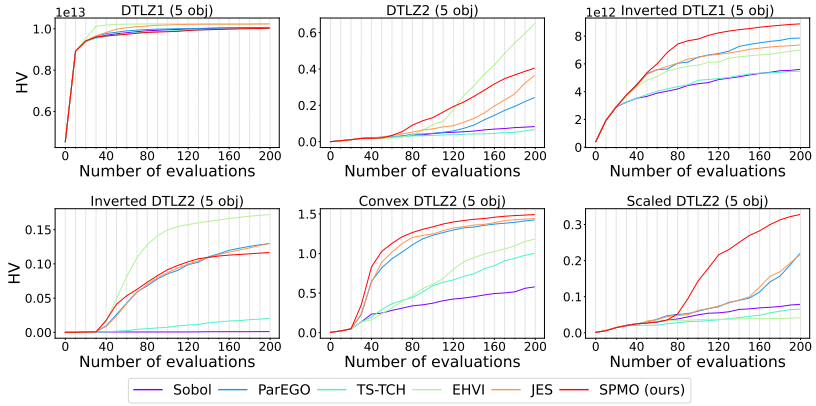
Figure 19: Violin plots of the HV of all evaluated solutions obtained by the six methods with a batch size $q = 5$ on the problems with five objectives. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.



(a) Trajectories of the distance metric.



(b) Trajectories of the HV of the best solution.



(c) Trajectories of the HV of all evaluated solutions.

Figure 20: Trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO and the peer methods with a batch size $q = 5$. Each coloured line represents the mean metric value on 30 independent runs. Note that we report the HV of all evaluated solutions every 10 evaluations due to the high computational complexity of HV calculation.

F.4 Sensitivity analysis

F.4.1 Utopian Points Based on the True Ideal Point.

In this section, we conduct a sensitivity analysis to assess the effect of different utopian points. We consider three different settings based on the problem's true ideal point. The first one is slightly better than the ideal point, i.e., with a difference of 0.01, the second is fairly better than the ideal point (i.e. 0.1), and the last one is significantly better than the ideal point (i.e. 1.0). Tables 17, 18 and 19 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value), and the HV of all evaluated solutions obtained by the SPMO with four different utopian points, respectively. Figures 21, 22 and 23 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 17, 18 and 19, respectively. Figure 24 shows trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO with four different utopian points.

Table 17: Results of the distance-based metric (log distance) obtained by the SPMO with four different utopian points on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
SPMO_0.01	3.1e+0 (4.9e–1)$^{\sim}$	6.2e–4 (1.4e–3)$^{-}$	**2.8e+0 (4.2e–1)**$^{\sim}$	2.1e–1 (3.6e–5)$^{\sim}$	-2.1e+0 (1.6e–2)$^{\sim}$	6.4e–5 (3.9e–5)$^{-}$	**0/ 4/ 2**
SPMO_0.1	2.9e+0 (3.6e–1)$^{-}$	**3.0e–4 (2.3e–4)**$^{-}$	3.0e+0 (5.3e–1)$^{\sim}$	2.1e–1 (4.6e–5)$^{\sim}$	-2.1e+0 (2.6e–2)$^{\sim}$	**5.3e–5 (2.4e–5)**$^{-}$	**0/ 3/ 3**
SPMO_1.0	**2.8e+0 (6.1e–1)**$^{\sim}$	3.3e–4 (2.0e–4)$^{-}$	2.9e+0 (5.0e–1)$^{\sim}$	**2.1e–1 (3.4e–5)**$^{\sim}$	-2.1e+0 (1.6e–2)$^{\sim}$	5.7e–5 (2.4e–5)$^{-}$	**0/ 4/ 2**
SPMO	3.1e+0 (3.0e–1)	9.0e–4 (8.3e–4)	2.9e+0 (4.9e–1)	2.1e–1 (5.0e–5)	**-2.1e+0 (7.7e–3)**	1.7e–4 (1.2e–4)	

Table 18: The HV of the best solution (in terms of its HV value) obtained by the proposed SPMO with four different utopian points on the problems with five objectives on 30 independent runs. The method exhibiting the best mean is highlighted in bold. The symbols "+", "∼", and "−" denote that a method is statistically worse than, equivalent to, or better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
SPMO_0.01	9.4e+12 (3.7e+11)$^{\sim}$	**1.9e–1 (1.0e–2)**$^{\sim}$	**9.3e+12 (3.8e+11)**$^{\sim}$	4.9e–2 (9.0e–6)$^{\sim}$	**1.3e+0 (7.8e–3)**$^{\sim}$	1.6e–1 (1.4e–2)$^{\sim}$	**0/ 6/ 0**
SPMO_0.1	9.5e+12 (2.7e+11)$^{-}$	1.9e–1 (1.6e–2)$^{\sim}$	9.0e+12 (5.6e+11)$^{\sim}$	4.9e–2 (1.1e–5)$^{\sim}$	1.2e+0 (1.1e–2)$^{\sim}$	1.6e–1 (1.0e–2)$^{\sim}$	**0/ 5/ 1**
SPMO_1.0	**9.5e+12 (3.8e+11)**$^{-}$	1.9e–1 (1.3e–2)$^{\sim}$	9.1e+12 (4.2e+11)$^{\sim}$	**4.9e–2 (8.5e–6)**$^{\sim}$	1.3e+0 (7.8e–3)$^{\sim}$	1.6e–1 (1.4e–2)$^{\sim}$	**0/ 5/ 1**
SPMO	9.3e+12 (2.8e+11)	1.9e–1 (1.4e–2)	9.2e+12 (4.8e+11)	4.9e–2 (1.2e–5)	1.3e+0 (5.0e–3)	**1.6e–1 (1.6e–2)**	

Table 19: The HV of all the solutions obtained by the SPMO with four different utopian points on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	Sum up +/∼/−
SPMO_0.01	3.1e+13 (1.7e+13)$^{\sim}$	2.7e+0 (1.2e+0)$^{-}$	3.5e+13 (3.7e+13)$^{\sim}$	6.5e–1 (7.0e–1)$^{\sim}$	2.5e+1 (1.8e+1)$^{\sim}$	**2.2e+0 (1.5e+0)**$^{\sim}$	**0/ 5/ 1**
SPMO_0.1	3.5e+13 (1.5e+13)$^{\sim}$	2.7e+0 (1.2e+0)$^{-}$	2.6e+13 (1.7e+13)$^{\sim}$	6.1e–1 (4.5e–1)$^{\sim}$	2.7e+1 (1.7e+1)$^{\sim}$	1.8e+0 (1.0e+0)$^{\sim}$	**0/ 5/ 1**
SPMO_1.0	3.3e+13 (1.4e+13)$^{\sim}$	**3.0e+0 (1.6e+0)**$^{-}$	3.2e+13 (1.9e+13)$^{\sim}$	**7.1e–1 (6.3e–1)**$^{\sim}$	**2.9e+1 (2.2e+1)**$^{\sim}$	2.1e+0 (1.5e+0)$^{\sim}$	**0/ 5/ 1**
SPMO	**3.7e+13 (2.2e+13)**	1.5e+0 (7.0e–1)	**3.7e+13 (3.5e+13)**	5.9e–1 (3.3e–1)	2.6e+1 (1.7e+1)	1.9e+0 (1.6e+0)	

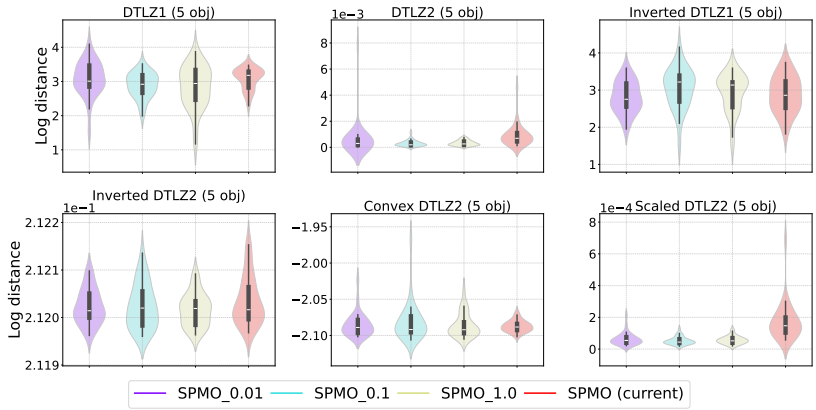


Figure 21: Violin plots of the distance-based metric (log distance) obtained by the four methods on the problems with five objectives. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

