A PARALLEL TECHNIQUE FOR MULTI-OBJECTIVE BAYESIAN GLOBAL OPTIMIZATION: USING A BATCH SELECTION OF PROBABILITY OF IMPROVEMENT

Abstract

Bayesian global optimization (BGO) is an efficient surrogate-assisted technique for problems involving expensive evaluations. A parallel technique can be used to parallelly evaluate the true-expensive objective functions in one iteration to boost the execution time. An effective and straightforward approach is to design an acquisition function that can evaluate the performance of a bath of multiple solutions, instead of a single point/solution, in one iteration. This paper proposes five alternatives of Probability of Improvement (PoI) with multiple points in a batch (q-PoI) for multi-objective Bayesian global optimization (MOBGO), taking the covariance among multiple points into account. Both exact computational formulas and the Monte Carlo approximation algorithms for all proposed q-PoIs are provided. Based on the distribution of the multiple points relevant to the Pareto-front, the position-dependent behavior of the five q-PoIs is investigated. Moreover, the five q-PoIs are compared with the other nine state-of-the-art and recently proposed batch MOBGO algorithms on twenty bio-objective benchmarks. The empirical experiments on different variety of benchmarks are conducted to demonstrate the effectiveness of two greedy q-PoIs (q-PoI best and q-PoI all) on low-dimensional problems and the effectiveness of two explorative q-PoIs (q-PoI one and q-PoI worst) on high-dimensional problems with difficult-to-approximate Pareto front boundaries.

Keywords Surrogate model · Parallelization · Multi-objective Bayesian global Optimization · Probability of Improvement · Batch Selection · Gaussian Processes

1 Introduction

The surrogate model is regarded as a promising approach to incorporating powerful computational techniques into simulation-based optimization, as they replace exact but expensive simulation outputs with approximations learned from past outputs. Compared to the exact evaluations, the more expensive the evaluation of the simulation is, the greater the leverage of such approaches is in terms of the runtime boost. Two typical examples of time-consuming simulation models are finite element simulations [1] and computational fluid dynamics (CDF) simulations [2], and in both cases, a single run can consume many hours. Therefore, most evolutionary optimization algorithms can not be directly applied to such expensive simulation problems. The most common remedy for computationally expensive simulation is to
replace exact objective function evaluations with predictions from surrogate models, for instance, Gaussian processes or Kriging models [2, 3], random forest [4], supported vector machine [5], symbolic regression [1], to name a few.

Bayesian Global Optimization (BGO) was proposed by Jonas Mockus and Antanas Žilinskas [6, 7], and it is popularized by Jones et al. [8] (known as EGO). Underpinned by surrogate-assisted modeling techniques, it sequentially selects promising solutions by using the prediction and uncertainty of the surrogate model. The basic idea of BGO is to build a Gaussian process model to reflect the relationship between a decision vector $x = (x_1, \cdots, x_d)^T$ and its corresponding objective value $y = f(x)$. A BGO algorithm searches for an optimal solution $x^*$ by using the predictions of the Gaussian process model instead of evaluating the really expensive objective function. During this step, an infill criterion, also known as an acquisition function, quantitatively measures how good the optimal solution $x^*$ is. Then, the optimal solution $x^*$ and its corresponding objective value, evaluated by the real objective function $f$, will be used to update the Gaussian process model. However, only a single solution can be optimized within one iteration, which prevents to take advantage of parallel computation facilities for expensive simulations.

A batch acquisition function for single-objective problems, multiple-point expected improvement (q-EI), was firstly pointed out, though not developed, by Schonlau [9] to measure the expected improvement (EI) for a batch with $q$ points $(x^{(1)}, \cdots, x^{(q)})$ by their corresponding predicted distributions $\hat{Y} = (\hat{y}^{(1)}, \cdots, \hat{y}^{(q)})$. The predicted distributions $\hat{Y}$ follows a multivariate normal distribution with a mean vector $M = (\mu^{(1)}, \cdots, \mu^{(q)})$ and a covariance matrix $\Sigma$, where both $M$ and $\Sigma$ can be estimated by a Gaussian process predictor. Later, q-EI was well developed by Ginsbourger et al. in [10, 11, 12], where the explicit formula of q-EI was provided. By maximizing the q-EI, an optimal batch consisting of $q$ points can be searched. This batch criterion is of particular interest in real-world applications, as it allows multiple CPUs to evaluate the $q$ points simultaneously and thus shortens the execution time of Bayesian global optimization. Moreover, the correlation among predictions, which can be reflected by the covariance $\text{Cov}(\hat{y}^{(i)}, \hat{y}^{(j)}) = E[\hat{y}^{(i)} \hat{y}^{(j)}]_{i,j=1,\cdots,q, i\neq j}$, are involved in the computation of q-EI [11, 12]. The correlation in the computation of a batch criterion can measure the difference affected by the relationships between any two predictions in $Y$. In this sense, correlations can theoretically promote the performance of a batch-criterion-based Bayesian global optimization algorithm, as shown in [11].

Generalizing the BGO into multi-objective cases, a multi-objective Bayesian global optimization (MOBGO) algorithm builds up independent models for each objective. Various approaches have been studied and can be utilized for parallelization in MOBGO. For instance, MOEA/D-EGO proposed in [13] generalizes ParEGO by setting different weights and then performs MOEA/D parallelly to search for the multiple points. Ginsbourger et al. [11] proposed to approximate q-EI by using the techniques of Kriging believer (KB) and Constant liar (CL). Both KB and CL can be directly and easily extended for MOBGO algorithms. Bradford et al. proposed to utilize Thompson Sampling on the GP posterior as an acquisition function and to select the batch by maximizing the hypervolume (HV) from the population optimized by NSGA-II. Yang et al. proposed to divide an objective space into several sub-spaces, and to search/evaluate the optimal solutions in each sub-space by maximizing the truncated expected hypervolume improvement parallelly [14]. Gaudrie et al. [15] proposed to search for multiple optimal solutions in several different preferred regions simultaneously by maximizing the expected hypervolume improvement (EHVI) with different reference point settings. DGEMO proposed in [16] utilizes the mean function of GP posterior as the acquisition function, then optimizes the acquisition function by a so-called ‘discovery algorithm’ in [17], and then selects the batch based on the diversity information in both decision and objective spaces. MOEA/D-ASS proposed in [18] utilizes the adaptive lower confidence bound as the acquisition function and introduces an adaptive subproblem selection (ASS) to identify the most promising subproblems for further modeling. Recently, the $\epsilon$-greedy strategy, which combines greedy search and a random selection, also shows the efficacy on both single- and multi-objective problems [19, 20] especially on high-dimensional problems [21, 20].

However, distinguished from q-EI in the single-objective BGO, none of the parallel techniques mentioned above consider correlations among multiple predictions within a batch in MOBGO. Recently, Daulton et al. [22] proposed the multiple-point EHVI (q-EHVI) that incorporates the correlations among multiple predictions in a single coordinate. However, the correlation is utilized by using the Monte Carlo (MC) method instead of an exact calculation method. Consequently, the approximation error of q-EHVI by the MC method may render MOBGO’s performance in optimization processes. Because for an indicator-based optimization algorithm, the optimization results of using an approximation method are not competitive compared with that of exact computation methods [23].

This paper focuses on another common acquisition function, namely, probability of improvement (Pol) [24, 25], which is widely used within the frameworks of BGO and MOBGO [26]. The main contribution of the paper is that we propose five different types of multiple-point probability of improvement (q-Pol), and also provide explicit formulas for their exact computations. The remaining parts of this paper are structured as follows: Section 2 introduces the related definitions and the background of multi-objective Bayesian global optimization. Section 3 describes the assumptions and proposes the five different q-Pol, provides both the MC method and explicit formulas to approximate/compute
q-PoIs, and analyzes the position-dependent behaviors of q-PoIs w.r.t. standard deviation and coefficient. Section 4 discusses the optimization studies using the five q-PoIs within the MOBGO framework on twenty bio-objective optimization problems.

2 Multi-objective Bayesian Global Optimization

2.1 Multi-objective Optimization Problems

A multi-objective optimization (MOO) problem involves multiple objective functions to be minimized simultaneously. A MOO problem can be formulated as:

$$\min_{x} f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^T \quad \text{for} \quad x \in \mathcal{X} \subseteq \mathbb{R}^d$$

where $m$ is the number of objective functions, $f_i$ stands for the $i$-th objective function $f_i : \mathcal{X} \rightarrow \mathbb{R}, i = 1, \ldots, m$, $\mathcal{X}$ is a decision vector subset. For simplicity, we restrict $\mathcal{X}$ to be a subset of a continuous space, that is $\mathcal{X} \subseteq \mathbb{R}^d$ in this paper. Theoretically, $\mathcal{X}$ can also be a subset of a discrete alphabet $\mathcal{X} \subseteq \{0,1\}^d$ or even a mixed space, e.g., $\mathcal{X} \subseteq \mathbb{R}^{d_1} \times \mathbb{N}^{d_2} \times \{0,1\}^{d_3}$, which can be achieved by a heterogeneous metric for the computation of distance function [27] or the one-hot strategy encoding [28] in Gaussian process. In this paper, $d$ denotes the dimension of the search space $\mathcal{X}$.

2.2 Gaussian Process

Gaussian process regression is used as the surrogate model in Bayesian Global optimization to approximate the unknown objective function and quantify the uncertainty of a prediction. In this technique, the uncertainty of the objective function $f$ is modeled as a probability distribution of function, which is achieved by posing a prior Gaussian process on it. We consider $\mathcal{X} = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}, x^{(i)} \in \mathcal{X}$ a set of decision vectors, which are usually obtained by some sampling methods (e.g., the Latin hypercube sampling [29]) and associated objective function values

$$\psi = f(\mathcal{X}) = [f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(n)})]^T.$$ 

Then the objective function can be modeled as a centered Gaussian process (GP) prior to an unknown constant trend term $\mu$ (to be estimated):

$$f \sim \mathcal{GP}(0, k(\cdot, \cdot))$$  \hspace{1cm} (2-1)

where $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a positive definite function (a.k.a. kernel) that computes the autocovariance of the process, namely $k(x, x') = \text{Cov}(f(x), f(x'))$. The most well-known kernel is the so-called Gaussian kernel (a.k.a. radial basis function (RBF)) [30]:

$$k(x, x') = \sigma^2 \exp \left( -\sum_{i=1}^{d} \frac{(x_i - x'_i)^2}{2\theta_i^2} \right),$$  \hspace{1cm} (2-2)

where $\sigma^2$ models the variance of function values at each point and $\theta_i$ are kernel parameters representing variables’ importance, they are typically estimated from data using the maximum likelihood principle. The optimal $\theta = (\theta_1^{opt}, \ldots, \theta_d^{opt})$ of GP models can be optimized by any continuous optimization algorithm.

For an unknown point $x$, a Bayesian inference yields the posterior distribution of $f$, i.e., $p(f(x) \mid \psi) \propto p(\psi \mid f(x))p(f(x))$. Note that this posterior probability is also a conditional probability due to the fact that $f(x)$ and $\psi$ are jointly Gaussian, namely,

$$\begin{bmatrix} f(x) \\ \psi \end{bmatrix} \sim \mathcal{N} \left( 0, \begin{bmatrix} \sigma^2 & \mathbf{k}^\top \\ \mathbf{k} & \mathbf{K} \end{bmatrix} \right),$$

where $K_{ij} = k(x^{(i)}, x^{(j)})$ and $k(x) = (k(x, x^{(1)}), k(x, x^{(2)}), \ldots, k(x, x^{(n)}))^\top$. Conditioning on $\psi$, we obtain the posterior of $f$:

$$f(x) \mid \psi \sim \mathcal{N} \left( k^\top K^{-1} \psi, \sigma^2 - k^\top K^{-1} k \right).$$  \hspace{1cm} (2-3)

Given this posterior, it is obvious that the best-unbiased predictor of $Y$ is the posterior mean, i.e., $\mu = k^\top K^{-1} \psi$, which is also the Maximum a Posterior Probability (MAP) estimation. The MSE of $\mu$ is $s^2 = E\{\mu - f\}^2 = \sigma^2 - k^\top K^{-1} k$, which is also the posterior variance.
Moreover, to see the covariance structure of the posterior process, we can consider \( q \) unknown points \( x^{(1)}, x^{(2)}, \ldots, x^{(q)} \in \mathcal{X} \):

\[
\begin{bmatrix}
    f(x^{(1)}) \\
    f(x^{(2)}) \\
    \vdots \\
    f(x^{(q)}) \\
    \psi
\end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix}
    \sigma^2 & k(x^{(1)}, x^{(2)}) & \cdots & k(x^{(1)}, x^{(q)}) & k_1^T \\
    k(x^{(2)}, x^{(1)}) & \sigma^2 & \cdots & k(x^{(2)}, x^{(q)}) & k_2^T \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    k(x^{(q)}, x^{(1)}) & k(x^{(q)}, x^{(2)}) & \cdots & \sigma^2 & k_q^T \\
    k_1 & k_2 & \cdots & k_q & K
\end{bmatrix}),
\]

in which \( k_i = k(x^{(i)}), i \in \{1, 2, \ldots, q\} \). After conditioning on \( \psi \), we obtained the following distribution:

\[
[f(x^{(1)}, f(x^{(2)}), \ldots, f(x^{(q)}))]^T \mid \psi \sim \mathcal{N}(\mu, \Sigma).
\]

(2-4)

\( \mu = [k_1^T K^{-1} \psi, k_2^T K^{-1} \psi, \ldots, k_q^T K^{-1} \psi]^T \) and

\[
\Sigma = \begin{bmatrix}
    \sigma^2 - k_1^T K^{-1} k_1 & k(x^{(1)}, x^{(2)}) - k_1^T K^{-1} k_2 & \cdots & k(x^{(1)}, x^{(q)}) - k_1^T K^{-1} k_q \\
    k(x^{(2)}, x^{(1)}) - k_2^T K^{-1} k_1 & \sigma^2 - k_2^T K^{-1} k_2 & \cdots & k(x^{(2)}, x^{(q)}) - k_2^T K^{-1} k_q \\
    \vdots & \vdots & \ddots & \vdots \\
    k(x^{(q)}, x^{(1)}) - k_q^T K^{-1} k_1 & k(x^{(q)}, x^{(2)}) - k_q^T K^{-1} k_2 & \cdots & \sigma^2 - k_q^T K^{-1} k_q
\end{bmatrix}.
\]

In this posterior formulation, it is clear to see that the covariance at two different arbitrary locations \( (x^{(1)} \) and \( x^{(2)} \) is expressed in the cross-term of the posterior covariance matrix as follows:

\[
k'(x^{(1)}, x^{(2)}) = \text{Cov}(f(x^{(1)}), f(x^{(2)}) \mid \psi) = k(x^{(1)}, x^{(2)}) - k(x^{(1)})^T K^{-1} k(x^{(2)}),
\]

(2-5)

where \( k' \) is additional information and is essential to design a multiple-point infill criterion (a.k.a. acquisition function) parallel for techniques in BGO algorithms, e.g., the multi-point expected improvement (q-EI) [31]. However, \( k' \) is not utilized to compute a multi-objective acquisition function in terms of an exact computational approach \(^1\).

