Swarm and Evolutionary Computation 44 (2019) 945–956

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient

Kaifeng Yang *, Michael Emmerich, André Deutz, Thomas Bäck

LIACS, Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, the Netherlands

ARTICLE INFO

Keywords:
Bayesian global optimization
Expected hypervolume improvement
Expected hypervolume improvement gradient
Kriging stopping criterion

ABSTRACT

The Expected Hypervolume Improvement (EHVI) is a frequently used infill criterion in Multi-Objective Bayesian Global Optimization (MOBGO), due to its good ability to lead the exploration. Recently, the computational complexity of EHVI calculation is reduced to $O(n \log n)$ for both 2-D and 3-D cases. However, the optimizer in MOBGO still requires a significant amount of time, because the calculation of EHVI is carried out in each iteration and usually tens of thousands of the EHVI calculations are required. This paper derives a formula for the Expected Hypervolume Improvement Gradient (EHVIG) and proposes an efficient algorithm to calculate EHVIG. The new criterion (EHVIG) is utilized by two different strategies to improve the efficiency of the optimizer discussed in this paper. Firstly, it enables gradient ascent methods to be used in MOBGO. Moreover, since the EHVIG of an optimal solution should be a zero vector, it can be regarded as a stopping criterion in global optimization, e.g., in Evolution Strategies. Empirical experiments are performed on seven benchmark problems. The experimental results show that the second proposed strategy, using EHVIG as a stopping criterion for local search, can outperform the normal MOBGO on problems where the optimal solutions are located in the interior of the search space. For the ZDT series test problems, EHVIG still can perform better when gradient projection is applied.

1. Introduction

Evolutionary Algorithms (EAs) and Bayesian Global Optimization (BGO) are two main branches in the field of optimization. Both of them share a similar structure: initialization, evaluation of a black box function at a given search point, an update of the current search point for seeking an improvement in the next loop and repetition of the evaluation and adjustment loop. The difference lies in the update mechanism. For EAs, this is accomplished by evolutionary operators, such as recombination and mutation. For the Bayesian global optimization, this is based on learning from the past evaluations and determining the next search point by optimization of an infill criterion formulated on that method. Compared to EAs, BGO requires only a small budget of function evaluations. Therefore, it can be applied to real-world optimization problems with expensive evaluations [1], e.g., evaluations occurring in computational fluid dynamics simulations or process control simulation.

In the context of Bayesian Global Optimization, a pre-selection or infill criterion is utilized to estimate the performance of the improvement for a new point. For single objective optimization, *Expected Improvement* (EI) and *Probability of Improvement* (PoI) are usually applied in BGO. The EI was introduced by Mockus et al. [2] in 1978, and it exploits both the Kriging prediction and the variance to give a quantitative measure of the improvement for the points in the search space. Later, EI became more popular due to the work of Jones et al. [3]. Currently, EI is widely used in Bayesian Global Optimization and machine learning. In 2005, Emmerich generalized EI into EHVI based on the hypervolume indicator [4]. Similar to EI, the EHVI is the expected increment of the hypervolume indicator, considering a Pareto-front approximation set and a predictive multivariate Gaussian distribution at a new point.

EHVI has been in existence for more than a decade, and it has the property to achieve a good convergence and diversity to a true Pareto front [5–8]. It also yields good results when applied as an infill criterion in BGO and pre-selection criterion in Evolution Strategies in optimization studies. However, it was frequently criticized for the high computational effort that seemed to be required when computing the underlying

* Corresponding author.
E-mail addresses: k.yang@liacs.leidenuniv.nl (K. Yang), m.t.m.emmerich@liacs.leidenuniv.nl (M. Emmerich), a.h.deutz@liacs.leidenuniv.nl (A. Deutz), t.h.w.baeck@liacs.leidenuniv.nl (T. Bäck).