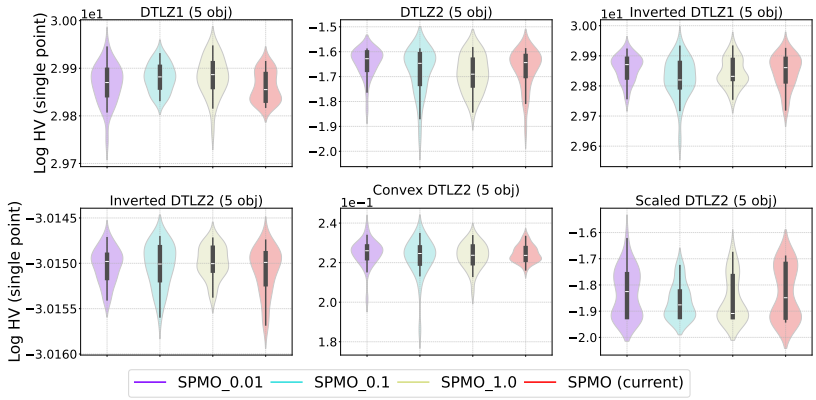


Figure 22: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the four methods on the problems with five objectives. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.

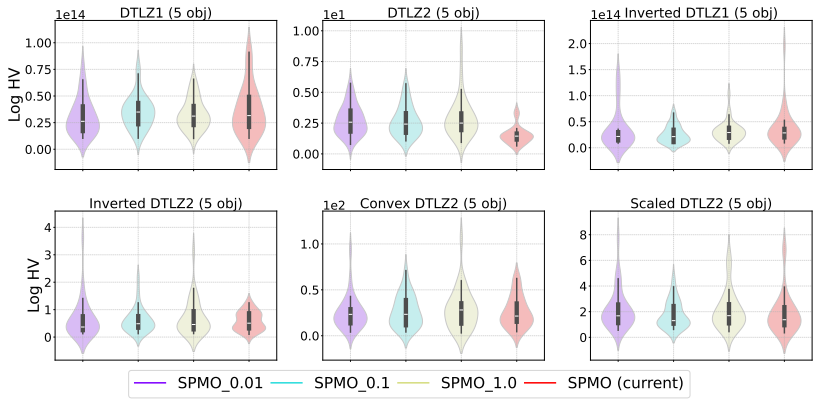
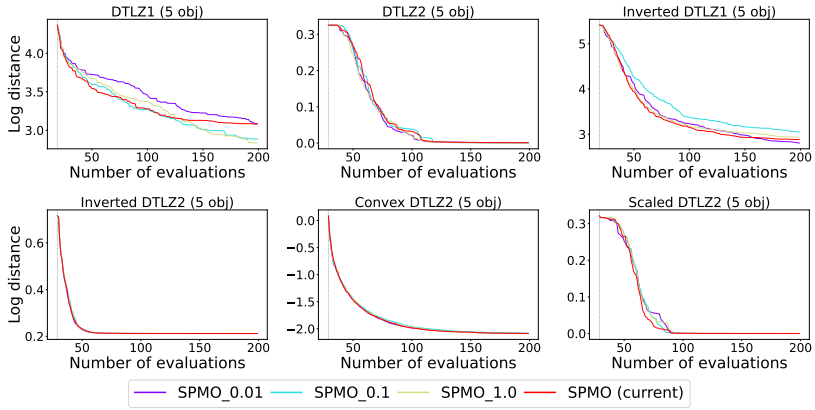
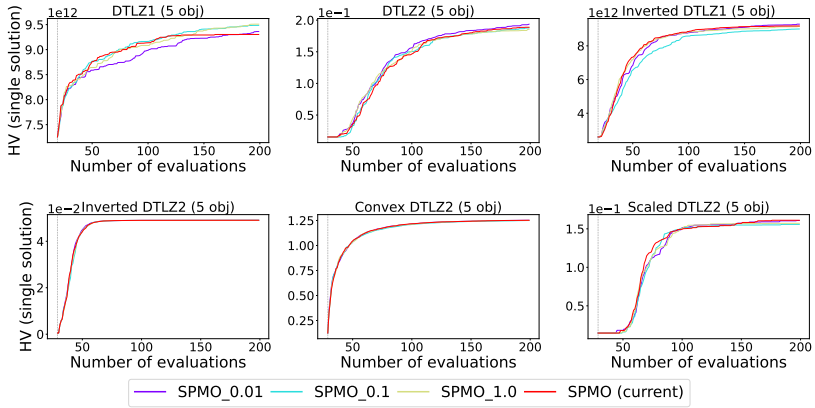


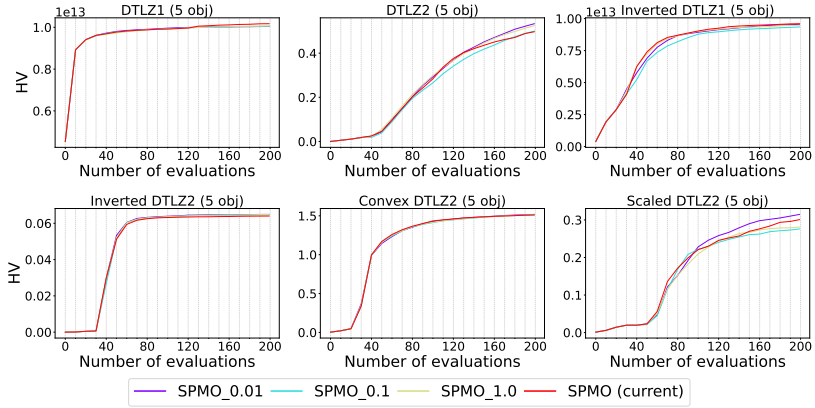
Figure 23: Violin plots of the HV of all evaluated solutions obtained by the four methods on the problems with five objectives. Each violin represents the distribution of the HV of all evaluated solutions obtained by a method over 30 independent runs.



(a) Trajectories of the distance metric.



(b) Trajectories of the HV of the best solution.



(c) Trajectories of the HV of all evaluated solutions.

Figure 24: Trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO with four different utopian points. Each coloured line represents the mean metric value on 30 independent runs. Note that we report the HV of all evaluated solutions every 10 evaluations due to the high computational complexity of HV calculation.

F.4.2 Utopian Points Based on Observed Solutions based on the observed solutions.

Previously, we investigated SPMO using different utopian points based on the problem's true ideal point. However, there may exist situations in the real world where one may have very little knowledge about the optimisation problem. To further examine a more practical setting, we consider dynamically setting the utopian point based only on the observed solutions during optimisation. We consider four settings, where the observed objective values are first normalised and the utopian point is then set to $(-0.1, \ldots, -0.1)$, $(-0.2, \ldots, -0.2)$, $(-0.5, \ldots, -0.5)$, and $(-1.0, \ldots, -1.0)$, respectively. These variants are denoted as SPMO$_{dynamic_0.1}$, SPMO$_{dynamic_0.2}$, SPMO$_{dynamic_0.5}$, and SPMO$_{dynamic_1.0}$, respectively. Tables 20, 21 and 22 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value), and the HV of all evaluated solutions obtained by the SPMO with four different utopian points, respectively. Figures 25, 26 and 27 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 20, 21 and 22, respectively. Figure 28 shows trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO with five different utopian points.