### 2.3 Structure of MOBGO

Similar to single-objective Bayesian global optimization, MOBGO starts with sampling an initial design of experiment (DoE) with a size of \( \eta \) (line 2 in Alg. 1), \( X = \{x^{(1)}, x^{(2)}, \ldots, x^{(\eta)}\} \subseteq \mathcal{X} \). DoE is usually generated by simple random sampling or Latin Hypercube Sampling [32]. By using the initial DoE, \( X \) and its corresponding objective values, \( Y = \{f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(\eta)})\} \subseteq \mathbb{R}^{m \times \eta} \) (line 3 in Alg. 1), surrogate models \( \mathcal{M}_i \) can be constructed to describe the probability distribution of the objective function \( f_i \) conditioned on the initial evidence \( Y_i \), namely \( \text{Pr}(f_i \mid X, Y_i) \) (line 4 in Alg. 1), where \( Y_i \) represents \( f_i(X) \). Between two surrogate models, it is widely assumed that model \( \mathcal{M}_i \) is independent of \( \mathcal{M}_j, \forall i \neq j, i, j \in \{1, 2, \ldots, m\} \).

Once the surrogate models \( \mathcal{M} \) are constructed, MOBGO enters the main loop until a stopping criterion is fulfilled \(^3\), as shown in Alg. 1 from line 6 to line 12. The main-loop starts with searching for a decision vector set \( X' \) in the search space \( \mathcal{X} \) by maximizing the acquisition function \( \phi' \) with parameters of \( \gamma \) and surrogate models \( \mathcal{M} \) (line 7 in Alg. 1). \( q \) represents the batch size or the number of possible decision vectors in \( X' \). The value of \( q \) is determined by the acquisition function’s theoretical definition and properties \( \phi' \). For the acquisition functions of PoI and EHVI, \( q \) can only be set as 1 by using an exact computation of the \( \phi' \). Other approaches, like KB, CL\(^4\), and other methods in [34, 15], theoretically can compute exact \( \phi' \) by using \( q = 1 \). Therefore, these methods can not utilize the covariance among multiple predictions \( k' \) in Eq. (2-5) to guide the optimization processes at line 7 in Alg. 1. A single-objective optimization algorithm searches for the optimal decision vector set \( \mathcal{X}^* \). Theoretically, any single-objective optimization algorithm can be utilized, e.g., genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization algorithm (ACO), covariance matrix adaptation evolution strategy (CMA-ES), and even gradient-ascent algorithms [35]. The optimal decision vector set \( X^* \) will then be evaluated by the ‘true’ objective functions \( f \). When \( q > 1 \), parallelization techniques can be utilized to evaluate multiple solutions in \( X^* \). The surrogate models \( \mathcal{M} \) will be retrained by the updated \( X \) and \( Y \).

\(^{1}\)A recent work in [22] utilizes \( \Sigma \) using MC integration with samples from the joint posterior in terms of sampling by an MC method to approximate multiple-point Expected Hypervolume Improvement, instead of an exact computational method.

\(^{3}\)In [33], a so-called dependent Gaussian process was proposed to learn the correlation between different processes. The research in this paper only considers independent Gaussian Processes.

\(^{4}\)In this paper, we restrict the stopping criterion as the number of iterations.

\(^{4}\)The details of KB and CL can be found in Algorithm 1, and Algorithm 2 in [11].
Algorithm 1: Multi-objective Bayesian Global Optimization

```plaintext
MOBGO(f, Φ, X, M, γ, η, Tc)
/* f: objective functions, Φ: acquisition function, X: search space, γ:
parameters of Φ, M: a surrogate model to train, Tc: maximum number of
function evaluations */
1:Generate the initial DoE: X = \{x(1), x(2), ..., x(η)\} \subset X;
2:Evaluate Y ← \{f(x(1)), f(x(2)), ..., f(x(η))\};
3:Train surrogate models \mathcal{M}_i on (X, Y_i), where i = 1, ..., m;
4:g ← η;
5:while g < Tc do
6:X* ← \arg\max \Phi(X'; \mathcal{M}, γ), where \mathcal{M} = \{M_1, ..., M_m\} and X' = \{x'(1), x'(2), ..., x'(η)\} \in X;
7:Y* ← f(X*);
8:X ← X ∪ \{X*\};
9:Y ← Y ∪ \{Y*\};
10:Re-train the surrogate models \mathcal{M}_i on (X, Y_i), where i = 1, ..., m;
11:g ← g + q;
```

3 Multiple-point Probability of Improvement

3.1 Related Definitions

Pareto dominance, or briefly dominance, is an ordering relationship on a set of potential solutions. Dominance is defined as follows:

Definition 3.1 (Dominance – ≪ [36]) Given two decision vectors x(1), x(2) ∈ \mathbb{R}^d and their corresponding objective values y(1) = f(x(1)), y(2) = f(x(2)) in a minimization problem, it is said that y(1) dominates y(2), being represented by y(1) ≪ y(2), iff ∀i ∈ \{1, 2, ..., m\} : f_i(x(1)) ≤ f_i(x(2)) and ∃j ∈ \{1, 2, ..., m\} : f_j(x(1)) < f_j(x(2)).

Definition 3.2 (Non-Dominated Space of a Set [37]) Let \mathcal{F} be a subset of \mathbb{R}^m and let a reference point r ∈ \mathbb{R}^m be such that ∀p ∈ \mathcal{F} : p ≪ r. The non-dominated space of \mathcal{F} with respect to r, denoted as ndom(\mathcal{F}), is then defined as:

\[
\text{ndom}(\mathcal{F}, r) := \{y ∈ \mathbb{R}^m | y ≪ r \text{ and } \nexists p ∈ \mathcal{F} \text{ such that } p ≪ y\} \tag{3-1}
\]

Note that a reference point r shall be chosen so that every possible solution dominates r. A reference point r should avoid being an infinity vector in hypervolume-based indicators. In this paper, r is an infinity vector for PoI and its variants, as the maximum value is bounded by 1.

Definition 3.3 (Probability of Improvement [24, 26]) Given the predictions \hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m)^T with the parameters of the multivariate predictive distribution \mu = (\mu_1, \mu_2, \ldots, \mu_m)^T, s = (s_1, s_2, \ldots, s_m)^T and the Pareto-front approximation set \mathcal{F}, the Probability of Improvement (PoI) is defined as:

\[
\text{PoI} (\mu, s, \mathcal{F}) := \int_{\mathbb{R}^m} \mathbb{I}(\text{impr } \mathcal{F}) \xi_{\mu, s}(\hat{y}) \text{d}\hat{y} \text{ and } \mathbb{I}(v) = \begin{cases} 1, & v = \text{true} \\ 0, & v = \text{false} \end{cases} \tag{3-2}
\]

where \xi_{\mu, s} is the multivariate independent normal distribution with the mean values \mu ∈ \mathbb{R}^m and the standard deviations s ∈ \mathbb{R}^m. Here \text{impr } \mathcal{F} represents \hat{y} ∈ \text{ndom}(\mathcal{F}).

Here, \hat{y}_i ∼ \mathcal{N}(\mu_i, s_i), i ∈ \{1, 2, ..., m\}. 

---

---
3.2 Assumptions in q-PoI

Definition 3.2 can be generalized to multiple-point PoI (q-PoI), which is a cumulative probability of a batch \( \hat{\mathbf{Y}} = \{ \hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(q)} \} \) in the whole \( \text{ndom}(\mathcal{P}_F) \), where \( \hat{y}^{(i)} = (\hat{y}_1^{(i)}, \ldots, \hat{y}_m^{(i)})^\top \) \( |i=1,\ldots,q| \). In this section, five q-PoIs are defined accordingly by generalizing the single point PoI in Eq. (3.2).

Like the assumption in single-point PoI in Eq. (3.2), each objective is also assumed to be independent of the other in q-PoI. However, the correlation of multiple points in a particular objective, which is available in the posterior distributions of Gaussian processes, will be considered in q-PoI. That is to say: Suppose a prediction \( y^{(j)}_i \sim \mathcal{N}(\mu^{(j)}_i, \sigma^{(j)}_i) \), where \( i \in \mathbb{M} := \{1, 2, \cdots, m\} \) and \( j \in \mathbb{Q} := \{1, 2, \cdots, q\} \), we consider that:

1. \( y^{(j)}_i \not\perp \perp y^{(j)}_i \) \( \implies \text{Cov}(y^{(j)}_i, y^{(j)}_i) \neq 0, \quad \forall i \in \mathbb{M} \) and \( \forall j, jj \in \mathbb{Q}|_{j \neq jj} \)
2. \( y^{(j)}_i \not\perp \perp y^{(j)}_i \) \( \implies \text{Cov}(y^{(j)}_i, y^{(j)}_i) = 0, \quad \forall i, ii \in \mathbb{M}|_{i \neq ii} \) and \( \forall j, jj \in \mathbb{Q}|_{j \neq jj} \)

For simplicity, we use \( \mathbf{M} \) and \( \Sigma \) to denote the predicted \( \mu \) matrix and covariance matrices of a batch \( \hat{\mathbf{Y}} = \{ \hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(q)} \} \), respectively. They are defined as:

\[
\mathbf{M}_{m \times q} = \begin{bmatrix}
\mu^{(1)}_1 & \cdots & \mu^{(q)}_1 \\
\vdots & \ddots & \vdots \\
\mu^{(1)}_m & \cdots & \mu^{(q)}_m
\end{bmatrix} = \begin{bmatrix}
\mathbf{\mu}_1, \cdots, \mathbf{\mu}_m
\end{bmatrix}^\top = \begin{bmatrix}
\mathbf{\mu}^{(1)}, \cdots, \mathbf{\mu}^{(q)}
\end{bmatrix},
\]

where \( \mathbf{\mu}_i = (\mu^{(1)}_i, \cdots, \mu^{(q)}_i)^\top \) and \( \mathbf{\mu}^{(j)} = (\mu^{(j)}_1, \cdots, \mu^{(j)}_m)^\top \),

\[
\Sigma = \{ \Sigma_1, \cdots, \Sigma_m \} \text{ where } \Sigma_i = \begin{bmatrix}
\text{Cov}(s^{(1)}_i, s^{(1)}_i) & \cdots & \text{Cov}(s^{(1)}_i, s^{(q)}_i) \\
\vdots & \ddots & \vdots \\
\text{Cov}(s^{(q)}_i, s^{(1)}_i) & \cdots & \text{Cov}(s^{(q)}_i, s^{(q)}_i)
\end{bmatrix}
\]

In Eq. (3.4), \( \text{Cov}(s^{(j)}_i, s^{(j)}_j) = (\sigma^{(j)}_j)^2, i = 1, 2, \cdots, m \) and \( j = 1, 2, \cdots, q \). Moreover, to simplify, \( \Sigma \) represents the set of covariance matrices of \( q \) predictions over \( m \) objectives/coordinates (\( \Sigma_i |_{i \in \mathbb{M}} \)) with a size of \( m \times q \times q \). If only the diagonal elements in \( \Sigma_i |_{i \in \mathbb{M}} \) are considered. In the other words, the correlation is not considered. \( \Sigma \) can be reduced from a \( m \times q \times q \) matrix into a \( m \times q \) matrix, and noted as:

\[
\Lambda = \begin{bmatrix}
s^{(1)}_1 & \cdots & s^{(q)}_1 \\
\vdots & \ddots & \vdots \\
s^{(1)}_m & \cdots & s^{(q)}_m
\end{bmatrix} = [s^{(1)}, \cdots, s^{(q)}] \text{ where } \begin{cases}
\mathbf{s}^{(i)} = [s^{(i)}_1, \cdots, s^{(i)}_m]^\top \\
i \in \{1, \cdots, q\}
\end{cases}
\]

Remark: Notice that each element in \( \Lambda \) is a standard deviation of a normal distribution. It differs from \( \Sigma \) in which each element is a (co-)variance matrix.