https://doi.org/10.1016/j.swevo.2018.10.007
Received 27 September 2017; Received in revised form 24 September 2018; Accepted 15 October 2018
Available online 28 October 2018

multi-variable integrals. The first method suggested for EHVI calculation was Monte Carlo integration and it was first proposed by Emmerich in Refs. [4] and [9]. This method is simple and straightforward. However, the accuracy of EHVI highly depends on the number of the iterations. The first exact EHVI calculation algorithm for a 2-D case was derived in Ref. [10], with the computational complexity $O(n^3 \log n)$. Couckuyt et al. introduced exact EHVI calculation for $d > 2$ in Ref. [5]. This method was also practically much faster than those discussed in Ref. [10], although a detailed complexity analysis was missing. In 2015, Hupkens et al. reduced the time complexity to $O(n^2)$ and $O(n^3)$ [11] for the two and the three-dimensional case, respectively. These algorithms also further improved the practical efficiency of EHVI on test data. However, there still exists a large gap to use EHVI in applications. Considering the expensive computation of EHVI and inspired by the EHVI, Couckuyt et al. proposed the *Hypervolume based Probability of Improvement* in Ref. [5]. Luo et al. used an approximate algorithm to calculate EHVI based on Monte Carlo sampling for high dimensional cases ($d > 6$) in Ref. [7], where d stands for the dimension in objective space. Finally, Emmerich et al. proposed an asymptotically optimal algorithm for the bi-objective case with time complexity $\Theta(n \log n)$ in Ref. [12], where n is the number of non-dominated points in the archive. More recently, Yang et al. [13] proposed an algorithm to calculate 3-D EHVI with computational complexity $\Theta(n \log n)$.

However, compared to EAs, Multi-Objective Bayesian Global Optimization still performs much slower with the infill criterion EHVI, because EHVI needs to be called many times in the process of searching for the optimal point based on the Kriging models. Since the calculation of the EHVI can be formulated in closed form, it is possible to analyze its differentiability. It is easy to see, that all components of the EHVI expression are differentiable. However, a precise formula of the EHVIG has not been derived until now. By using the formula for the EHVIG, it could speed up the MOBGO in the process of searching for the optimal point by using the gradient ascent algorithm or using it as a stopping criterion in EAs. This is the motivation of the research in this paper.

This paper mainly discusses the computation of the 2-D EHVIG and how to apply EHVIG in MOBGO, both for local search (gradient ascent) and as a stopping criterion. The paper is structured as follows: Section 2 briefly describes *Bayesian Global Optimization*, some basic infill criteria, and how to compute 2-D EHVI efficiently; Section 3 introduces the definition of the EHVIG, and proposes an efficient algorithm to calculate 2-D EHVIG, including a computational performance assessment of the proposed efficient exact calculation method and numerical calculation method in 2-D EHVIG case; Section 4 introduces the techniques on how to apply EHVIG in MOBGO; Section 5 shows some empirical, experimental results; Section 6 draws the main conclusions of this paper and discusses some potential topics for future research.

2. State of the art[1]

2.1. Bayesian Global Optimization

Bayesian Global Optimization (BGO), also known as *Efficient Global Optimization* [3] or *Expected Improvement Algorithm* [14], was proposed by the Lithuanian research group of Jonas Mockus and Antanas Žilinskas ([2,15–17]) in the 1970s. In BGO, it is assumed that the objective function is the realization of a Gaussian random field, which is also called Gaussian process (GP) or Kriging, in particular in 1-D.

Kriging is a statistical interpolation method. Being a Gaussian process based modelling method, it is cheap to evaluate [18]. Kriging has been proven to be a popular surrogate model to approximate noise-free data in computer experiments, where Kriging models are fitted on previously evaluated points and then replace the real time-consuming simulation model [19]. Given a set of n decision vectors $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n)})^\top$ in m dimensional search space, and associated function values $y(\mathbf{X}) = (y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(2)}), \ldots, y(\mathbf{x}^{(n)}))^\top$, Kriging assumes y to be a realization of a random process Y and it is of the form [3,20]:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x}) \tag{2-1}$$

where $\mu(\mathbf{x})$ is estimated mean value over all given sampled points, and $\epsilon(\mathbf{x})$ is a realization of a normally distributed Gaussian random process with zero mean and variance σ^2. The regression part $\mu(\mathbf{x})$ approximates globally the function Y and Kriging/Gaussian process $\epsilon(\mathbf{x})$ takes local variations into account. Moreover, as opposed to other regression methods, such as supported vector machine (SVM), Kriging/GP also provides an uncertainty qualification of a prediction. The correlation between the deviations at two points ($\mathbf{x}$ and $\mathbf{x}'$) is defined as:

$$Corr[\epsilon(\mathbf{x}), \epsilon(\mathbf{x}')] = R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{m} R_i(x_i, x_i') \tag{2-2}$$