Table 20: Results of the distance-based metric (log distance) obtained by the SPMO with five different utopian points on the benchmark problems with five objectives and the car side impact problem on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{dynamic_0.1}$	3.0e+0 (4.7e−1)$^{\sim}$	**8.8e−5 (5.3e−5)**$^{-}$	3.0e+0 (5.3e−1)$^{\sim}$	2.4e−1 (1.2e−2)$^{+}$	-2.0e+0 (3.5e−2)$^{+}$	9.6e−5 (4.6e−5)$^{-}$	-3.4e−1 (6.3e−3)$^{+}$	**3/ 2/ 2**
SPMO$_{dynamic_0.2}$	3.1e+0 (5.1e−1)$^{\sim}$	1.2e−4 (8.7e−5)$^{-}$	2.9e+0 (6.2e−1)$^{\sim}$	2.3e−1 (7.0e−3)$^{+}$	-2.0e+0 (3.0e−2)$^{+}$	1.2e−4 (5.8e−5)$^{\sim}$	-3.4e−1 (5.6e−4)$^{+}$	**3/ 3/ 1**
SPMO$_{dynamic_0.5}$	3.0e+0 (4.5e−1)$^{\sim}$	1.8e−4 (1.6e−4)$^{-}$	3.0e+0 (5.6e−1)$^{\sim}$	2.2e−1 (3.6e−3)$^{+}$	-2.0e+0 (3.4e−2)$^{+}$	1.6e−4 (1.0e−4)$^{\sim}$	-3.4e−1 (2.5e−4)$^{+}$	**3/ 3/ 1**
SPMO$_{dynamic_1.0}$	**2.9e+0 (5.0e−1)**$^{\sim}$	3.0e−4 (2.3e−4)$^{-}$	**2.8e+0 (4.9e−1)**$^{\sim}$	2.1e−1 (6.2e−5)$^{\sim}$	-2.1e+0 (2.6e−2)$^{\sim}$	**6.0e−5 (3.5e−5)**$^{-}$	**-3.4e−1 (4.6e−7)**$^{\sim}$	**0/ 5/ 2**
SPMO	3.1e+0 (3.0e−1)	9.0e−4 (8.3e−4)	2.9e+0 (4.9e−1)	**2.1e−1 (5.0e−5)**	**-2.1e+0 (7.8e−3)**	1.7e−4 (1.2e−4)	-3.4e−1 (1.6e−6)	

Table 21: The HV of the best solution (in terms of its HV value) obtained by the proposed SPMO with five different utopian points on the benchmark problems with five objectives and the car side impact problem on 30 independent runs. The method exhibiting the best mean is highlighted in bold. The symbols "+", "∼", and "−" denote that a method is statistically worse than, equivalent to, or better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{dynamic_0.1}$	9.4e+12 (3.4e+11)$^{\sim}$	1.8e−1 (1.5e−2)$^{+}$	9.0e+12 (5.5e+11)$^{\sim}$	4.2e−2 (2.8e−3)$^{+}$	1.3e+0 (7.8e−3)$^{\sim}$	**1.8e−1 (1.4e−2)**$^{-}$	3.6e−1 (6.1e−3)$^{\sim}$	**2/ 4/ 1**
SPMO$_{dynamic_0.2}$	9.3e+12 (3.5e+11)$^{\sim}$	1.8e−1 (1.2e−2)$^{+}$	9.1e+12 (5.7e+11)$^{\sim}$	4.4e−2 (1.7e−3)$^{+}$	1.2e+0 (6.9e−3)$^{+}$	1.8e−1 (1.4e−2)$^{-}$	3.7e−1 (5.8e−4)$^{-}$	**3/ 2/ 2**
SPMO$_{dynamic_0.5}$	9.4e+12 (3.6e+11)$^{\sim}$	1.8e−1 (1.5e−2)$^{+}$	9.1e+12 (6.2e+11)$^{\sim}$	4.7e−2 (8.7e−4)$^{+}$	1.2e+0 (7.4e−3)$^{+}$	1.8e−1 (1.3e−2)$^{-}$	**3.7e−1 (1.4e−4)**$^{-}$	**3/ 2/ 2**
SPMO$_{dynamic_1.0}$	**9.5e+12 (3.5e+11)**$^{-}$	1.9e−1 (1.6e−2)$^{\sim}$	**9.3e+12 (4.0e+11)**$^{\sim}$	4.9e−2 (1.5e−5)$^{\sim}$	1.2e+0 (1.0e−2)$^{+}$	1.6e−1 (1.2e−2)$^{\sim}$	3.6e−1 (2.6e−3)$^{\sim}$	**1/ 5/ 1**
SPMO	9.3e+12 (2.8e+11)	**1.9e−1 (1.4e−2)**	9.2e+12 (4.8e+11)	**4.9e−2 (1.2e−5)**	**1.3e+0 (5.0e−3)**	1.6e−1 (1.6e−2)	3.6e−1 (3.2e−3)	

Table 22: The HV of all the solutions obtained by the SPMO with five different utopian points on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO (current), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{dynamic_0.1}$	1.0e+13 (1.3e+11)$^{+}$	4.3e−1 (9.0e−2)$^{+}$	9.5e+12 (4.5e+11)$^{\sim}$	5.3e−2 (5.4e−3)$^{+}$	1.5e+0 (3.4e−2)$^{\sim}$	4.3e−1 (1.0e−1)$^{-}$	4.9e−1 (3.5e−2)$^{+}$	**4/ 2/ 1**
SPMO$_{dynamic_0.2}$	1.0e+13 (1.2e+11)$^{+}$	4.3e−1 (1.0e−1)$^{+}$	9.6e+12 (5.1e+11)$^{\sim}$	5.7e−2 (3.3e−3)$^{+}$	1.5e+0 (3.5e−2)$^{\sim}$	**4.8e−1 (9.4e−2)**$^{-}$	5.3e−1 (3.9e−2)$^{\sim}$	**3/ 3/ 1**
SPMO$_{dynamic_0.5}$	1.0e+13 (1.0e+11)$^{+}$	4.6e−1 (6.7e−2)$^{+}$	9.5e+12 (5.2e+11)$^{\sim}$	6.4e−2 (4.2e−3)$^{\sim}$	1.5e+0 (2.5e−2)$^{\sim}$	4.4e−1 (8.3e−2)$^{-}$	**5.8e−1 (2.6e−2)**$^{-}$	**2/ 3/ 2**
SPMO$_{dynamic_1.0}$	1.0e+13 (1.0e+11)$^{+}$	**5.0e−1 (8.2e−2)**$^{\sim}$	9.6e+12 (2.9e+11)$^{\sim}$	**6.4e−2 (2.5e−3)**$^{\sim}$	**1.5e+0 (3.0e−2)**$^{\sim}$	3.0e−1 (8.8e−2)$^{\sim}$	5.3e−1 (2.5e−2)$^{\sim}$	**1/ 6/ 0**
SPMO	**1.0e+13 (8.1e+10)**	5.0e−1 (8.2e−2)	**9.6e+12 (2.9e+11)**	6.4e−2 (2.8e−3)	1.5e+0 (3.2e−2)	3.0e−1 (1.1e−1)	5.3e−1 (3.1e−2)	

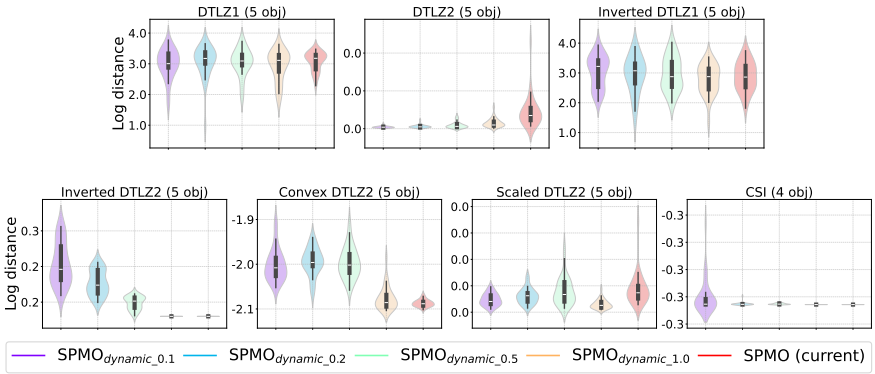


Figure 25: Violin plots of the distance-based metric (log distance) obtained by the five methods on the benchmark problems with five objectives and the car side impact problem. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