Example 3.1 In Fig. 1, a Pareto-front approximation is composed of \( y^{(1)} = (3, 1) \), \( y^{(2)} = (2, 1.5) \) and \( y^{(3)} = (1, 2.5) \) and a batch \( \hat{\mathbf{Y}} = \{ \hat{y}^{(1)}, \hat{y}^{(2)} \} \), of which parameters \( \mathbf{\mu} \) and \( \mathbf{s} \) are: \( \mu^{(1)}_1 = 1.5, \mu^{(1)}_2 = 0.5, \mu^{(2)}_1 = 2.5, \mu^{(2)}_2 = 0 \) and \( s^{(1)}_1 = 0.6, s^{(2)}_1 = 0.7, i \in \{1, 2\} \). The orange spheres are the multivariate normal distributions of \( \hat{y}^{(1)} \) and \( \hat{y}^{(2)} \).

3.3 Definitions of Five Proposed q-PoIs in Different Cases

The motivation for introducing the following q-PoI variants is based on different perspectives on designing the improvement function \( I(.) \). The first straightforward idea is to guarantee that all points in the batch can improve \( \mathcal{P}_F \) simultaneously. This idea formulates the concept of q-PoI \textit{all}. Similarly, the idea behind q-PoI \textit{one} is to make sure at least one point in the batch size can improve \( \mathcal{P}_F \). The idea of q-PoI \textit{best} and q-PoI \textit{worst} is to ensure that the \textit{best} or \textit{worst} projected values (from all points onto each dimension) can improve \( \mathcal{P}_F \). In q-PoI \textit{mean}, the average PoI value of each point in a batch is computed. In the following, we present five different q-PoIs under concern.

Definition 3.4 Given \( m \times q \) multivariate normal distributions \( \hat{\mathbf{Y}} = \{ \hat{y}^{(1)}, \hat{y}^{(2)}, \cdots, \hat{y}^{(q)} \} \) with a mean matrix \( \mathbf{M} = [\mu^{(1)}, \cdots, \mu^{(q)}]^\top \) and covariance matrices \( \Sigma = \{ \Sigma_1, \cdots, \} \).
\( \Sigma_m \), where a root of diagonal elements in \( \Sigma_m \) composes a standard deviation matrix \( \Lambda = [s^{(1)}, \ldots, s^{(q)}]^{\top} \). For a Pareto-front approximation set \( \mathcal{P} \), we define the following PoIs for the batch \( \hat{Y} \):

\[
\begin{align*}
q-PoI_{\text{all}}(M, \Sigma, \mathcal{P} \mathcal{F}) & := \int_{\mathbb{R}^{m \times q}} I(\hat{y}^{(1)}_{\text{impr}} \mathcal{P} \mathcal{F}) \cap \cdots \cap I(\hat{y}^{(q)}_{\text{impr}} \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y}) \, d\hat{Y} \\
q-PoI_{\text{one}}(M, \Sigma, \mathcal{P} \mathcal{F}) & := \int_{\mathbb{R}^{m \times q}} I(\hat{y}^{(1)}_{\text{impr}} \mathcal{P} \mathcal{F}) \cup \cdots \cup I(\hat{y}^{(q)}_{\text{impr}} \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y}) \, d\hat{Y} \\
q-PoI_{\text{best}}(M, \Sigma, \mathcal{P} \mathcal{F}) & := \int_{\mathbb{R}^{m \times q}} I(\text{DisC}(\hat{Y})_{\text{impr}} \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y}) \, d\hat{Y} \\
q-PoI_{\text{worst}}(M, \Sigma, \mathcal{P} \mathcal{F}) & := \int_{\mathbb{R}^{m \times q}} I(\text{ConJ}(\hat{Y})_{\text{impr}} \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y}) \, d\hat{Y} \\
q-PoI_{\text{mean}}(M, \Sigma, \mathcal{P} \mathcal{F}) & := \frac{1}{q} \sum_{i=1}^{q} \int_{\mathbb{R}^{m}} I(\hat{y}^{(i)}_{\text{impr}} \mathcal{P} \mathcal{F}) \xi_{\mu^{(i)}_{\text{impr}}, \sigma^{(i)}_{\text{impr}}}(\hat{y}^{(i)}) \, d\hat{y}^{(i)}
\end{align*}
\]

where \( \xi_{M, \Sigma} \) is the multivariate normal distribution, \( \text{DisC}(\hat{Y}) = (\hat{Y}_1 \lor \hat{Y}_2 \lor \cdots \lor \hat{Y}_m) = \max(\hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_m)^\top \), \( \text{ConJ}(\hat{Y}) = (\hat{Y}_1 \land \hat{Y}_2 \land \cdots \land \hat{Y}_m) = \min(\hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_m)^\top \), \( \hat{Y}_i = (\hat{y}^{(1)}_i, \ldots, \hat{y}^{(q)}_i) \) for \( i \in \mathbb{M} \), and \( \cdot \) is again the indicator function defined in Eq. (3-2).

Note that correlations in Eq. (3-9) are exclusive of marginal probability density functions, and only the diagonal elements \( \Lambda \) in \( \Sigma_{m} \) are used here.

The required condition in q-PoI\textsubscript{best} is sufficient for the condition in q-PoI\textsubscript{all}, but not necessary. Given a \( \hat{Y} \) of parameters \( M, \Sigma \) and a \( \mathcal{P} \), if q-PoI\textsubscript{best}(\( M, \Sigma, \mathcal{P} \mathcal{F} \)) = 1, then q-PoI\textsubscript{all}(\( M, \Sigma, \mathcal{P} \mathcal{F} \)) = 1. However, q-PoI\textsubscript{all}(\( M, \Sigma, \mathcal{P} \mathcal{F} \)) = 1 can not guarantee that q-PoI\textsubscript{best}(\( M, \Sigma, \mathcal{P} \mathcal{F} \)) = 1, due to the stricter condition in q-PoI\textsubscript{best}. Therefore, q-PoI\textsubscript{best} is theoretically more greedy than q-PoI\textsubscript{all}. 
3.4 Approximation of q-PoI

Suppose that we have a Pareto-front approximation set \( \mathcal{PF} \), a mean matrix \( M \), two covariance matrices \( \Sigma_1 \) and \( \Sigma_2 \) for each objective/coordinate, where:

\[
M = \begin{bmatrix}
\mu^{(1)}_1 & \mu^{(2)}_1 \\
\mu^{(1)}_2 & \mu^{(2)}_2
\end{bmatrix}
\quad \text{and} \quad
\Sigma_i = \begin{bmatrix}
\sigma^{(1)}_i & \text{Cov}(s^{(1)}_i, s^{(2)}_i) \\
\text{Cov}(s^{(1)}_i, s^{(2)}_i) & \sigma^{(2)}_i
\end{bmatrix}, \ i = 1, 2
\]

The parameter of \( M, \Sigma_1 \) and \( \Sigma_2 \) can compose a predicted batch in a 2-dimensional objective space, where \( \hat{\mathbf{y}} = (\hat{y}^{(1)}, \hat{y}^{(2)}) \). That is to say:

\[
\hat{y}^{(1)} \sim \mathcal{N}(\mu^{(1)}, \sigma^{(1)}_1) \quad \text{and} \quad
\hat{y}^{(2)} \sim \mathcal{N}(\mu^{(2)}, \sigma^{(2)}_2)
\]

Algorithm 2: Monte Carlo approximation for q-PoI of \( q = 2 \)

| Input: | Mean matrix \( M = [\mu_1, \mu_2] \), covariance matrices \( \Sigma = \{\Sigma_1, \Sigma_2\} \), number of samples \( n_{\text{sample}} \). A Pareto-front approximation set \( \mathcal{PF} \). |
|---|---|
| Output: | q-PoI case for case \( \in \{\text{best, worst, all, one, mean}\} \) |
| for \( i = 1 \) to \( n_{\text{sample}} \) do | /* Main loop */ |
| Random generate a vector \( \hat{\mathbf{y}}_1 = (\hat{y}^{(1)}_1, \hat{y}^{(2)}_1) \sim \mathcal{N}(M_1, \Sigma_1) \); | |
| Random generate a vector \( \hat{\mathbf{y}}_2 = (\hat{y}^{(1)}_2, \hat{y}^{(2)}_2) \sim \mathcal{N}(M_2, \Sigma_2) \); | |
| Compose the two solutions \( \hat{\mathbf{y}}^{(1)} = (\hat{y}^{(1)}_1, \hat{y}^{(1)}_2)^\top \) and \( \hat{\mathbf{y}}^{(2)} = (\hat{y}^{(2)}_1, \hat{y}^{(2)}_2)^\top \); | |
| flag\_best = flag\_best + 1(I(\max(\hat{y}_1), \max(\hat{y}_2))^\top \prec \mathcal{PF}); | |
| flag\_worst = flag\_worst + 1(I(\min(\hat{y}_1), \min(\hat{y}_2))^\top \prec \mathcal{PF}); | |
| flag\_all = flag\_all + 1(\hat{y}^{(1)} \prec \mathcal{PF}) \land 1(\hat{y}^{(2)} \prec \mathcal{PF}); | |
| flag\_one = flag\_one + 1(\hat{y}^{(1)} \prec \mathcal{PF}) \lor 1(\hat{y}^{(2)} \prec \mathcal{PF}); | |
| q-PoI case = flag\_case \div n_{\text{sample}} \text{ for case } \in \{\text{best, worst, all, one, mean}\}; | |

3.5 Explicit Computational Formulas for q-PoI based on MCDF

In this subsection, we present both explicit computational formulas for the proposed five q-Pois, and focus on the bi-objective (i.e., \( m = 2 \)) optimization problems for \( q = 2 \). All the following formulas assume \( q = 2 \) if there is no special statement.

For consistent lucidity, we briefly describe the partitioning method for bi-objective optimization problems in [37, 39]. For a sorted Pareto-front set \( \mathcal{PF} = (\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(n)}) \) by descending order in \( f_2 \), we augment \( \mathcal{PF} \) with two sentinels: \( \mathbf{y}^{(0)} = (r_1, -\infty) \) and \( \mathbf{y}^{(n+1)} = (-\infty, r_2) \). Then the non-dominated space of \( \mathcal{PF} \) can be represented by the
To evaluate the integrals in Eq. (3-12), one can adopt multiple numerical estimation methods, including an adaptive quadrature on a transformation of the t-density for bivariate and trivariate distributions [40, 41, 42], a quasi-Monte Carlo integration algorithm for higher dimensional distributions (m ≤ 4) [43, 44], and other methods as described in [45, 46]. The built-in function of mvncdf in MATLAB is employed for the computation of MCDF\((x, \mu, \Sigma)\). In this case, to calculate the cumulative probability density for Cartesian product domain of the form \((a, b] \times (c, d]\), we introduce the notation \(\Gamma(a, b, c, d, \mu, \Sigma)\) as below

\[
\Gamma(a, b, c, d, \mu, \Sigma) := \int_{(a, c)}^{(b, d)} \xi_{\mu, \Sigma}(y) dy
\]

\[
= \left( \int_{(-\infty, -\infty)}^{(b, d)} + \int_{(-\infty, -\infty)}^{(a, c)} - \int_{(-\infty, -\infty)}^{(a, d)} - \int_{(-\infty, -\infty)}^{(b, c)} \right) \xi_{\mu, \Sigma}(y) dy
\]  

To provide the explicit formulas for each q-PoI based on the MCDF function. Note that some formulas can be trivially extended to a more general case \(q > 2\).

---

7The definition of MCDF here is for general cases, and all the formulas of q-PoI are restricted to bivariate normal distributions without an additional specific statement in this paper.
**q-PoI all** For the sake of simplification, in this case, we again only present the formula for \( q = 2 \).

\[
q\text{-PoI all}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{2 \times 2}} (I(\hat{y}^{(1)} \text{ impr } \mathcal{P} \mathcal{F}) \cup (I(\hat{y}^{(2)} \text{ impr } \mathcal{P} \mathcal{F})) \xi_{M, \Sigma}(\hat{Y})d\hat{Y}
\]

\[
= \int_{y_1=(-\infty,-\infty)}^{(\infty,\infty)} \int_{y_2=(-\infty,-\infty)}^{(\infty,\infty)} (I(\hat{y}^{(1)} \text{ impr } \mathcal{P} \mathcal{F}) \cup (I(\hat{y}^{(2)} \text{ impr } \mathcal{P} \mathcal{F})) \xi_{M, \Sigma}(\hat{Y})d\hat{Y}
\]

\[
= \sum_{j=1}^{n+1} \sum_{j=1}^{m+1} \left( \int_{(l_1^{(j)}, 1)}^{(u_1^{(j)}, 1)} \xi_{\mu_1, \Sigma_1}(\hat{Y}_1)d\hat{Y}_1 \times \int_{(l_2^{(j)}, 2)}^{(u_2^{(j)}, 2)} \xi_{\mu_2, \Sigma_2}(\hat{Y}_2)d\hat{Y}_2 \right)
\]

\[
= \sum_{j=1}^{n+1} \sum_{j=1}^{m+1} \prod_{i=1}^{m} \left( I(\hat{y}_i^{(j)} \in \mathcal{P} \mathcal{F}) \xi_{\mu_i, \Sigma_i}(\hat{Y}_i)d\hat{Y}_i \right)
\]

(3-14)

For more general case of \( q > 2 \), the Eq. (3-14) can be derived from the same spirit presented above, while integrations need to permute all the combinations of the \( q \) entries \( \hat{y}^{(q)} \) over every rectangular stripe in \( \text{ndom}(\mathcal{P} \mathcal{F}) \).