Here $R(.,.)$ is the correlation function, which can be a cubic or a spline function. Commonly, a Gaussian function (also known as squared exponential) is chosen:

$$R(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{m} \exp(-\theta_i (x_i - x_i')^2) \qquad (\theta_i >= 0) \tag{2-3}$$

where θ are parameters of correlation model and they can be interpreted as measuring the importance of the variable. Then the covariance matrix can be expressed by the correlation function:

$$Cov(\epsilon) = \sigma^2 \mathbf{\Sigma}, \qquad \text{where} \qquad \mathbf{\Sigma}_{ij} = R(\mathbf{x_i}, \mathbf{x_j}) \tag{2-4}$$

When $\mu(\mathbf{x})$ is assumed to be an unknown constant, this unbiased prediction is called ordinary Kriging (OK). In OK, the Kriging model determines the hyperparameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_n]$ by maximizing the likelihood of the observed dataset. The expression of the likelihood function is:

$$L = -\frac{n}{2} \ln(\sigma^2) - \frac{1}{2} \ln(|\mathbf{\Sigma}|) \tag{2-5}$$

The maximum likelihood estimates of the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ are given by Ref. [3]:

$$\hat{\mu} = \frac{\mathbf{1}_n^\top \mathbf{\Sigma}^{-1} \mathbf{y}}{\mathbf{1}_n^\top \mathbf{\Sigma}^{-1} \mathbf{1}_n} \tag{2-6}$$

$$\hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{1}_n \hat{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{1}_n \hat{\mu}) \tag{2-7}$$

Then the predictor of the mean and the variance at point x^t can be derived and they are shown as follows [3]:

$$\mu(\mathbf{x}^t) = \hat{\mu} + \mathbf{c}^\top \mathbf{\Sigma}^{-1} (\mathbf{y} - \hat{\mu} \mathbf{1}_n) \tag{2-8}$$

$$\sigma^2(\mathbf{x}^t) = \hat{\sigma}^2 \left(1 - \mathbf{c}^\top \mathbf{\Sigma}^{-1} \mathbf{c} + \frac{1 - \mathbf{c}^\top \Sigma^\top \mathbf{c}}{\mathbf{1}_n^\top \Sigma^{-1} \mathbf{1}_n} \right) \tag{2-9}$$

where $\mathbf{c} = (Corr[Y(\mathbf{x}^t), Y(\mathbf{x}_1)], \ldots, Corr[Y(\mathbf{x}^t), Y(\mathbf{x}_n)])^\top$. The time complexity of computing these values at an input vector $\mathbf{x}$, once the hyperparameters are fixed, is $(O(mn))$ per computation of $\hat{\mu}(.)$, and $O(n^2 + mn)$ per computation of $\hat{\sigma}(.)$.

The basic idea of BGO is to use a surrogate model based on Kriging or a Gaussian process. A surrogate model reflects the relationship between decision vectors and their corresponding objective values. This surrogate model is learned from previous evaluations. For multi-objective problems, the family of these algorithms is called *Multi-Objective Bayesian Global Optimization* (MOBGO). The scheme of a MOBGO algorithm is to sequentially update a surrogate model, by the optimal point searched by an optimizer and the corresponding objective function values. An optimizer in MOBGO is utilized to search for a promising point $\mathbf{x}^*$ by maximizing/minimizing an infill criterion with respect to surrogate models, instead of using 'true' objective functions.

[1] For the convenience of the visualization, this paper only considers minimization problems.