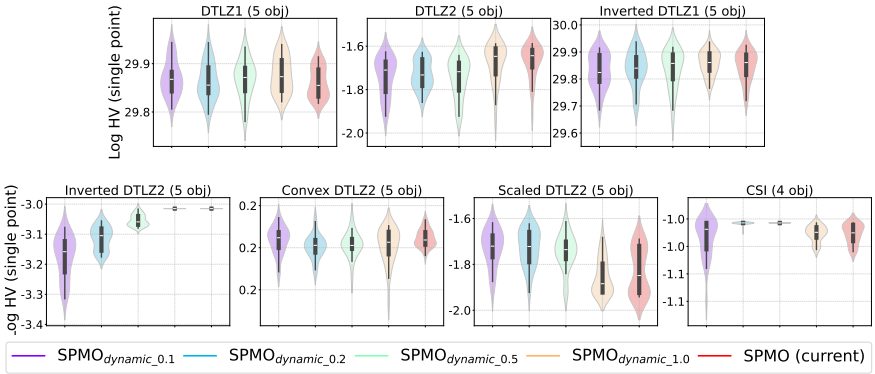


Figure 26: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the five methods on the benchmark problems with five objectives and the car side impact problem. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.

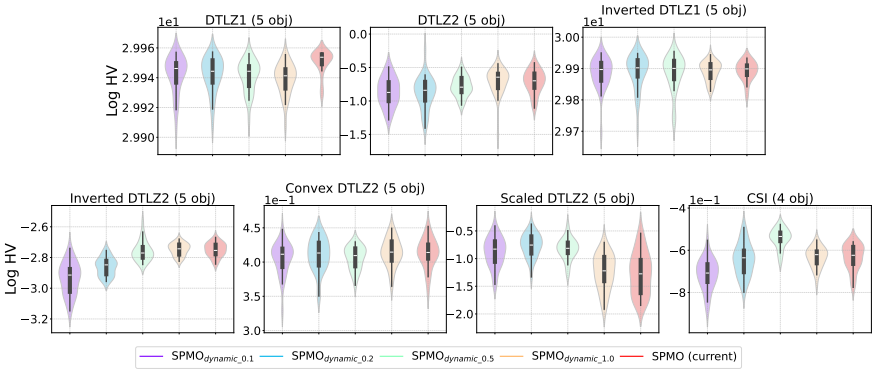
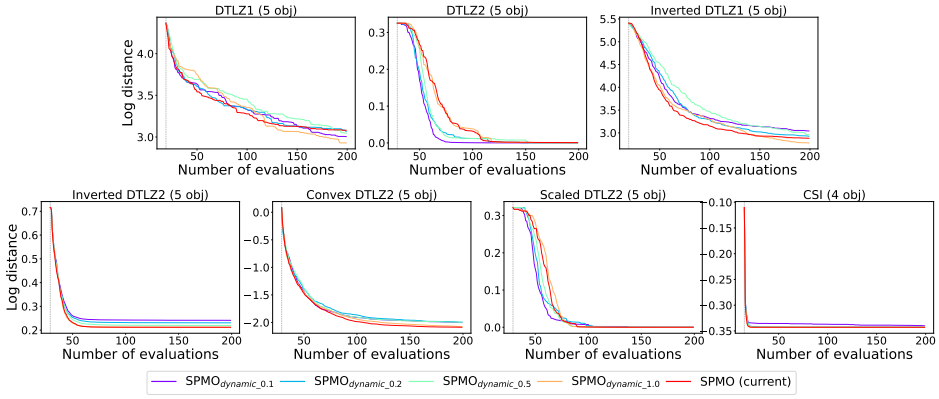
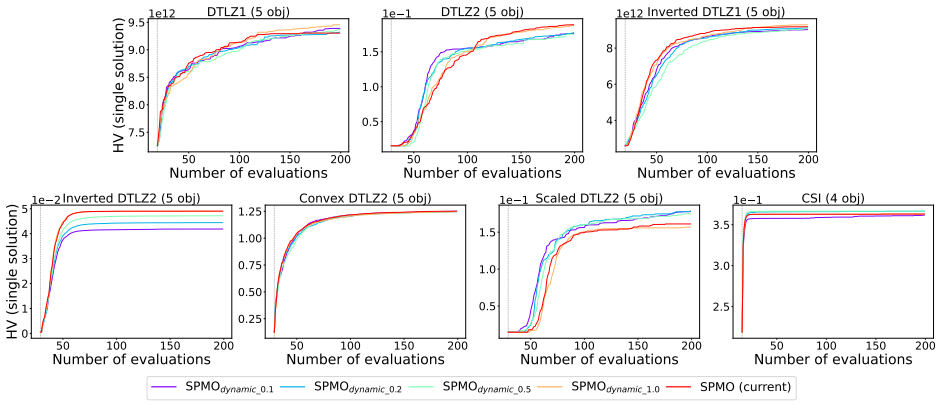


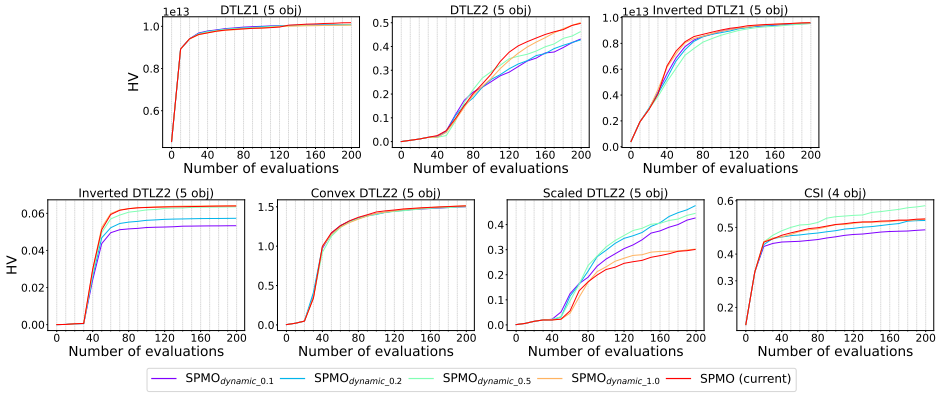
Figure 27: Violin plots of the HV of all evaluated solutions obtained by the five methods on the benchmark problems with five objectives and the car side impact problem. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.



(a) Trajectories of the distance metric.



(b) Trajectories of the HV of the best solution.



(c) Trajectories of the HV of all evaluated solutions.

Figure 28: Trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO with five different utopian points. Each coloured line represents the mean metric value on 30 independent runs. Note that we report the HV of all evaluated solutions every 10 evaluations due to the high computational complexity of HV calculation.

F.5 Comparison of Single-Point Metrics within SPMO

In the proposed SPMO framework, we employ a distance metric (i.e., the distance of a solution to the utopian point), denoted as SPMO$_{dist}$. However, different metrics can be adopted provided that they can reflect the quality of a solution in achieving a good trade-off between objectives. We now consider three other well-known metrics, weighted sum, Tchebycheff scalarisation, and single-point HV, denoted by SPMO$_{ws}$, SPMO$_{Tch}$, and SPMO$_{hv}$, respectively. The equal weights $(\frac{1}{m}, \ldots, \frac{1}{m})$ are used in SPMO$_{ws}$ and SPMO$_{Tch}$. The reference point of SPMO$_{hv}$ is set to 1.1 times the observed Pareto front, since using a fixed reference point (e.g., $(1.1, \ldots, 1.1) \in \mathbb{R}^m$ for DTLZ2) can lead to zero acquisition values when no candidate solution dominates the reference point. Here, we compare these four variants of SPMO, i.e., SPMO$_{dist}$, SPMO$_{Tch}$, SPMO$_{ws}$, and SPMO$_{hv}$. Tables 23, 24 and 25 show the distance-based metric (log distance), the HV of the best solution (in terms of its HV value) and the HV of all evaluated solutions obtained by the SPMO with four different single-point metrics, respectively. Figures 29, 30 and 31 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 23, 24 and 25, respectively. Figure 32 shows trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by each method.