**q-PoI one** The formula for \( q = 2 \), in this case, is the following

\[
q\text{-PoI one}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{2 \times 2}} (I(\hat{y}^{(1)} \text{ impr } \mathcal{P} \mathcal{F}) \cup (I(\hat{y}^{(2)} \text{ impr } \mathcal{P} \mathcal{F})) \xi_{M, \Sigma}(\hat{Y})d\hat{Y}
\]

\[
= \left( \sum_{j=1}^{q=2} \int_{\mathbb{R}^{2 \times 2}} I(\hat{y}^{(j)} \text{ impr } \mathcal{P} \mathcal{F}) \xi_{\mu^{(j)}, \Sigma^{(j)}}(\hat{Y}^{(j)})d\hat{Y}^{(j)} \right) - q\text{-PoI all}(M, \Sigma, \mathcal{P} \mathcal{F})
\]

(3-15)

Note that the above formula can not be directly extended to the case of \( q > 2 \). However, a more general formula can be derived from the basic calculations by permuting different combinations of the adjoint distribution of \( j \) different entries from the set \( \{ \hat{y}^{(j)} \} \) for \( j = 2 \) to \( j = q \). In Eq. (3-15), \( \sum_{j=1}^{q=2} \int_{\mathbb{R}^{2 \times 2}} I(\hat{y}^{(j)} \text{ impr } \mathcal{P} \mathcal{F}) \xi_{\mu^{(j)}, \Sigma^{(j)}}(\hat{Y}^{(j)})d\hat{Y}^{(j)} \) can be computed simply by \( 2 \times q\text{-PoI mean} \text{ with } q = 2 \) in Eq. (3-18).

**q-PoI best** According to the definition of q-PoI best in Eq. (3-7) and the property of integration, q-PoI best when \( q = 2 \) can be explicitly calculated as follows:

\[
q\text{-PoI best}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{2 \times q}} I(\text{DisC}(\hat{Y}) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y}
\]

\[
= \int_{Y_1=(-\infty,-\infty)}^{(\infty,\infty)} \int_{Y_2=(-\infty,-\infty)}^{(\infty,\infty)} (I(\hat{Y}_1 \lor \hat{Y}_2) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y}
\]

\[
= \sum_{j=1}^{n+1} \left( \int_{(l_1^{(j)}, 1)}^{(u_1^{(j)}, 1)} \xi_{\mu_1, \Sigma_1}(\hat{Y}_1)d\hat{Y}_1 \times \int_{(l_2^{(j)}, 2)}^{(u_2^{(j)}, 2)} \xi_{\mu_2, \Sigma_2}(\hat{Y}_2)d\hat{Y}_2 \right)
\]

\[
= \sum_{j=1}^{n+1} \sum_{j=1}^{m+1} \prod_{i=1}^{m} \left( \text{MCDF}((u_i^{(j)}, u_i^{(j)}), \mu_i, \Sigma_i) - \text{MCDF}((l_i^{(j)}, l_i^{(j)}), \mu_i, \Sigma_i) \right)
\]

(3-16)
Remark: When \( q > 2 \), the calculations for \( \text{q-PoI best} \) and \( \text{q-PoI worst} \) can be computed in a similar manner. One needs to consider all possible ranges of integration carefully. The idea of \( \text{q-PoI best} \) and \( \text{q-PoI worst} \) is to treat a joint multivariate normal distribution for each dimension and then multiply the PoI of each dimension. The Range of the integral in \( \text{q-PoI best} \) and \( \text{q-PoI worst} \) for each dimension are shown in Figure 3 by grey areas.

**q-PoI worst** Similarly, by using the definition of \( \text{q-PoI worst} \) in Eq. (3.8), \( \text{q-PoI worst} \) when \( q = 2 \) can also be explicitly computed:

\[
\text{q-PoI worst}(M, \Sigma, P,F) = \int_{\mathbb{R}_{2 \times q}} \left( I(\text{ConJ}(\hat{Y}) \text{ impr } P,F) \xi_{M,\Sigma}(\hat{Y}) \right) d\hat{Y}
\]

\[
= \sum_{i=1}^{n+1} \left\{ \left( \int_{l_i^{(1)},u_i^{(1)}}^{(u_i^{(j)},\infty)} + \int_{l_i^{(1)},u_i^{(1)}}^{(+\infty,u_i^{(j)})} \right) \xi_{\mu_i,\Sigma_i}(\hat{Y}_1) d\hat{Y}_1 \times \right. \\
\left. \int_{l_i^{(2)},u_i^{(2)}}^{(u_i^{(j)},\infty)} + \int_{l_i^{(2)},u_i^{(2)}}^{(+\infty,u_i^{(j)})} \right) \xi_{\mu_2,\Sigma_2}(\hat{Y}_2) d\hat{Y}_2
\]

\[
= \sum_{i=1}^{n+1} \prod_{j=2}^{m=2} \left\{ \left( \text{MCDF}(u_i^{(j)},\infty,\mu_i,\Sigma_i) + \text{MCDF}(\infty, u_i^{(j)},\mu_i,\Sigma_i) \right) \\
- \text{MCDF}(\infty, l_i^{(j)},\mu_i,\Sigma_i) + \text{MCDF}(l_i^{(j)},\infty,\mu_i,\Sigma_i) \right\}
\]

Notice that the last line in (3-17) contains the same component as (3-16) but with a subtract symbol.

**Remark:** When \( q > 2 \), the calculations for \( \text{q-PoI best} \) and for \( \text{q-PoI worst} \) can be computed in a similar manner. One needs to consider all possible ranges of integration carefully. The idea of \( \text{q-PoI best} \) and \( \text{q-PoI worst} \) is to treat a joint multivariate normal distribution \( \hat{Y} \) as a \( m \times 1 \) normal distribution \( \hat{Y}^t \) and calculate the PoI of the \( \hat{Y}^t \) in the best and worst senses as Definition (3-7) and (3-8) says, respectively. In \( \text{q-PoI best} \) and \( \text{q-PoI worst} \), we first calculate the PoI of a joint multivariate normal distribution for each dimension and then multiply the PoI of each dimension. The Range of the integral in \( \text{q-PoI best} \) and \( \text{q-PoI worst} \) for each dimension are shown in Figure 3 by grey areas.

**q-PoI mean** In this formulation, we consider the case \( q \geq 2 \).

\[
\text{q-PoI mean}(M, \Sigma, P,F) = \frac{1}{q} \sum_{i=1}^{q} \sum_{j=1}^{n+1} \int_{l_i^{(j)}}^{u_i^{(j)}} \xi_{\mu_i^{(j)},\Sigma_i^{(j)}} dy_1 \int_{l_i^{(j)}}^{u_i^{(j)}} \xi_{\mu_i^{(j)},\Sigma_i^{(j)}} dy_2
\]

**Remark:** The explicit formulas of the proposed five PoI variants can be easily extended into high-dimensional case. See the details in Appendix.
### 3.6 Computational Complexity in Bi-objective Case

Assuming the computation of a q-dimensional cumulative probability density function takes \( O(q) \) time units. The computational complexities of q-PoI best and q-PoI worst are bounded by \( O(2n \log n) = O(n \log n) \) for every \( q \), as both of them only need the evaluation of the MCDF on a \( \mathbb{R}^{2 \times 2} \) space. In practice, q-PoI worst needs more execution time as it evaluates four times more for the MCDF in every iteration, in comparison with q-PoI best. It is easy to conclude the computational complexity of q-PoI mean is \( O(qn \log n) \). The computational complexity of q-PoI all is \( O(n^q \log n) \) as the number of the combinations of \( q \) entries in \( Y \) is \( n^q \). Following the same idea, q-PoI one requires \( \sum_{j=1}^{q} {\binom{q}{j}} n^j = (n + 1)^q - 1 \) times calculations for q-dimensional cumulative probability density function. Therefore, the computational complexity of q-PoI one is \( O((n + 1)^q \log n) = O(n^q \log n) \).

**Remark:** In practice, a trick for reducing time complexity is to compute the integrals in the stripes that locate between \( \mu - 3\sigma \) and \( \mu + 3\sigma \). This trick is not used in the computational speed test part to scientifically analyze the ‘computational complexity’ but is used in experiments to save algorithms’ execution time.

### Computational Speed Test

Five different exact q-PoIs in Section 3.3 are assessed on CONVEXSPHERICAL and CONCAVESPHERICAL Pareto-front approximation sets [47]. The results are compared for validation with the MC integration in Algorithm 2. The MC method is allowed to run for 100,000 iterations. All the experiments are executed on the same computer: Intel(R) i7-4940MX CPU @ 3.10GHz, RAM 32GB. The operating system is Windows 10 (64-bit), and the software platform is MATLAB 9.9.0.1467703 (R2020b).

Table 1 shows the empirical speed experiments for the exact q-PoI method and the MC method. Both the exact calculation method and the MC method are executed without parallel computing\(^8\). Pareto front sizes are \( |P| \in \{10, 100, 1000\} \). The parameters in \( G \) are: \( M = [4 9 8 7]^T \) and \( M = [1 5 5 1]^T \) for CONVEXSPHERICAL and CONCAVESPHERICAL, respectively; the standard deviation matrix and covariance matrices \( \Sigma \) are used for both CONVEXSPHERICAL and CONCAVESPHERICAL, \( \mathcal{P} \):

\[
\Lambda = \begin{bmatrix}
 s_1^{(1)} & s_1^{(2)} \\
 s_2^{(1)} & s_2^{(2)}
\end{bmatrix}, \quad \Sigma = \begin{bmatrix}
 s_1^{(1)2} & \rho_1 s_1^{(1)} s_1^{(2)} \\
 \rho_1 s_1^{(1)} s_1^{(2)} & s_1^{(2)2}
\end{bmatrix}, \quad i \in \{1, 2\}, \quad \text{where} \quad \rho_1 = -\rho_2 = 0.5 \quad \text{and} \quad \rho = \text{Cov}(s^{(1)}, s^{(2)}) / (s^{(1)2} s^{(2)2} ).
\]

The average running time of ten repetitions is computed and shown in Table 1. The result confirms that q-PoI mean processes the lowest running time, which does not require covariance. On the other hand, both q-PoI one and q-PoI all require a large amount of execution time. The running time of these two q-PoIs is increased by a factor of \( |\mathcal{P}|^2 \). The running times of q-PoI worst and q-PoI best are confirmed with an increase of a factor \( |\mathcal{P}| \).

The comparison of q-PoI best, worst, and q-PoI mean indicates that the constant \( C \) of q-PoI best, worst is roughly 10 times of q-PoI mean because the CDF of multi-variate normal distribution (MCDF) requires more computational time than that of normal distribution’s CDF. Note that this study should not serve as a speed comparison, as the MC method is not precise.

### Accuracy Comparison

Figure 4 shows the randomly generated CONVEXSPHERICAL and CONCAVESPHERICAL Pareto fronts of \( |\mathcal{P}| = 100 \) for the 2-D case from [47] in the Fig. 4a, and the convergence figures of the MC integration of the five different

---

\(^8\) Any parallel technique can be utilized to speed-up execution times of exact calculation method and the MC method.
Figure 4: The studies of accuracy comparison between the Monte Carlo (MC) method and exact computation (ET), where $N_{MC}$ represents the number of iterations in the MC method. Note that the y-axis limit differs in (b), (c), (d), (e) and (f).

q-PoIs in the remaining subfigures. The parameters of the evaluated batch $\hat{Y}$ are mean matrices $M = [4 \ 9; \ 8 \ 7]^T$ and $M = [1 \ 5; \ 5 \ 1]^T$ for CONVEXSPHERICAL and CONCAVESPHERICAL Pareto-front approximation sets, respectively.
As reported in [26, 21], the position of $M$ leads to variance monotonic properties of PoI. Here we investigate similar behaviors for q-PoIs. When two points are considered in an objective space, there are three different cases: Case I – both two points in the mean matrix $M$ are located in the dominated space; Case II – both two points in $M$ are located in the non-dominated space; Case III – one point in $M$ is located in the non-dominated space, and the other one is located in the dominated space. Therefore, the behaviours of q-PoIs under varying covariance matrices $\Sigma$ in a batch $\hat{Y}$ are analyzed in three different mean matrices $M$, for $M = [1.5 \ 2.7; 2.5 \ 1.7]^T$, $M = [1.25 \ 1.25; 2.5 \ 0.75]^T$, $M = [1.5 \ 2; 3.5 \ 1.5]^T$ corresponding to Case I in Fig. 5a, Case II in Fig. 5b, and Case III in Fig. 5c, respectively. A Pareto approximation set is designed to be $\hat{F} = [1 \ 2.5; 2 \ 1.5; 3 \ 1]^T$. The standard deviation matrix is $\Lambda = [\alpha^2 \Sigma]$ and $\rho_1 = -\rho_2 = 0.5$. The results show that the q-PoI values based on the MC method are similar to the exact method after 50,000 iterations. However, the MC method requires more iterations to get a sufficiently accurate value.