2.2. Infill criteria

In *Multi-Objective Bayesian Global Optimization*, some common *infill criteria* are: *Hypervolume Indicator* (HV) [21], *Hypervolume Improvement* (HVI) [22],[2] *Hypervolume Contribution* (HVC) [23], *Lower Confidence Bound* (LCB) [9,24,25], EHVI [4,26], *Probability of Improvement* (PoI) [27] and *Truncated Expected Hypervolume Improvement* (TEHVI) [28,29]. Many of these are based on the hypervolume indicator.

The HV was proposed by Zitzler and Thiele [30], and it measures the size of the dominated subspace bounded from above by a reference point $\mathbf{r}$. This reference point should be chosen by a user, and it should satisfy the condition that it is dominated by all the elements of the Pareto-front approximation sets which might occur during the optimization process, if possible. The hypervolume can indicate the performance of a Pareto-front approximation set $\mathcal{P} \subset \mathbb{R}^d$, and the maximization of HV can lead to a Pareto-front approximation set that is close to the true Pareto front. In 2-D and 3-D cases, the hypervolume indicator can be computed in time $\Theta(n \log n)$ [31]. In more than 3 dimensions, the algorithm proposed by Chan [32] achieves $O\left(n^{\frac{d}{3}}\text{polylog}\, n\right)$ time complexity. The hypervolume indicator is defined as:

Definition 2.1. (Hypervolume Indicator) *Given a finite approximation to a Pareto front, say $\mathcal{P} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\} \subset \mathbb{R}^d$, the Hypervolume Indicator (HV) of $\mathcal{P}$ is defined as the d -dimensional Lebesgue measure of the subspace dominated by $\mathcal{P}$ and bounded from above by a reference point $\mathbf{r}$:*

$$HV(\mathcal{P}) = \lambda_d(\cup_{\mathbf{y}\in\mathcal{P}}[\mathbf{y}, \mathbf{r}]) \tag{2-10}$$

with λ_d being the Lebesgue measure on $\mathbb{R}^d$.

Two straightforward derived criteria are the HVI and the HVC. Emmerich et al. proposed an asymptotically optimal algorithm to calculate HVC with time complexity $\Theta(n \log n)$ for $d = 2, 3$ in Ref. [33]. The basic idea behind these two criteria is the same, that is to calculate the difference of the hypervolume between two Pareto-front approximation sets. The definition of HVC of a point $\mathbf{y} \in \mathcal{P}$ is the difference between the hypervolume of $\mathcal{P}$ and the hypervolume of $\mathcal{P} \setminus \{\mathbf{y}\}$. *Hypervolume Improvement* is defined as:

Definition 2.2. (Hypervolume Improvement) *Given a finite collection of vectors $\mathcal{P} \subset \mathbb{R}^d$, the Hypervolume Improvement of a vector $\mathbf{y} \in \mathbb{R}^d$ is defined as:*

$$HVI(\mathcal{P}, \mathbf{y}) = HV(\mathcal{P} \cup \{\mathbf{y}\}) - HV(\mathcal{P}) \tag{2-11}$$

In case we want to emphasize the reference point $\mathbf{r}$, the notation $\text{HVI}(\mathcal{P}, \mathbf{y}, \mathbf{r})$ will be used to denote the Hypervolume Improvement. Note that $\text{HVI}(\mathcal{P}, \mathbf{y}) = 0$, in case $\mathbf{y} \in \mathcal{P}$.

EHVI is a generalization of EI to the multi-objective case, based on the theory of the HV. Similar to EI, the definition of EHVI is with respect to the predictions in the Gaussian random field and it measures how much hypervolume improvement could be achieved by evaluating the new point, considering the uncertainty of the prediction. It is defined as:

Definition 2.3. (Expected Hypervolume Improvement)[3] *Given parameters of the multivariate predictive distribution $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and the Pareto-front approximation $\mathcal{P}$ the expected hypervolume improvement (EHVI) is defined as:*

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) \coloneqq \int_{\mathbb{R}^d} HVI(\mathcal{P}, \mathbf{y}) \cdot PDF_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{2.12}$$

where $PDF_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is the multivariate independent normal distribution for mean values $\boldsymbol{\mu} \in \mathbb{R}^d$, and standard deviations $\boldsymbol{\sigma} \in \mathbb{R}^d_+$.