Table 23: Results of the distance-based metric (log distance) obtained by the SPMO using four different single-point metrics on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO (i.e., SPMO$_{dist}$), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{Tch}$	3.5e+0 (3.2e−1)$^+$	**3.5e−5 (4.3e−5)**$^-$	2.9e+0 (6.5e−1)$^\sim$	3.1e−1 (3.3e−2)$^+$	-1.8e+0 (1.7e−1)$^+$	**2.9e−5 (1.9e−5)**$^-$	-2.8e−1 (3.5e−2)$^+$	**4/ 1/ 2**
SPMO$_{ws}$	3.6e+0 (1.3e−1)$^+$	2.1e−4 (1.5e−4)$^-$	3.1e+0 (5.4e−1)$^\sim$	2.2e−1 (2.7e−3)$^+$	-1.2e+0 (2.7e−1)$^+$	3.2e−4 (3.4e−4)$^+$	-3.1e−1 (5.5e−2)$^+$	**5/ 1/ 1**
SPMO$_{hv}$	3.5e+0 (1.5e−1)$^+$	6.2e−4 (6.3e−4)$^\sim$	3.9e+0 (5.5e−1)$^+$	2.2e−1 (3.5e−3)$^+$	-1.3e+0 (2.5e−1)$^+$	1.3e−1 (9.0e−2)$^+$	-3.3e−1 (7.6e−3)$^+$	**6/ 1/ 0**
SPMO	**3.1e+0 (3.0e−1)**	9.0e−4 (8.3e−4)	**2.9e+0 (4.9e−1)**	**2.1e−1 (5.0e−5)**	**-2.1e+0 (7.8e−3)**	1.7e−4 (1.2e−4)	**-3.4e−1 (1.6e−6)**	

Table 24: The HV of the best solution (in terms of its HV value) obtained by the proposed SPMO using four different single-point metrics on the problems with five objectives on 30 independent runs. The method exhibiting the best mean is highlighted in bold. The symbols "+", "∼", and "−" denote that a method is statistically worse than, equivalent to, or better than SPMO (i.e., SPMO$_{dist}$), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{Tch}$	9.1e+12 (3.0e+11)$^+$	1.6e−1 (1.9e−2)$^+$	9.1e+12 (5.7e+11)$^\sim$	2.7e−2 (5.9e−3)$^+$	1.2e+0 (4.2e−2)$^+$	1.6e−1 (2.2e−2)$^\sim$	2.9e−1 (3.0e−2)$^+$	**5/ 2/ 0**
SPMO$_{ws}$	9.1e+12 (1.3e+11)$^+$	1.7e−1 (1.4e−2)$^+$	9.0e+12 (6.5e+11)$^\sim$	4.8e−2 (6.7e−4)$^+$	9.7e−1 (1.2e−1)$^+$	**1.7e−1 (1.5e−2)**$^\sim$	3.5e−1 (3.0e−2)$^\sim$	**4/ 3/ 0**
SPMO$_{hv}$	9.1e+12 (1.6e+11)$^+$	1.8e−1 (1.7e−2)$^+$	7.6e+12 (1.1e+12)$^+$	4.8e−2 (8.3e−4)$^+$	1.0e+0 (9.9e−2)$^+$	8.1e−2 (5.3e−2)$^+$	3.6e−1 (6.1e−3)$^+$	**7/ 0/ 0**
SPMO	**9.3e+12 (2.8e+11)**	**1.9e−1 (1.4e−2)**	**9.2e+12 (4.8e+11)**	**4.9e−2 (1.2e−5)**	**1.3e+0 (5.0e−3)**	1.6e−1 (1.6e−2)	**3.6e−1 (3.2e−3)**	

Table 25: The HV of all the solutions obtained by the SPMO using four different single-point metrics on the problems with five objectives on 30 independent runs. The method with the best mean is highlighted in bold. The symbols "+", "∼", and "−" indicate that a method is statistically worse than, equivalent to, and better than SPMO (i.e., SPMO$_{dist}$), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{Tch}$	9.9e+12 (1.7e+11)$^+$	3.7e−1 (8.6e−2)$^+$	9.3e+12 (5.1e+11)$^+$	3.2e−2 (7.8e−3)$^+$	1.4e+0 (7.1e−2)$^+$	3.4e−1 (8.0e−2)$^\sim$	4.2e−1 (2.2e−2)$^+$	**6/ 1/ 0**
SPMO$_{ws}$	1.0e+13 (2.1e+11)$^+$	4.3e−1 (7.8e−2)$^+$	9.4e+12 (5.5e+11)$^\sim$	**7.2e−2 (4.9e−3)**$^-$	1.4e+0 (1.2e−1)$^+$	**4.1e−1 (1.0e−1)**$^-$	5.7e−1 (3.9e−2)$^-$	**3/ 1/ 3**
SPMO$_{hv}$	**1.0e+13 (7.0e+10)**$^\sim$	**5.3e−1 (1.2e−1)**$^\sim$	7.9e+12 (1.0e+12)$^+$	6.7e−2 (3.6e−3)$^-$	1.4e+0 (1.1e−1)$^+$	1.9e−1 (1.5e−1)$^+$	**7.4e−1 (1.7e−2)**$^-$	**3/ 2/ 2**
SPMO	1.0e+13 (8.1e+10)	5.0e−1 (8.2e−2)	**9.6e+12 (2.9e+11)**	6.4e−2 (2.8e−3)	**1.5e+0 (3.2e−2)**	3.0e−1 (1.1e−1)	5.3e−1 (3.1e−2)	

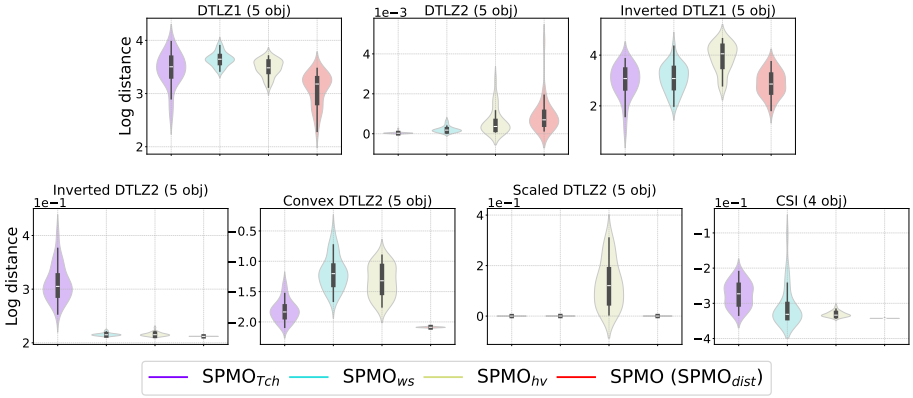


Figure 29: Violin plots of the distance-based metric (log distance) obtained by the SPMO using four different single-point metrics on the problems with five objectives. Each violin represents the distribution of the distance-based metric obtained by a method over 30 independent runs.