### 3.7 Influence of Covariance Matrices on q-PoIs

In the left column, correlation coefficients are constant, but the standard deviation matrix $\Lambda$ is varied by multiplying a factor $\alpha = [0.1, 2.0]$ with stepsize 0.05. In the right column, where the $\Lambda$ is constant, the correlation coefficients $\rho_1 = \rho_2$ change from $-0.95$ to $0.95$ with a step size of $0.05$. The five q-PoIs are computed by both exact formulas (represented by black curves) and by the MC method (represented by red curves), where the number of iterations in the MC method is 1000. The first, the second, and the third rows are the corresponding results of Case I (Fig. 6a and 6b), Case II (Fig. 6c and 6d) and Case III (Fig. 6e and 6f), respectively.

In the left column of Figure 6, when both points in $M$ are dominated by the $\hat{F}$ (Case I), all q-PoIs increase w.r.t. a standard deviation matrix $\Lambda$, except for q-PoI $\text{worst}$, because a large variance of a point in the dominated space would increase the probability of a sampled point falling into a non-dominated space. For the reason of q-PoI $\text{worst}$ decreases in Case I is that $\text{ConJ}(M)$ dominates $\hat{F}$ when $\Lambda$ is a zero matrix. Therefore, an increasing $\Lambda$ decreases q-PoI $\text{worst}$. In Case II, when both points in $M$ are not dominated by $\hat{F}$, all q-PoIs decrease w.r.t. a standard deviation matrix $\Lambda$, because points in $M$ are located in the non-dominated space of $\hat{F}$, and a large variance will lead to widespread distributions of $\hat{Y}$. In Case III, each q-PoI either decreases or increases w.r.t $\Lambda$, except for q-PoI $\text{one}$, of which the curve is convex. This is reasonable because a point in $M$ dominates the $\hat{F}$, and another point in $M$ is dominated by the $\hat{F}$. The distributions of the left red and right red points in Fig. 5c cover more dominated and non-dominated space, respectively, when $\Lambda$ increases. Since q-PoI $\text{one}$ is defined as the PoI of at least one point dominates a $\hat{F}$, there is a trade-off balancing between the effects of two multivariate normal distributions. Therefore, a stationary point regarding $\Lambda$ exists in q-PoI $\text{worst}$ and q-PoI $\text{one}$, the difference between these two q-PoIs can be clearly distinguished when at least one point in $\hat{Y}$ is located close to $\hat{F}$ (i.e., Case I and Case III).

In the right column of Figure 6, we find that q-PoI $\text{worst}$ and q-PoI $\text{one}$ decrease w.r.t. $\rho_1, \rho_2$, while q-PoI $\text{best}$ and q-PoI $\text{all}$ increases in all the three cases. Unsurprisingly, a correlation coefficient does not influence q-PoI $\text{mean}$. Note
Figure 6: The behavior studies of five different q-PoIs w.r.t. standard deviation matrix $\Sigma$ and correlation coefficient $\rho$ in three different cases.

that the values of all the q-PoIs at $\rho = 0$0.5cm(118,436),(292,463) represent their corresponding q-PoI values without correlations. The influence of $\rho$ on q-PoI all and on q-PoI best is positively correlated because a positive covariance leads the coordinate values of two solutions closer, and vice versa. One can also conclude that $\rho$ has a negative influence on q-PoI one since Eq. (3-15) shows that $\rho$ is negatively correlated.

Both q-PoI all and q-PoI best have a stricter requirement in their definitions than the other three q-PoIs. Compared with q-PoI all, q-PoI best has a more ‘relaxed-exploration’ property and a greedy characteristic because of its strict requirement in the definition. This is why the values of q-PoI best are always smaller than that of q-PoI all. Comparing q-PoI worst and q-PoI one, q-PoI worst enhances exploration caused by a more relaxed requirement of $I(\cdot)$ by its definition\(^{9}\). This explains why the values of q-PoI worst are larger than those of q-PoI one.

\(^{9}\)PoI’s explorative property can be also enhanced by introducing an $\epsilon$-improvement with a carefully designed [26]. The proposed k-PoI can also incorporate the $\epsilon$ strategy for the same purpose.
4 Empirical Experiments

4.1 Benchmarks

In this paper, experiments on 20 artificial bi-objective test problems are performed, including ZDT1-3 [48], GSP-1 with $\gamma = 0.4$, GSP-2 [49] with $\gamma = 1.8$, MaF1, MaF5, and MaF12 [50], mDTLZ1-4 [51], WOSGZ1-8 [52]. The dimensions of decision space are 3, 5, 10, and 15 for GSP1-2, ZDT1-3, mDTLZ1-4, and WOSGZ1-8, respectively. The reference points are [11, 11] for ZDT1-3, [1.2, 1.2] for mDTLZ1-4 and WOSGZ1-8, and [5, 5] for the other problems.

4.2 Algorithm Configuration

The five proposed q-PoIs are compared with other indicator-based MOBGO algorithms (original PoI, two parallel techniques of PoI by using Kriging Believer and Constant Liar with a ‘mean’ liar strategy in Alg. 1, and q-EHVI [22]), two state-of-the-art multiple-point MOBGO (ParEGO [53], MOEA/D-EGO [13]), and three recently proposed multiple-point surrogated-assisted multi-objective optimization algorithms (TSEMO [54], DGEMO [16], and MOEA/D-ASS [18]). The platform of TSEMO, ParEGO, MOED/D-EGO, and DGEMO is Python, and the platform of the other algorithms is MATLAB in this paper.\(^{10}\)

In all the experiments, the number of DoE ($\eta$) is $\min\{6 \times d, 60\}$ and the maximum function evaluation is $T_c = \min\{\eta \times 9, \eta + (200 - \eta) \times 2\}$. In indicator-based MOBGO algorithms, the acquisition function is optimized by CMA-ES (of 1 restart and at most 2000 iterations) to search for optimal $X^*$ due to its favorable performance on BBOB function testbed [55]. The optimizers of the other algorithms in this paper are NSGA-II, CMA-ES, MOEA/D, GA, and a so-called ‘discovery optimization’ [17], respectively, in TSEMO [56], a batch-version ParEGO [16], MOEA/D-EGO [13], MOEA/D-ASS [18], and DGEMO [16]. The maximum iteration of NSGA-II and MOEA/D is $2 \times 2000$. Among all the optimizers, the ‘discovery optimization’ [17] is the only optimization algorithm that requires the gradient and the Hessian matrix of the predictions of Gaussian Processes.

4.3 Experimental Results

We evaluate the Pareto-front approximation sets in this section using the HV indicator. In the experimental studies, Wilcoxon’s rank-sum test at a 0.05 significance level was implemented between an algorithm and its competitors to test the statistical significance. In the following tables, "+", "≈", and "−" denote that an algorithm in the first row performs better than, worse than, and similar to its competitors in the first column, respectively.

\(^{10}\)The Python source code is available on https://github.com/yunshengtian/DGEMO, and the source code of MOEA/D-ASS is available on https://github.com/ZhenkunWang/MOEAD-ASS
| Problem | HV | GSP-1 | GSP-2 | MaFi | MaFi2 |
|---------|---------|---------|---------|--------|--------|
|         |         |         |         |        |        |
| ZDT1    |         |         |         |        |        |
| min     | 115.14  | 116.81  | 116.19  | 110.4  | 12.29  |
| max     | 115.54  | 127.02  | 122.25  | 111.35 | 13.32  |
| median  | 115.34  | 124.36  | 120.39  | 110.4  | 12.77  |
| mean    | 115.34  | 124.08  | 120.15  | 110.0  | 12.77  |
| std     | 0.03    | 4.56    | 2.82    | 0.24   | 1.18   |
| Pol     | 120.00  | 124.00  | 120.00  | 120.00 | 120.00 |
| q-Pol_al | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_all | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_con | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_max | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Pol_best | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PB-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Cl-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-EHVI | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| TSEMO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| DSGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-EGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-ASS | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PurEGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Problem | HV | GSP-1 | GSP-2 | MaFi | MaFi2 |
|---------|---------|---------|---------|--------|--------|
|         |         |         |         |        |        |
| ZDT2    |         |         |         |        |        |
| min     | 116.03  | 116.57  | 113.55  | 113.55 | 112.83 |
| max     | 116.42  | 127.21  | 123.45  | 123.45 | 123.45 |
| median  | 116.22  | 126.86  | 122.8  | 122.8  | 122.8  |
| mean    | 116.22  | 126.86  | 122.8  | 122.8  | 122.8  |
| std     | 0.15    | 4.66    | 2.82   | 0.24   | 1.23   |
| Pol     | 120.00  | 124.00  | 120.00  | 120.00 | 120.00 |
| q-Pol_al | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_all | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_con | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_max | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Pol_best | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PB-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Cl-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-EHVI | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| TSEMO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| DSGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-EGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-ASS | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PurEGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Problem | HV | GSP-1 | GSP-2 | MaFi | MaFi2 |
|---------|---------|---------|---------|--------|--------|
|         |         |         |         |        |        |
| ZDT3    |         |         |         |        |        |
| min     | 116.03  | 116.63  | 113.55  | 113.55 | 112.83 |
| max     | 116.42  | 127.21  | 123.45  | 123.45 | 123.45 |
| median  | 116.22  | 126.86  | 122.8  | 122.8  | 122.8  |
| mean    | 116.22  | 126.86  | 122.8  | 122.8  | 122.8  |
| std     | 0.15    | 4.66    | 2.82   | 0.24   | 1.23   |
| Pol     | 120.00  | 124.00  | 120.00  | 120.00 | 120.00 |
| q-Pol_al | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_all | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_con | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-Pol_max | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Pol_best | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PB-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| Cl-Pol | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| q-EHVI | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| TSEMO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| DSGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-EGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| MEOAD-ASS | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
| PurEGO | 120.00 | 124.00 | 120.00 | 120.00 | 120.00 |
Table 2: Empirical Comparisons w.r.t HV (Continued).

| Problem | HV | min | max | mean | std | Pol | q-Pol_all | q-Pol_ecc | q-Pol_worst | q-Pol_mean | KB-Pol | CL-Pol | q-EHVI | TSEMO | DGEMO | MOEA/D-EGO | MOEA/D-ASS | PaXEGO |
|---------|----|-----|-----|------|-----|-----|----------|----------|-------------|-----------|--------|--------|--------|-------|-------|-----------|-----------|--------|
| mDTLZ1  |    |     |     |      |     | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| mDTLZ2  |    |     |     |      |     | 0.18247 | 1.17364 | 1.20386 | 1.13358 | 1.16227 | 1.17192 | 1.19177 | 1.19466 | 1.12891 | 1.06826 | 1.22216 | 1.06720 | 1.19868 | 1.11915 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| mDTLZ3  |    |     |     |      |     | 0.19715 | 1.18787 | 1.21162 | 1.17667 | 1.17866 | 1.19107 | 1.20662 | 1.20599 | 1.14050 | 1.10199 | 1.22257 | 1.11898 | 1.20429 | 1.12973 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| mDTLZ4  |    |     |     |      |     | 0.19424 | 1.18143 | 1.20772 | 1.16128 | 1.17038 | 1.19125 | 1.20686 | 1.20076 | 1.13422 | 1.08040 | 1.22242 | 1.08276 | 1.20316 | 1.12274 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| WOSGZ1  |    |     |     |      |     | 0.14987 | 0.22788 | 0.62268 | 0.60772 | 0.39301 | 0.04799 | 0.06076 | 0.04789 | 0.04789 | 0.21700 | 0.59513 | 0.00115 | 0.54565 | 0.30253 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| WOSGZ2  |    |     |     |      |     | 0.15064 | 0.66733 | 0.70684 | 0.75891 | 0.62987 | 0.44427 | 0.24908 | 0.21974 | 0.40188 | 0.47250 | 0.64282 | 0.54772 | 0.70863 | 0.60599 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| WOSGZ3  |    |     |     |      |     | 0.14944 | 0.10122 | 0.67170 | 0.66779 | 0.54108 | 0.10835 | 0.06076 | 0.14587 | 0.40027 | 0.61511 | 0.42960 | 0.65388 | 0.52181 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| WOSGZ4  |    |     |     |      |     | 0.15006 | 0.55571 | 0.67182 | 0.66795 | 0.35847 | 0.15833 | 0.10475 | 0.19488 | 0.39087 | 0.67199 | 0.38699 | 0.63501 | 0.51989 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |

PREPRINT - JULY 9, 2022
Table 2: Empirical Comparisons w.r.t HV (Continued).

| Problem | HV | Pol | q-Pol | q-Pol, one | q-Pol, best | q-Pol, worst | q-Pol, mean | KB-Pol | CL-Pol | q-EHVI | TSEMO | DGEMO | MOEA/D-EGO | MOEA/D-ASS | ParEGO |
|---------|----|-----|-------|------------|-------------|---------------|-------------|--------|--------|--------|-------|-------|------------|------------|--------|
| WOSGZ4  | min | 0.11851 | 0.14308 | 0.24885 | 0.26282 | 0.37081 | 0.30107 | 0.26690 | 0.14300 | 0.17616 | 0.03116 | 0.80794 | 0.47208 | 0.62254 | 0.13715 |
|         | median | 0.26133 | 0.21919 | 0.25482 | 0.39850 | 0.49187 | 0.49192 | 0.47525 | 0.44972 | 0.50380 | 0.30116 | 0.80794 | 0.47208 | 0.62254 | 0.13715 |
|         | mean | 0.72973 | 0.20414 | 0.39036 | 0.07906 | 0.03901 | 0.06226 | 0.06980 | 0.10242 | 0.08855 | 0.00004 | 0.00983 | 0.11508 | 0.07814 | 0.02368 |
| WOSGZ5  | min | 0.00000 | 0.19230 | 0.23698 | 0.08040 | 0.09403 | 0.00000 | 0.00000 | 0.00000 | 0.20199 | 0.00000 | 0.09721 | 0.17512 | 0.27686 | 0.00000 |
|         | median | 0.40094 | 0.41823 | 0.43869 | 0.41707 | 0.19333 | 0.30283 | 0.39668 | 0.14773 | 0.75531 | 0.40134 | 0.10763 | 0.30635 | 0.00000 |
|         | mean | 0.79021 | 0.28674 | 0.24757 | 0.18488 | 0.22081 | 0.23251 | 0.13908 | 0.00000 | 0.73486 | 0.30968 | 0.10763 | 0.30635 | 0.00000 |
| WOSGZ6  | min | 0.00000 | 0.13158 | 0.27422 | 0.17468 | 0.21386 | 0.01753 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
|         | median | 0.30735 | 0.35878 | 0.44799 | 0.45358 | 0.41707 | 0.19333 | 0.30283 | 0.39668 | 0.14773 | 0.75531 | 0.40134 | 0.10763 | 0.30635 |
|         | mean | 0.72618 | 0.54088 | 0.52963 | 0.52963 | 0.21953 | 0.15313 | 0.20199 | 0.16600 | 0.01036 | 0.73498 | 0.30968 | 0.10763 | 0.30635 |
| WOSGZ7  | min | 0.00000 | 0.04630 | 0.10639 | 0.04497 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 |
|         | median | 0.14578 | 0.22187 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 | 0.35448 |
|         | mean | 0.84741 | 0.05325 | 0.10639 | 0.04497 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 | 0.09294 |
| WOSGZ8  | min | 0.00000 | 0.42719 | 0.58854 | 0.76026 | 0.74336 | 0.76379 | 0.54815 | 0.58127 | 0.59457 | 0.00000 | 1.20256 | 0.49690 | 0.84191 | 0.07449 |
|         | median | 0.77579 | 0.09163 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 | 0.91561 |
|         | mean | 0.56930 | 0.81753 | 0.81896 | 0.85445 | 0.84356 | 0.78780 | 0.75715 | 0.76224 | 0.84950 | 0.01589 | 1.21432 | 0.70529 | 0.91844 | 0.24869 |
|         | std. | 0.60000 | 0.80795 | 0.82062 | 0.84928 | 0.84741 | 0.75319 | 0.78084 | 0.82227 | 0.05353 | 1.21488 | 0.71125 | 0.93911 | 0.25920 | 0.18515 |
The results of all the test algorithms are summarized in Table 2. Since all the test algorithms failed to locate a Pareto-front approximation set that dominates the reference point on mDTLZ1 and mDTLZ3 problems, the results of these two problems are not visualized or counted in the pairwise Wilcoxon’s Rank-Sum test. From Table 2, it can be observed that DGEMO and MOEA/D-ASS yield the best and the second best results among all the test algorithms w.r.t. the mean and the standard deviation (std.) of HV values. Between DGEMO and MOEA/D-ASS, DGEMO outperforms MOEA/D-ASS w.r.t. mean HV because DGEMO incorporates the diversity knowledge from both design and objective spaces in the batch selection. Additionally, the DGEMO’s optimizer is based on a first-order approximation of the Pareto front, which can discover piecewise continuous regions of the Pareto front rather than individual points on the Pareto front to be captured [17].

Table 3 show the performance of pairwise Wilcoxon’s Rank-Sum test (+/≈/−) matrix among all indicator-based MOBGO algorithms test algorithms on 18 benchmarks. The sum of Wilcoxon’s Rank-Sum test (sum of +/≈/−) indicates q-PoI one performs best, as it significantly outperforms 78 pairwise instances between algorithms and problems. Additionally, the PoI variants (q-PoI, i ∈ all, one, best, worst) that consider correlations between multiple point predictions outperform PoI and the other two parallel techniques of PoI (KB-PoI and CL-PoI) in most cases.

Table 2 and Table 4 show that q-EHVI performs best on low-dimensional problems (ZDT1-3, GSP problems, and MaF problems) among all the indicator-based MOBGO algorithms. q-PoI worst yields the second best results on the low-dimensional test problems. On high-dimensional problems (mDTLZ and WOSGZ problems) of which boundaries of the Pareto fronts are difficult-to-approximate (DtaA), q-PoI one yields the best results w.r.t. mean HV (see Table 2) and the pairwise Wilcoxon’s Rank-Sum test in Table 5. Comparing all the indicator-based MOBGO algorithms, the poor performance of q-EHVI on high-dimensional problems is explained as follows. Compared to q-EHVI, the objective area involved in the computation of PoI and its variants is larger. The computation of HV-based acquisition functions only covers the non-dominated space that dominates the reference point. Otherwise, the HV will be infinity. Introducing a reference point makes it impossible to explore boundary non-dominated solutions dominated by the reference point, that is, the space ndom(\(P^F, \infty^m\)) \ ndom(\(P^F, x\)). On the other hand, PoI and its variants don’t have this limitation, as their computations cover an entire non-dominated space. Therefore, PoI and its variants are easier to locate the boundary non-dominated solutions for problems with DtaA Pareto-front boundaries. Another possible reason is that the computational error caused by the MC method can deteriorate the performance of the CMA-ES optimizer.

Figure 7 exhibits the average HV convergence curves of 15 independent runs of 14 algorithms on the 18 test problems. At the beginning optimization stage, DGEMO converges much faster than the other algorithms in low-dimensional problems but converges slowly in high-dimensional problems. This is because it is more difficult to quantify a credible diversity knowledge in both design and objective spaces for problems with DtaA boundaries when the number of samples is small. The convergence of q-EHVI is fast on low-dimensional problems but is slow on high-dimensional problems. The reason is mainly because of its greedy property in theory, compared with PoI and q-PoIs. Among all the indicator-based MOBGO algorithms, q-PoI worst converges second fast in low-dimensional problems, and q-PoI one converges fastest in high-dimensional problems. The reason relates to the strictness of the acquisition function. The more strict condition to fulfill is, the more greedy the acquisition function will be, and vice versa. In low-dimensional problems, q-PoI best is the most greedy due to its strict requirement to fulfill. However, this greedy strategy is efficient on low-dimensional problems. Therefore, q-PoI all and q-PoI best converge much faster than the other indicator-based MOBGO algorithms on low-dimensional test problems (see Figure 7a). When the test problem is complex w.r.t. the number of decision variables and the property of DtaA PF boundaries, the exploration acquisition function is more effective as it is easier to jump out of the local optima. This is also the reason why q-PoI worst converges the fastest on WOSGZ1 problem (see Figure 7b).

Figure 8, Figure 9 and Figure 10 show the best, median, and worst empirical attainment curves of the Pareto-front approximation set on ZDT3, mDTLZ4, and WOSGZ7, respectively, by using the empirical first-order attainment function in [57]. The ranges of \(f_1\) and \(f_2\) are the same for all the algorithm’s attainment curves on a specific problem. On the discontinued ZDT3 problem (see Figure 8), DGEMO finds all the Pareto fronts due to the utilization of diversity knowledge from both design and objective spaces. The best attainment curve of q-PoI worst finds more Pareto fronts among all the indicator-based MOBGO algorithms because its requirement is most relaxed, and it is much more explorative than the other acquisition functions. In Figure 9, it is easy to observe that the acquisition functions that incorporate correlation information yield better results than the other acquisition functions that don’t use the correlation information. WOSGZ7 problem is more difficult than WOSGZ1-6 because of the imbalance between the middle and the boundary regions of the Pareto front [52]. This problem entails more difficulty for optimization algorithms in fulfilling breadth diversity. Thus, an optimization algorithm with more exploration-property works better on WOSGZ7, which is the reason why q-PoI one and q-PoI worst work much better than the other indicator-based algorithms and DGEMO. The explanation for the poor performance of DGEMO on WOSGZ7 is that credible diversity knowledge of this kind of problem requires more samples. This also explains the reason for the fast convergence of DGEMO at the end of the optimization stage on WOSGZ7 in Figure 7b.
Figure 7: HV convergence plots of 18 algorithms on the benchmarks are visualized in terms of the average HV over 15 independent runs.
Figure 8: The best, median, and the worst empirical attainment curves on ZDT3, where $f_1$, $f_2$, and $f_3$ represent the best, the worst, and the median Pareto-front approximation sets over 15 independent runs, respectively.
Figure 9: The best, median, and the worst empirical attainment curves on mDTLZ4, where $f_1$, $f_2$, and $f_3$ represent the best, the worst, and the median Pareto-front approximation sets over 15 independent runs, respectively.
Figure 10: The best, median, and the worst empirical attainment curves on WOSGZ7, where , and represent the best, the worst, and the median Pareto-front approximation sets over 15 independent runs, respectively.
Table 3: The pairwise Wilcoxon’s Rank-Sum test (+/ ≈ / −) matrix at a 0.05 significance level was performed among nine indicator-based MOBGO algorithms, where algorithms in all the columns (except the first column) are compared with the algorithms in the first column.

|        | Pol       | q-Pol all | q-Pol one | q-Pol best | q-Pol worst | q-Pol mean | KB-Pol   | CL-Pol   | q-EHVI   |
|--------|-----------|-----------|-----------|------------|-------------|------------|----------|----------|----------|
| Pol    | 0/0/0     | 9/6/3     | 14/1/3    | 11/5/2     | 12/2/4      | 9/6/3      | 8/9/1    | 9/8/1    | 12/4/2   |
| q-Pol all | 3/7/8     | 0/0/0     | 8/5/5     | 8/6/4      | 7/4/7       | 5/7/6      | 5/5/8    | 5/6/7    | 6/7/5    |
| q-Pol one | 3/1/14    | 5/5/8     | 0/0/0     | 6/5/7      | 2/9/7       | 3/7/8      | 3/3/12   | 3/3/12   | 5/5/8    |
| q-Pol best | 2/5/11    | 4/6/8     | 7/5/6     | 0/0/0      | 7/4/7       | 3/8/7      | 2/5/11   | 3/6/9    | 5/6/7    |
| q-Pol worst | 4/2/12    | 8/3/7     | 7/9/2     | 7/4/7      | 0/0/0       | 6/4/8      | 5/3/10   | 4/4/10   | 8/3/7    |
| q-Pol mean | 4/5/9     | 6/7/5     | 8/7/3     | 7/8/3      | 9/3/6       | 0/0/0      | 5/8/5    | 4/7/7    | 7/7/4    |
| KB-Pol  | 1/9/8     | 9/4/5     | 12/3/3    | 11/5/2     | 10/3/5      | 5/8/5      | 0/0/0    | 3/12/3   | 8/7/3    |
| CL-Pol  | 1/8/9     | 7/6/5     | 14/1/3    | 9/6/3      | 10/4/4      | 7/7/4      | 3/13/2   | 0/0/0    | 8/6/4    |
| q-EHVI  | 2/4/12    | 5/7/6     | 8/5/5     | 7/6/5      | 7/3/8       | 4/7/7      | 3/7/8    | 4/7/7    | 0/0/0    |

Sum of +/ ≈ / − | 20/41/83 | 53/44/47 | **78/36/30** | 66/45/33 | 64/32/48 | 42/54/48 | 34/53/57 | 35/53/56 | 59/45/40 |
5 Conclusion

This paper proposed five alternative acquisition functions by generalizing PoI from a single point into a batch with multiple points. For each proposed q-PoI, explicit computational formulas and the MC method for approximation are provided. Computational complexities of the exact computational formula are given. Among the five proposed q-PoIs, q-PoI one processes the highest computational complexity as it has to compute the sum of PoI for each single solution and q-PoI all, while q-PoI mean only considers the diagonal elements in covariance matrix $\Sigma$. All the proposed q-PoI leave spaces for parallel techniques in MOBGO to evaluate multiple solutions in each batch simultaneously. Some of the behaviors of q-PoIs and their connections to the standard deviation matrix and correlation coefficient are analyzed for the five proposed q-PoI variants in three cases.

This paper also compares the performance of the five proposed acquisition functions with that of PoI, two parallel techniques for PoI, q-EHVI, and five other batch-selection surrogate-assisted multi-objective optimization algorithms on 20 benchmarks. The experiments show that the acquisition functions with more greedy characteristics (q-EHVI, q-PoI all, and q-PoI best) perform much better than the other acquisition functions on low-dimensional test problems. For high-dimensional problems, the acquisition functions (q-PoI one and q-PoI worst) with more explorative properties outperform the other acquisition functions, especially on the problems with DTA Pareto front boundaries. DGEMO outperforms the other test algorithms, but its convergence is slower than most of the proposed q-PoI variants in this paper on the problems with DTA Pareto front boundaries.