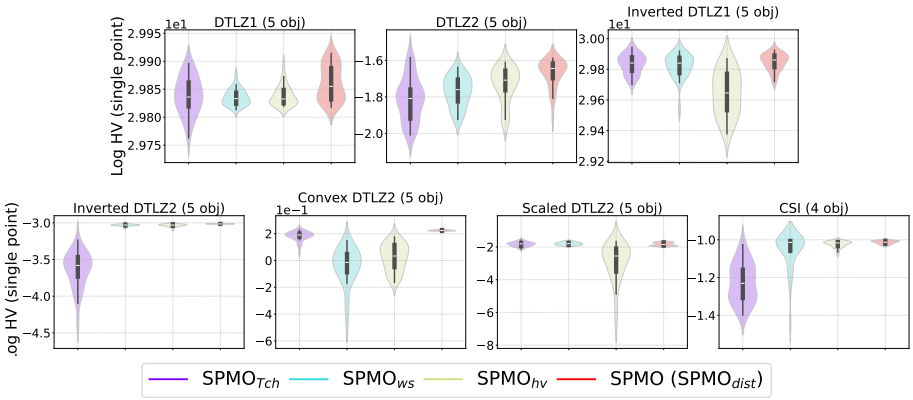


Figure 30: Violin plots of the HV of the best solution (in terms of its HV value) obtained by the SPMO using four different single-point metrics on the problems with 5 objectives. Each violin represents the distribution of the HV of the best solution obtained by a method over 30 independent runs.

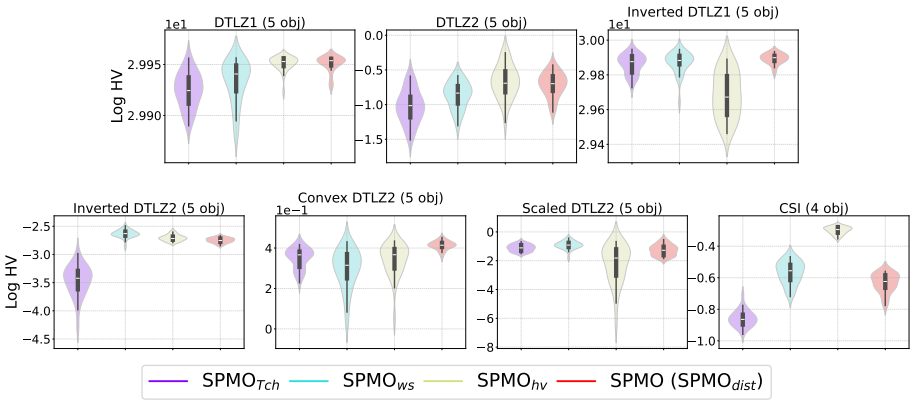
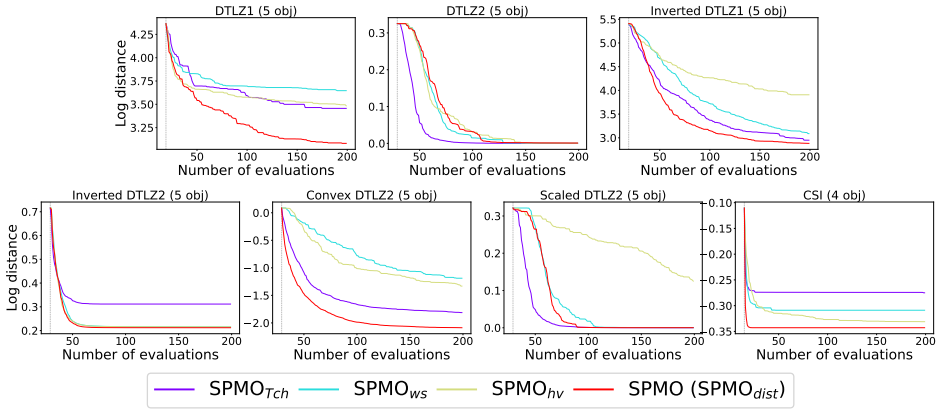
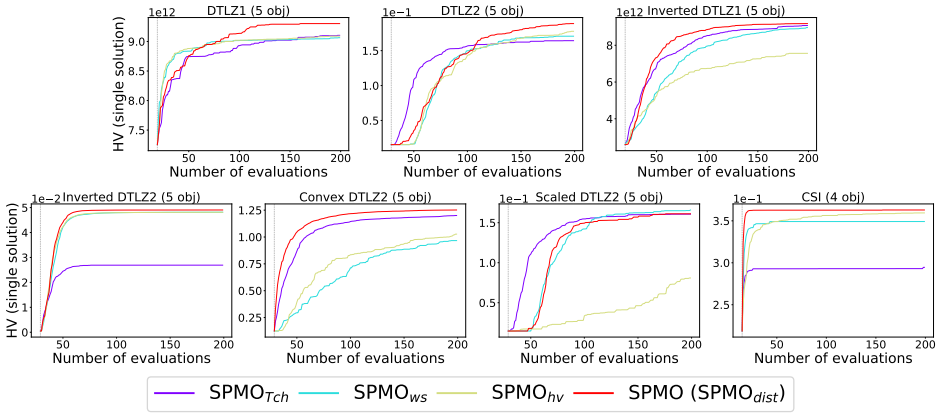


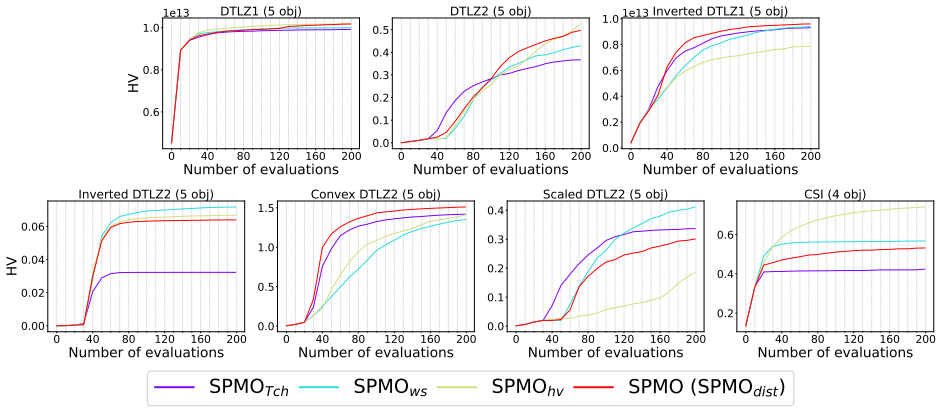
Figure 31: Violin plots of the HV of all evaluated solutions obtained by the SPMO using four different single-point metrics on the problems with five objectives. Each violin represents the distribution of maximum HV values obtained by a method over 30 independent runs.



(a) Trajectories of the distance metric.



(b) Trajectories of the HV of the best solution.



(c) Trajectories of the HV of all evaluated solutions.

Figure 32: Trajectories of the distance metric, the HV of the best solution, and the HV of all evaluated solutions obtained by the SPMO with four different single-point metrics. Each coloured line represents the mean metric value on 30 independent runs. Note that we report the HV of all evaluated solutions every 10 evaluations due to the high computational complexity of HV calculation.

F.6 Incorporating Preference Information into SPMO.

Previously, we focused on the preference-free setting, where no prior knowledge of the decision maker's preferences is available. However, SPMO can naturally incorporate preference information into its single-point metrics. We extend the four metrics introduced above, i.e., SPMO$_{dist}$, SPMO$_{ws}$, SPMO$_{Tch}$, and SPMO$_{hv}$, to the preference-based setting. Given a weight vector $\boldsymbol{w}$, a preference-based distance metric can be defined as $\|\boldsymbol{w} \odot (\boldsymbol{y} - \boldsymbol{z})\|_2$, where $\odot$ denotes element-wise multiplication. Weighted-sum and Tchebycheff scalarisation can incorporate preference information directly through a weight vector. Regarding the single-point HV metric, given a reference point $\boldsymbol{r}$, preference-based single-point HV can be defined as $\prod_{i=1}^{m} \max(w_i(y_i - r_i), 0)$.