The experimental results confirm that strict requirements in q-Pois’ definitions can refine algorithms’ exploitation property. Due to the strict condition in q-PoI best and q-PoI all, the algorithm behaves greedily. This exploitative behavior allows MOBGO to quickly converge to the true Pareto front at the early optimization stage. Still, it renders MOBGO performance when the solutions of the Pareto front are highly biased or the problem’s landscape has DTA PF boundaries. On the other hand, relaxed requirements of q-PoI one and q-PoI worst can enhance the exploration property of PoI, which purely focuses on exploration. Strategically, one can use strict acquisition functions q-PoI best at the early stage of iteration and then switch to a relaxed acquisition function q-PoI one or q-PoI worst when the HV convergence velocity starts to slow down during the optimization processes on low-dimensional problems. On high-dimensional problems, especially those with DTA Pareto front boundaries, it is recommended to use q-PoI one or q-PoI worst.

The proposed algorithms can be applied to many real-world applications involving expensive simulations, including box-type boom designing problem [58], material flow optimization problems [59], algorithm selection problems [60], and hyperparameter tuning problems in the machine learning field, just to name a few. For future work, it is worthwhile to study the multi-objective acquisition functions that consider the correlations among each coordinate using the multi-output Gaussian process, as all of the existing multi-objective acquisition functions simply assume the independence among each objective. It is also interesting to incorporate the diversity control mechanism in indicator-based MOBGO algorithms due to its effectiveness shown in DGEMO. An intuitive solution is to use truncated normal distribution in PoI and its variants to split objective space into several subregions and then search for the optima in each subregion.

Appendix

A.1 Explicit formulas of PoIs for high-dimensional problems

Note that the explicit formulas for computing q-PoIs in the case of $m$ objectives for $m > 2$ can be easily generalized from the definition and the formulas for $m = 2$ provided in Section 3.3. For completeness, we list the formulas for the case $m > 2$:

$$q-PoI_{all}(M, \Sigma, \mathcal{F}) = \int_{\mathbb{R}^{m\times 2}} I(\hat{y}^{(1)} \text{ impr} \mathcal{F}) \cap I(\hat{y}^{(2)} \text{ impr} \mathcal{F}) \xi_M, \Sigma(\hat{Y}) d\hat{Y}$$

$$= \sum_{j=1}^{N_m} \sum_{j=1}^{N_m} \left( \int_{(u_1^{(j)}, u_2^{(j)})} \xi_{\mu_1, \Sigma_1}(\hat{Y}_1) d\hat{Y}_1 \times \cdots \times \int_{(u_1^{(j)}, u_2^{(j)})} \xi_{\mu_m, \Sigma_m}(\hat{Y}_m) d\hat{Y}_m \right)$$

$$= \sum_{j=1}^{N_m} \sum_{j=1}^{N_m} \prod_{i=1}^{m} \Gamma(i_i^{(j)}, u_i^{(j)}, p_i^{(jj)}, \mu_i, \Sigma_i)$$

(5-1)
\[ \text{q-PoI}_{\text{one}}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{m \times 2}} I(\hat{y}(1) \text{ impr } \mathcal{P} \mathcal{F}) \cup I(\hat{y}(2) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y} \]

\[ = \sum_{j=1}^{q=2} \int_{\mathbb{R}^{m}} I(\hat{y}(j) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{\mu_{j}, \Sigma_{j}}(\hat{y}(j))d\hat{y}(j) - \text{q-PoI}_{\text{all}}(M, \Sigma, \mathcal{P} \mathcal{F}) \]  

(5-2)

\[ \text{q-PoI}_{\text{worst}}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{m \times 2}} I(\text{ConJ}(\hat{Y}) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y} \]

\[ = \sum_{j=1}^{N_{m}} \left( \int_{(u_{1}^{(j)}, +\infty)} + \int_{(t_{1}^{(j)}, l_{1}^{(j)})} \right) \xi_{\mu_{j}, \Sigma_{j}}(\hat{Y}(j))d\hat{Y}_{1} \times \cdots \times \left( \int_{(u_{m}^{(j)}, +\infty)} + \int_{(t_{m}^{(j)}, l_{m}^{(j)})} \right) \xi_{\mu_{m}, \Sigma_{m}}(\hat{Y}_{m})d\hat{Y}_{m} \]

\[ = \sum_{j=1}^{N_{m}} \prod_{i=1}^{m} \left( \text{MCDF}((u_{i}^{(j)}, \infty), \mu_{i}, \Sigma_{i}) + \text{MCDF}((\infty, u_{i}^{(j)}), \mu_{i}, \Sigma_{i}) \right) \]

\[ - \left( \text{MCDF}((\infty, t_{i}^{(j)}), \mu_{i}, \Sigma_{i}) + \text{MCDF}((t_{i}^{(j)}, \infty), \mu_{i}, \Sigma_{i}) \right) \]

\[ - \left( \text{MCDF}((u_{i}^{(j)}, t_{i}^{(j)}), \mu_{i}, \Sigma_{i}) - \text{MCDF}((t_{i}^{(j)}, u_{i}^{(j)}), \mu_{i}, \Sigma_{i}) \right) \]  

(5-3)

\[ \text{q-PoI}_{\text{best}}(M, \Sigma, \mathcal{P} \mathcal{F}) = \int_{\mathbb{R}^{m \times 2}} I(\text{DisC}(\hat{Y}) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y} \]

\[ = \int_{Y_{1} = (-\infty, -\infty)} \cdots \int_{Y_{m} = (-\infty, -\infty)} \left( I(\hat{Y}(1) \lor \cdots \lor \hat{Y}(m) \text{ impr } \mathcal{P} \mathcal{F}) \xi_{M, \Sigma}(\hat{Y})d\hat{Y} \right) \]

\[ = \sum_{j=1}^{N_{m}} \prod_{i=1}^{m} \left( \text{MCDF}((u_{i}^{(j)}, t_{i}^{(j)}), \mu_{i}, \Sigma_{i}) - \text{MCDF}((t_{i}^{(j)}, u_{i}^{(j)}), \mu_{i}, \Sigma_{i}) \right) \]  

(5-4)

\[ \text{q-PoI}_{\text{mean}}(M, \Sigma, \mathcal{P} \mathcal{F}) = \frac{1}{q} \sum_{q=1}^{q} \sum_{j=1}^{N_{m}} \int_{(u_{1}^{(j)}, l_{1}^{(j)})} \xi_{\mu_{1}^{(j)}, \Sigma_{1}^{(j)}} dy_{1} \cdots \int_{(u_{m}^{(j)}, l_{m}^{(j)})} \xi_{\mu_{m}^{(j)}, \Sigma_{m}^{(j)}} dy_{m} \]  

(5-5)

In the formulas above, \( N_{m} \) is the number of decomposed non-dominated areas in \( m \)-dimensional objective space. By using the methods in [37], this number can be reduced into \( n+1 \) and \( 2n+1 \) for bi- and tri-objective spaces, respectively. Please also note that all the q-PoIs formulas for high-dimensional objective spaces in this appendix assume that no overlapped stripes/cells/boxes exist in a dominated space. Therefore, the grid decomposition method and the decomposition methods for \( m = 2, 3 \) in [37] work on these q-PoI formulas. However, it is not recommended to use the grid decomposition method due to its high computational complexity introduced. The decomposition method for \( m \geq 4 \) mentioned in [37] cannot be directly applied to the q-PoI formulas in this appendix. One has to subtract the integral of the overlapped cells by using the decomposition method that exists in overlapped cells.

A.2 Tables

References

[1] P. Fleck, D. Entner, C. Münzer, M. Kommenda, T. Prante, M. Schwarz, M. Hächl, M. Affenzeller, Box-Type Boom Design Using Surrogate Modeling: Introducing an Industrial Optimization Benchmark, Springer International Publishing, Cham, 2019, pp. 355–370. doi:10.1007/978-3-319-89890-2_23.
URL https://doi.org/10.1007/978-3-319-89890-2_23

27
Table 4: Low-dimensional case (ZDT1-3, GSP1-2, MaF1/5/12): The pairwise Wilcoxon’s Rank-Sum test (+/ ≈ /−) matrix at a 0.05 significance level was performed among nine indicator-based MOBGO algorithms, where algorithms in all the columns (except the first column) are compared with the algorithms in the first column.

|        | PoI  | q-PoI_all | q-PoI_one | q-PoI_best | q-PoI_worst | q-PoI_mean | KB-PoI | CL-PoI | q-EHVI |
|--------|------|-----------|-----------|------------|-------------|------------|--------|--------|--------|
| PoI    | 0/0/0| 4/3/1     | 4/1/3     | 5/2/1      | 4/1/3       | 4/2/2      | 4/4/0  | 4/4/0  | 7/0/1  |
| q-PoI_all | 1/4/3 | 0/0/0     | 0/3/5     | 5/2/1      | 0/3/5       | 2/3/3      | 3/3/2  | 5/2/1  |
| q-PoI_one | 3/1/4 | 5/3/0     | 0/0/0     | 6/1/1      | 1/3/4       | 3/4/1      | 3/3/2  | 5/2/1  |
| q-PoI_best | 1/2/5 | 1/2/5     | 1/1/6     | 0/0/0      | 1/1/6       | 1/3/4      | 1/2/5  | 1/2/5  | 4/1/3  |
| q-PoI_worst | 3/1/4 | 5/3/0     | 4/3/1     | 6/1/1      | 0/0/0       | 5/1/2      | 4/1/3  | 3/3/2  | 8/0/0  |
| q-PoI_mean | 2/2/4 | 3/3/2     | 1/4/3     | 4/3/1      | 3/0/5       | 0/0/0      | 4/2/2  | 2/3/3  | 6/0/2  |
| KB-PoI | 0/4/4 | 2/3/3     | 3/2/3     | 5/2/1      | 3/1/4       | 2/4/2      | 0/0/0  | 2/4/2  | 5/1/2  |
| CL-PoI | 0/4/4 | 2/3/3     | 4/1/3     | 5/2/1      | 2/3/3       | 3/3/2      | 2/5/1  | 0/0/0  | 5/1/2  |
| q-EHVI | 1/0/7 | 1/2/5     | 1/2/5     | 3/1/4      | 0/0/8       | 4/2/2      | 2/3/3  | 1/2/5  | 0/0/0  |

Sum of +/ ≈ /−: 11/18/35 23/22/19 18/17/29 39/14/11 14/12/38 22/18/24 23/20/21 20/22/22 45/7/12

Table 5: High-dimensional case (mDTLZ1-4, WOSGZ1-8): The pairwise Wilcoxon’s Rank-Sum test (+/ ≈ /−) matrix at a 0.05 significance level was performed among nine indicator-based MOBGO algorithms, where algorithms in all the columns (except the first column) are compared with the algorithms in the first column.

|        | PoI  | q-PoI_all | q-PoI_one | q-PoI_best | q-PoI_worst | q-PoI_mean | KB-PoI | CL-PoI | q-EHVI |
|--------|------|-----------|-----------|------------|-------------|------------|--------|--------|--------|
| PoI    | 0/0/0| 5/3/2     | 10/0/0    | 6/3/1      | 8/1/1       | 5/4/1      | 4/5/1  | 5/4/1  |
| q-PoI_all | 2/3/5 | 0/0/0     | 8/2/0     | 3/4/3      | 7/1/2       | 3/4/3      | 2/6/2  | 2/3/5  | 1/5/4  |
| q-PoI_one | 0/0/10| 0/2/8     | 0/0/0     | 0/4/6      | 1/6/3       | 0/3/7      | 0/1/9  | 0/1/9  | 0/3/7  |
| q-PoI_best | 1/3/6 | 3/4/3     | 6/4/0     | 0/0/0      | 6/3/1       | 2/5/3      | 1/3/6  | 2/4/4  | 1/5/4  |
| q-PoI_worse | 1/1/8 | 3/0/7     | 3/6/1     | 1/3/6      | 0/0/0       | 1/3/6      | 1/2/7  | 1/1/8  | 0/3/7  |
| q-PoI_mean | 2/3/5 | 3/4/3     | 7/3/0     | 3/5/2      | 6/3/1       | 0/0/0      | 1/6/3  | 2/4/4  | 1/7/2  |
| KB-PoI | 1/5/4 | 7/1/2     | 9/1/0     | 6/3/1      | 7/2/1       | 3/6/1      | 0/0/0  | 1/8/1  | 3/6/1  |
| CL-PoI | 1/4/5 | 5/3/2     | 10/0/0    | 4/4/2      | 8/1/1       | 4/4/2      | 1/8/1  | 0/0/0  | 3/5/2  |
| q-EHVI | 1/4/5 | 4/5/1     | 7/3/0     | 4/5/1      | 7/3/0       | 2/7/1      | 1/6/3  | 2/6/2  | 0/0/0  |

Sum of +/ ≈ /−: 9/23/48 30/22/28 60/19/1 27/31/22 50/20/10 20/36/24 11/33/36 15/31/34 14/38/28

[2] N. Li, L. Zhao, C. Bao, G. Gong, X. Song, C. Tian, A real-time information integration framework for multidisciplinary coupling of complex aircrafts: an application of iiie, Journal of Industrial Information Integration 22 (2021) 100203. doi:https://doi.org/10.1016/j.jii.2021.100203.
URL https://www.sciencedirect.com/science/article/pii/S2452414X21000042

[3] D. Han, W. Du, X. Wang, W. Du, A surrogate-assisted evolutionary algorithm for expensive many-objective optimization in the refining process, Swarm and Evolutionary Computation 69 (2022) 100988. doi:https://doi.org/10.1016/j.swevo.2021.100988.
URL https://www.sciencedirect.com/science/article/pii/S2210650221001504

[4] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter, Auto-sklearn: Efficient and robust automated machine learning, in: F. Hutter, L. Kotthoff, J. Vanschoren (Eds.), Automated Machine Learning - Methods, Systems, Challenges, The Springer Series on Challenges in Machine Learning, Springer, 2019, pp. 113–134. doi:10.1007/978-3-030-05318-5_6.
URL https://doi.org/10.1007/978-3-030-05318-5_6

[5] Y. Dai, P. Zhao, A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization, Applied Energy 279 (2020) 115332. doi:https://doi.org/10.1016/j.apenergy.2020.115332.
URL https://www.sciencedirect.com/science/article/pii/S0306261920308448

[6] A. Žilinskas, J. Mockus, On one Bayesian method of search of the minimum, Avtomatica i Vychislitel’naya Teknika 4 (1972) 42–44.

[7] J. Mockus, On bayesian methods for seeking the extremum, Springer Berlin Heidelberg, Berlin, Heidelberg, 1975, pp. 400–404. doi:10.1007/3-540-07165-2_55.
URL https://doi.org/10.1007/3-540-07165-2_55

[8] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (4) (1998) 455–492.

[9] M. Schonlau, Computer experiments and global optimization, Ph.D. thesis (1997).
[10] D. Ginsbourger, Métamodèles multiples pour l’approximation et l’optimisation de fonctions numériques multivari-ables, Ph.D. thesis (2009).

[11] D. Ginsbourger, R. Le Riche, L. Carraro, Kriging is well-suited to parallelize optimization, in: Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Expensive Optimization Problems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 131–162. doi:10.1007/978-3-642-10701-6_6. URL https://doi.org/10.1007/978-3-642-10701-6_6

[12] C. Chevalier, D. Ginsbourger, Fast computation of the multi-points expected improvement with applications in batch selection, in: G. Nicosia, P. Pardalos (Eds.), Learning and Intelligent Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 59–69.

[13] Q. Zhang, W. Liu, E. Tsang, B. Virginais, Expensive multiobjective optimization by moea/d with gaussian process model, IEEE Transactions on Evolutionary Computation 14 (3) (2009) 456–474.

[14] K. Yang, P. S. Palar, M. Emmerich, K. Shimoyama, T. Bäck, A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’19, ACM, New York, NY, USA, 2019, pp. 656–663. doi:10.1145/3321707.3321784. URL http://doi.acm.org/10.1145/3321707.3321784

[15] D. Gaudrie, R. Le Riche, V. Picheny, B. Enaux, V. Herbert, Targeting solutions in bayesian multi-objective optimization: sequential and batch versions, Annals of Mathematics and Artificial Intelligence 88 (1) (2020) 187–212. doi:10.1007/s10472-019-09644-8. URL https://doi.org/10.1007/s10472-019-09644-8

[16] M. Konakovic Lukovic, Y. Tian, W. Matusik, Diversity-guided multi-objective bayesian optimization with batch evaluations, Advances in Neural Information Processing Systems 33 (2020) 17708–17720.

[17] A. Schulz, H. Wang, E. Grinspun, J. Solomon, W. Matusik, Interactive exploration of design trade-offs, ACM Transactions on Graphics (TOG) 37 (4) (2018) 1–14.

[18] Z. Wang, Q. Zhang, Y.-S. Ong, S. Yao, H. Liu, J. Luo, Choose appropriate subproblems for collaborative modeling in expensive multiobjective optimization, IEEE Transactions on Cybernetics (2021) 1–14. doi:10.1109/TCYB.2021.3126341.

[19] G. De Ath, R. M. Everson, J. E. Fieldsend, Asynchronous ε-greedy bayesian optimisation, in: Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 578–588.

[20] F. J. Gibson, R. M. Everson, J. E. Fieldsend, Multi-objective bayesian optimisation using an exploitative attainment front acquisition function, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 1503–1510.

[21] G. De Ath, R. M. Everson, A. A. Rahat, J. E. Fieldsend, Greed is good: Exploration and exploitation trade-offs in bayesian optimisation, ACM Transactions on Evolutionary Learning and Optimization 1 (1) (2021) 1–22.

[22] S. Daulton, M. Balandat, E. Bakshy, Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[23] C. A. C. Coello, S. G. Brambila, J. F. Gamboa, M. G. C. Tapia, R. H. Gómez, Evolutionary multiobjective optimization: open research areas and some challenges lying ahead, Complex & Intelligent Systems 6 (2) (2020) 221–236.

[24] A. Zilinskas, A review of statistical models for global optimization, J. Glob. Optim. 2 (2) (1992) 145–153. doi:10.1007/BF00122051. URL https://doi.org/10.1007/BF00122051

[25] D. R. Jones, A Taxonomy of Global Optimization Methods Based on Response Surfaces, J. Glob. Optim. 21 (4) (2001) 345–383. doi:10.1023/A:1012771025575. URL https://doi.org/10.1023/A:1012771025575

[26] M. Emmerich, K. Yang, A. Deutz, Infill Criteria for Multiobjective Bayesian Optimization. Springer International Publishing, Cham, 2020, pp. 3–16. doi:10.1007/978-3-030-18764-4_1. URL https://doi.org/10.1007/978-3-030-18764-4_1

[27] K. Yang, K. van der Blom, T. Bäck, M. Emmerich, Towards single- and multiobjective bayesian global optimization for mixed integer problems, AIP Conference Proceedings 2070 (1) (2019) 020044. arXiv:https://arxiv.org/pdf/10.1063/1.5090011. doi:10.1063/1.5090011. URL https://aip.scitation.org/doi/abs/10.1063/1.5090011
[28] E. C. Garrido-Merchán, D. Hernández-Lobato, Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes, Neurocomputing 380 (2020) 20 – 35. doi:https://doi.org/10.1016/j.neucom.2019.11.004. URL http://www.sciencedirect.com/science/article/pii/S0925231219315619

[29] M. McKaya, R. Beckmana, W. Conoverb, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 239–245.

[30] M. D. Buhmann, Radial Basis Functions - Theory and Implementations, Vol. 12 of Cambridge monographs on applied and computational mathematics, Cambridge University Press, 2009. URL http://www.cambridge.org/de/academic/subjects/mathematics/numerical-analysis/radial-basis-functions-theory-and-implementations

[31] C. Chevalier, D. Ginsbourger, Fast computation of the multi-points expected improvement with applications in batch selection, in: G. Nicosia, P. M. Pardalos (Eds.), Learning and Intelligent Optimization - 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers, Vol. 7997 of Lecture Notes in Computer Science, Springer, 2013, pp. 59–69. doi:10.1007/978-3-642-44973-4\_7

URL https://doi.org/10.1007/978-3-642-44973-4\_7

[32] T. J. Santner, B. J. Williams, W. I. Notz, The Design and Analysis of Computer Experiments, Springer series in statistics, Springer, 2003. doi:10.1007/978-1-4757-3799-8.

[33] P. Boyle, M. R. Frean, Dependent Gaussian Processes, in: Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada], 2004, pp. 217–224. URL http://papers.nips.cc/paper/2561-dependent-gaussian-processes

[34] K. Yang, P. S. Palar, M. Emmerich, K. Shimoyama, T. Bäck, A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’19, Association for Computing Machinery, New York, NY, USA, 2019, p. 656–663. doi:10.1145/3321707.3321784.

URL https://doi.org/10.1145/3321707.3321784

[35] K. Yang, M. Emmerich, A. Deutz, T. Bäck, Multi-objective bayesian global optimization using expected hypervolume improvement gradient, Swarm and Evolutionary Computation 44 (2019) 945 – 956. doi:https://doi.org/10.1016/j.swevo.2018.10.007.

URL http://www.sciencedirect.com/science/article/pii/S2210650217307861

[36] C. A. Coello Coello, Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition, in: J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, C. Ben-Youssef Brants, E. R. Hancock (Eds.), Proceedings of the Third Mexican conference on Pattern recognition, Springer, Berlin, Heidelberg, 2011, pp. 22–33. doi:10.1007/978-3-642-21587-2\_3.

URL http://dx.doi.org/10.1007/978-3-642-21587-2\_3

[37] K. Yang, M. Emmerich, A. Deutz, T. Bäck, Efficient computation of expected hypervolume improvement using box decomposition algorithms, Journal of Global Optimization 75 (1) (2019) 3–34. doi:10.1007/s10898-019-00798-7.

URL https://doi.org/10.1007/s10898-019-00798-7

[38] W. K. Hastings, Monte Carlo sampling methods using markov chains and their applications, Biometrika 57 (1) (1970) 97–109.

[39] M. Emmerich, K. Yang, A. Deutz, H. Wang, M. C. Fonseca, A multicriteria generalization of bayesian global optimization, in: P. M. Pardalos, A. Zhigljavsky, J. Žilinskas (Eds.), Advances in Stochastic and Deterministic Global Optimization, Springer, Berlin, Heidelberg, 2016, pp. 229–243.

[40] Z. Drezner, G. O. Wesolowsky, On the computation of the bivariate normal integral, Journal of Statistical Computation and Simulation 35 (1-2) (1990) 101–107. arXiv:https://doi.org/10.1080/00949659008811236, doi:10.1080/00949659008811236.

URL https://doi.org/10.1080/00949659008811236

[41] Z. Drezner, Computation of the trivariate normal integral, Mathematics of Computation 62 (205) (1994) 289–294.

[42] A. Genz, Numerical computation of rectangular bivariate and trivariate normal and t probabilities, Statistics and Computing 14 (3) (2004) 251–260.

[43] A. Genz, F. Bretz, Numerical computation of multivariate t-probabilities with application to power calculation of multiple contrasts, Journal of Statistical Computation and Simulation 63 (4) (1999) 103–117. arXiv:https://doi.org/10.1080/00949659908811962, doi:10.1080/00949659908811962.

URL https://doi.org/10.1080/00949659908811962
[44] A. Genz, F. Bretz, Comparison of methods for the computation of multivariate t probabilities, Journal of Computational and Graphical Statistics 11 (4) (2002) 950–971. arXiv:https://doi.org/10.1198/106186002394, doi:10.1198/106186002394. URL https://doi.org/10.1198/106186002394

[45] A. Genz, F. Bretz, Computation of multivariate normal and t probabilities, Vol. 195, Springer Science & Business Media, 2009.

[46] Z. I. Botev, The normal law under linear restrictions: simulation and estimation via minimax tilting, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79 (1) (2017) 125–148.

[47] M. Emmerich, C. M. Fonseca, Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results, in: Evolutionary Multi-Criterion Optimization, Springer, 2011, pp. 121–135.

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

[49] M. Emmerich, A. H. Deutz, Test problems based on Lamé superspheres, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2007, pp. 922–936.

[50] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, A benchmark test suite for evolutionary many-objective optimization, Complex & Intelligent Systems 3 (1) (2017) 67–81. doi:10.1007/s40747-017-0039-7.

[51] Z. Wang, Y.-S. Ong, H. Ishibuchi, On scalable multiobjective test problems with hardly dominated boundaries, IEEE Transactions on Evolutionary Computation 23 (2) (2019) 217–231. doi:10.1109/TEVC.2018.2844286.

[52] Z. Wang, Y.-S. Ong, J. Sun, A. Gupta, Q. Zhang, A generator for multiobjective test problems with difficult-to-approximate pareto front boundaries, IEEE Transactions on Evolutionary Computation 23 (4) (2019) 556–571. doi:10.1109/TEVC.2018.2872453.

[53] J. Knowles, ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 10 (1) (2006) 50–66.

[54] E. Bradford, A. M. Schweidtmann, A. Lapkin, Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm, Journal of global optimization 71 (2) (2018) 407–438.

[55] N. Hansen, Benchmarking a bi-population cma-es on the bbob-2009 function testbed, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO ’09, Association for Computing Machinery, New York, NY, USA, 2009, p. 2389–2396. doi:10.1145/1570256.1570333. URL https://doi.org/10.1145/1570256.1570333

[56] E. Bradford, A. M. Schweidtmann, A. Lapkin, Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm, Journal of Global Optimization 71 (2) (2018) 407–438. doi:10.1007/s10898-018-0609-2.

[57] C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, L. Paquete, On the computation of the empirical attainment function, in: R. H. C. Takahashi, K. Deb, E. F. Wanner, S. Greco (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 106–120.

[58] J. Karder, A. Beham, B. Werth, S. Wagner, M. Affenzeller, Asynchronous surrogate-assisted optimization networks, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1266–1267. doi:10.1145/3205651.3208246. URL https://doi.org/10.1145/3205651.3208246

[59] M. Affenzeller, A. Beham, S. Vonolken, E. Pitzer, S. M. Winkler, S. Hutterer, M. Kommenda, M. Kofler, G. Kronberger, S. Wagner, Simulation-Based Optimization with HeuristicLab: Practical Guidelines and Real-World Applications, Springer International Publishing, Cham, 2015, pp. 3–38. doi:10.1007/978-3-319-15033-8_1. URL https://doi.org/10.1007/978-3-319-15033-8_1

[60] B. Werth, E. Pitzer, M. Affenzeller, Surrogate-assisted fitness landscape analysis for computationally expensive optimization, in: R. Moreno-Díaz, F. Pichler, A. Quesada-Arencibia (Eds.), Computer Aided Systems Theory – EUROCAST 2019, Springer International Publishing, Cham, 2020, pp. 247–254.