For an m-objective problem, the weight vector is set to $\boldsymbol{w} = (0.1, 0.2, \ldots, 0.1m)$ and is normalised by its sum, i.e., $\boldsymbol{w} \leftarrow \boldsymbol{w}/\sum_{i=1}^{m} w_i$. Similar to the previous comparison, we use the preference-based distance metric and preference-based single-point HV as the performance metrics. Tables 26 and 27 show the weighted log distance and the preference-based single-point HV obtained by the SPMO with three different single-point metrics, respectively. Figures 33 and 34 present the violin plots, illustrating the distributions of the corresponding results reported in Tables 26 and 27, respectively. Figure 35 shows trajectories of the preference-based distance metric and the preference-based single-point HV of the best solution obtained by each method.

Table 26: Results in terms of the preference-based distance metric (log distance) obtained by SPMO with four different single-point metrics. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO (i.e., SPMO$_{dist}$), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{Tch}$	1.8e+0 (4.1e−1)$^+$	-2.7e+0 (3.4e−2)$^\sim$	1.3e+0 (6.4e−1)$^\sim$	-1.4e+0 (2.7e−2)$^+$	-3.7e+0 (1.7e−1)$^+$	-2.7e+0 (4.5e−2)$^+$	-2.2e+0 (1.0e−2)$^\sim$	**4/ 3/ 0**
SPMO$_{ws}$	1.5e+0 (5.2e−1)$^\sim$	-2.7e+0 (1.2e−1)$^+$	1.8e+0 (6.6e−1)$^+$	**-1.5e+0 (4.9e−3)**$^-$	-3.1e+0 (3.1e−1)$^+$	-2.7e+0 (1.6e−2)$^+$	-2.2e+0 (9.2e−3)$^\sim$	**4/ 2/ 1**
SPMO$_{hv}$	2.1e+0 (3.4e−1)$^+$	-2.3e+0 (3.1e−1)$^+$	2.2e+0 (6.0e−1)$^+$	-1.4e+0 (2.5e−2)$^+$	-2.9e+0 (3.3e−1)$^+$	-2.3e+0 (2.8e−1)$^+$	-2.2e+0 (1.6e−2)$^\sim$	**6/ 1/ 0**
SPMO$_{dist}$	**1.5e+0 (5.0e−1)**	**-2.7e+0 (4.0e−2)**	**1.2e+0 (4.1e−1)**	-1.4e+0 (4.9e−3)	**-4.0e+0 (3.3e−2)**	**-2.7e+0 (5.3e−5)**	**-2.2e+0 (1.5e−2)**	

Table 27: Results in terms of the preference-based single-point HV obtained by SPMO with four different single-point metrics. The method with the best mean is highlighted in bold. The symbols "+", "∼" and "−" indicate that the method is statistically worse than, equivalent to and better than our SPMO (i.e., SPMO$_{dist}$), respectively.

Method	DTLZ1 Mean (Std)	DTLZ2 Mean (Std)	Inverted DTLZ1 Mean (Std)	Inverted DTLZ2 Mean (Std)	Convex DTLZ2 Mean (Std)	Scaled DTLZ2 Mean (Std)	CSI Mean (Std)	Sum up +/∼/−
SPMO$_{Tch}$	1.4e+9 (3.7e+7)$^+$	**2.8e−5 (1.3e−6)**$^-$	1.4e+9 (1.1e+8)$^\sim$	1.2e−6 (5.2e−7)$^+$	1.8e−4 (1.1e−5)$^+$	**2.8e−5 (1.7e−6)**$^-$	5.5e−4 (8.5e−5)$^+$	**4/ 1/ 2**
SPMO$_{ws}$	1.4e+9 (1.9e+7)$^+$	2.3e−5 (7.8e−6)$^\sim$	1.3e+9 (1.5e+8)$^+$	4.0e−6 (4.0e−7)$^+$	1.3e−4 (2.8e−5)$^+$	2.6e−5 (3.2e−6)$^\sim$	5.5e−4 (8.0e−5)$^+$	**5/ 2/ 0**
SPMO$_{hv}$	1.4e+9 (2.0e+7)$^+$	2.8e−5 (2.3e−6)$^-$	1.2e+9 (1.8e+8)$^+$	**7.5e−6 (1.9e−7)**$^-$	1.6e−4 (1.6e−5)$^+$	2.7e−5 (2.9e−6)$^\sim$	8.2e−4 (1.6e−5)$^\sim$	**3/ 2/ 2**
SPMO$_{dist}$	**1.5e+9 (5.3e+7)**	2.4e−5 (4.8e−6)	**1.5e+9 (6.4e+7)**	5.7e−6 (1.8e−7)	**2.0e−4 (1.5e−6)**	2.6e−5 (2.7e−6)	**8.2e−4 (1.6e−5)**	

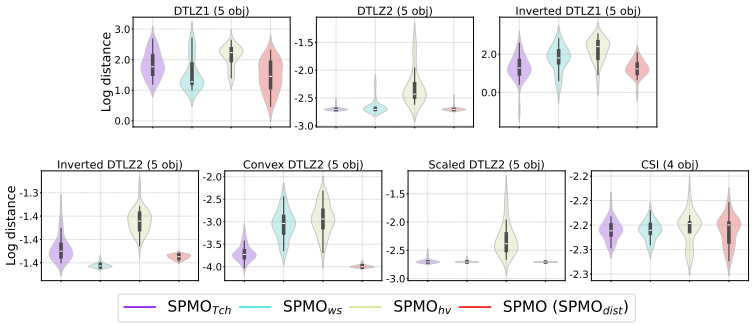


Figure 33: Violin plots of preference-based distance metric (log distance) obtained by the SPMO using four different single-point metrics on the problems with five objectives. Each violin represents the distribution of preference-based distance obtained by a method over 30 independent runs.

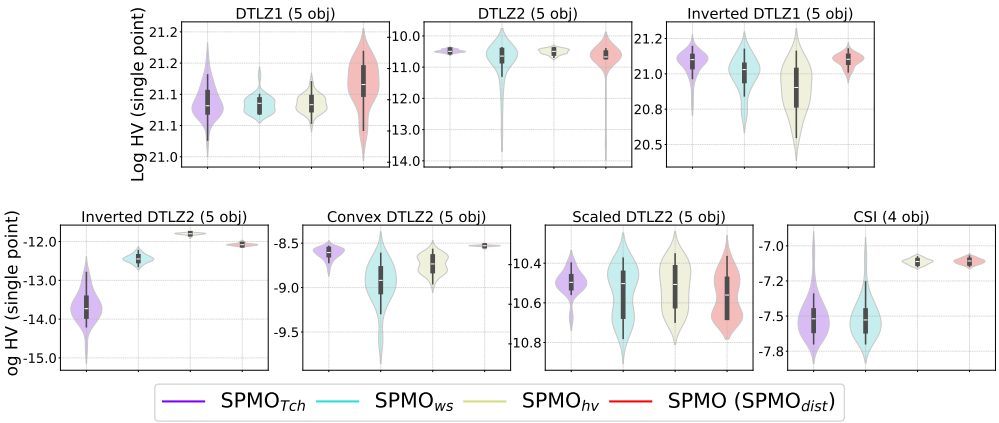
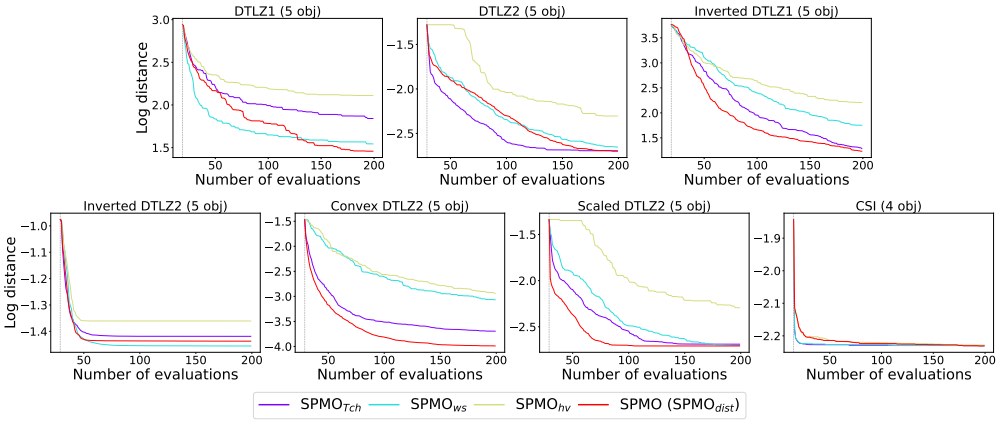
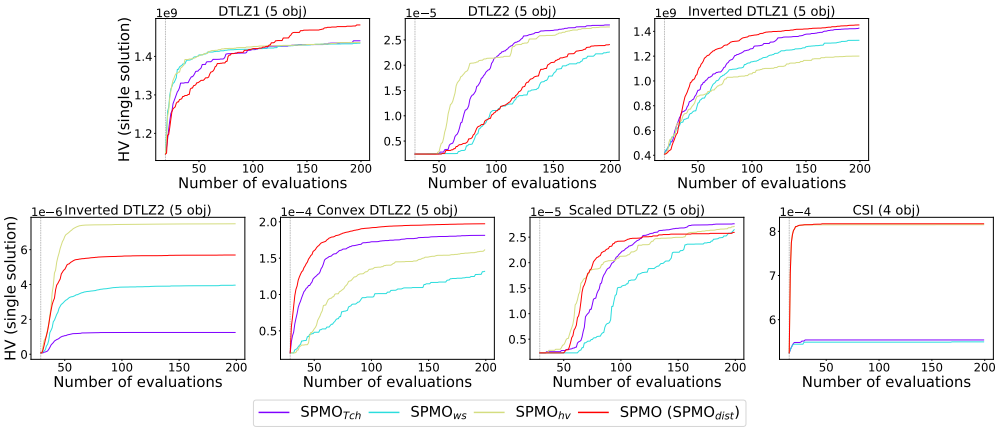


Figure 34: Violin plots of the preference-based single-point HV obtained by the SPMO using four different single-point metrics on the problems with 5 objectives. Each violin represents the distribution of the preference-based single-point HV obtained by a method over 30 independent runs.



(a) Trajectories of the preference-based distance metric (log distance).



(b) Trajectories of the preference-based single-point HV.

Figure 35: Trajectories of the preference-based distance metric and the preference-based single-point HV obtained by the SPMO using four different single-point metrics. Each coloured line represents the mean metric value on 30 independent runs (after the initial Sobol samples, represented by the dashed grey line).

F.7 Acquisition Wall Time

Table 28 presents the mean acquisition optimisation wall time of the eight methods. As shown, when the number of objectives is 3 or 5, the time of all the methods is acceptable with a maximum of 98 seconds. As the number of objectives increases to 10, hypervolume-based methods become very expensive. For example, EHVI and NEHVI take about half an hour and more than 3 hours, respectively. Note that although PSL takes only several minutes based on the initial samples, the hypervolume computation can take a day for a single evaluation when the number of non-dominated solutions becomes large. The proposed SPMO method shows high computational efficiency, achieving the lowest time requirement in four out of the six instances.

Table 28: Mean acquisition optimisation wall time in seconds based on the $2(d+1)$ initial Sobol samples on DTLZ1 problems with $m = 3, 5, 10$ objectives, where $d = m + 4$, over 30 runs. Experiments are conducted using a CPU (Intel Xeon CPU Platinum 8360Y @ 2.40 GHz) and a GPU (NVIDIA A100). Note that N/A means the wall time of NEHVI on DTLZ1 with 10 objectives exceeds 3 hours.

Device\Method	ParEGO	NParEGO	TS-TCH	EHVI	NEHVI	C-EHVI	JES	PSL	SPMO (ours)
CPU (3 obj)	3.46	3.49	12.29	4.23	6.15	10.23	9.24	12.24	2.40
GPU (3 obj)	5.19	3.94	14.74	3.05	5.73	9.48	14.02	8.06	2.24
CPU (5 obj)	2.59	2.14	28.32	36.03	97.98	29.89	25.16	19.35	2.58
GPU (5 obj)	9.02	10.33	22.29	10.32	63.90	23.89	25.80	15.42	5.31
CPU (10 obj)	8.76	5.97	87.44	1134.35	N/A	77.94	388.37	255.4	6.95
GPU (10 obj)	41.33	30.30	64.78	2426.04	N/A	86.45	676.31	218.64	24.05

G Applicability of the Proposed SPMO

In conventional multi-objective optimisation, algorithms are designed to approximate the entire Pareto front, so that a decision-maker can later select a preferred solution based on their own preferences. This is the ideal situation, as a well-represented Pareto front provides the most comprehensive view of the possible trade-offs (Jiang & Li, 2025a). However, under tight evaluation budgets - especially when many objectives are involved - it is often unrealistic, if not impossible, to obtain a good approximation of the Pareto front. In such settings, the decision-maker may benefit more from receiving a well-balanced solution that is close to the Pareto front, rather than a well-distributed solution set far from the front. The proposed SPMO framework is designed for this purpose: instead of spreading search effort across the whole front, it directs the optimisation towards a well-balanced solution. With a focused search effort, the obtained solution is often closer to the front, and thus has a higher likelihood of being selected by the decision-maker.

It is worth pointing out that if the optimisation problem under consideration is extremely simple (e.g., smooth, unimodal landscape with a very limited search space), on which finding a good representation of the entire Pareto front is possible under tight budgets, then our approach may not be desirable. Moreover, exploring the Pareto front can help decision-makers better understand the optimisation problem and facilitate the elicitation or refinement of their preferences. In such cases, our framework is not applicable.

H Extensions

The preceding discussion has addressed multi-objective optimisation problems in which evaluations are performed either sequentially or in batches, under both noiseless and noisy scenarios. However, not all multi-objective settings conform to these scenarios. To accommodate a broader class of problems, we propose several extensions that enable the methodology to handle more optimisation scenarios.

High-Dimensional Bayesian Optimisation (HDBO). High-dimensional black-box optimisation problems are highly challenging and frequently encountered in a wide range of applications. The dimensionality may range from tens to a billion (González-Duque et al., 2024; Papenmeier et al., 2023; Santoni et al., 2024; Wang et al., 2016; Hoang et al., 2025). To tackle such optimisation problems, various HDBO methods have been proposed (Binois & Wycoff, 2022; Chen et al., 2024; Nayebi et al., 2019; Wang et al., 2018; 2016; Xu et al., 2025). They can loosely be categorised into four classes (Santoni et al., 2024), i.e., variable selection (Eriksson & Jankowiak, 2021), additive models (Delbridge et al., 2020; Han et al., 2021; Wang et al.,

2018; Ziomek & Ammar, 2023), embeddings (Antonov et al., 2022; Letham et al., 2020; Raponi et al., 2020), and trust regions (Daulton et al., 2022b; Diouane et al., 2023; Eriksson et al., 2019). However, recent studies show that standard Gaussian processes without the above techniques can perform well in high-dimensional spaces (Hvarfner et al., 2024; Papenmeier et al., 2025; Xu et al., 2025; Jiang et al., 2026) and suggest that the main issue in high-dimensional BO is the gradient vanishing. Our work can be naturally extended to the high-dimensional setting by mitigating the gradient vanishing issue (Papenmeier et al., 2025; Xu et al., 2025).

Multi-Fidelity Bayesian Optimisation (MFBO). In many real-world optimisation scenarios, the evaluation is often available at multiple fidelity levels, where increasing fidelity typically leads to improved accuracy at the expense of higher computational cost. Many MFBO methods have been proposed to tackle such optimisation problems (Belakaria et al., 2020a; Kandasamy et al., 2017; Li et al., 2020; Moss et al., 2021; Song et al., 2019; Takeno et al., 2020; Wu et al., 2020; Zhang et al., 2017). Our proposed SPMO can be potentially extended to the multi-fidelity setting by integrating prior techniques, e.g., building multiple surrogate models of different levels of fidelity.