[2] The HVI was called the most likely improvement (MLI) in Ref. [22].

[3] The prediction of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ depends on a Kriging model and a target point $\mathbf{x}$ in the search space. Explicitly, EHVI is dependent on the target point $\mathbf{x}$.

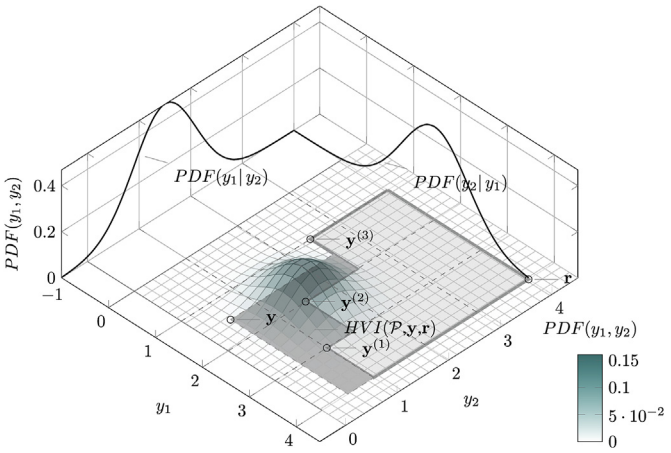

Fig. 1. EHVI in 2-D (cf. Example 2.1).

Example 2.1. An illustration of the EHVI is shown in Fig. 1. The light gray area is the dominated subspace of $\mathcal{P} = \{\mathbf{y}^{(1)} = (3,1)^\top, \mathbf{y}^{(2)} = (2,1.5)^\top, \mathbf{y}^{(3)} = (1,2.5)^\top\}$ *cut by the reference point* $\mathbf{r} = (4,4)^\top$. *The bivariate Gaussian distribution has the parameters* $\mu_1 = 2$, $\mu_2 = 1.5$, $\sigma_1 = 0.7$, $\sigma_2 = 0.6$. *The probability density function (PDF) of the bivariate Gaussian distribution is indicated as a 3-D plot. Here* $\mathbf{y}$ *is a sample from this distribution and the area of improvement relative to* $\mathcal{P}$ *is indicated by the dark shaded area. The variable* y_1 *stands for the* f_1 *value and* y_2 *for the* f_2 *value.*

For the convenience of expressing the formula of EHVI and EHVIG in later sections, it is useful to define a function we call Ψ.

Definition 2.4. (Ψ **function (see also** [11])) *Let* $\phi(s) = 1/\sqrt{2\pi}e^{-\frac{1}{2}s^2}, s \in \mathbb{R}$ *denote the PDF of the standard normal distribution and* $\Phi(s) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{s}{\sqrt{2}}\right)\right)$ *denote its cumulative probability distribution function (CDF). The general normal distribution with mean* μ *and variance* σ *has the PDF* $\phi_{\mu,\sigma}(s) = \frac{1}{\sigma}\phi(\frac{s-\mu}{\sigma})$ *and the CDF* $\boldsymbol{\Phi}_{\mu,\sigma}(s) = \Phi(\frac{s-\mu}{\sigma})$. *Moreover, a useful identity which we will frequently use is:*

$$\int_{-\infty}^{b} (a - z)\frac{1}{\sigma}\phi(\frac{z-\mu}{\sigma})dz = \sigma\phi(\frac{b-\mu}{\sigma}) + (a-\mu)\Phi\left(\frac{b-\mu}{\sigma}\right) \tag{2-13}$$

. We define the function Ψ *as follows:*

$$\Psi(a, b, \mu, \sigma) \coloneqq \int_{-\infty}^{b} (a - z)\frac{1}{\sigma}\phi(\frac{z-\mu}{\sigma})dz \tag{2-14}$$

2.3. Efficient algorithm for 2-D EHVI calculation

This paper focuses on bi-objective problems. The calculation of EHVIG in Section 3 shares the same partitioning method with 2-D EHVI calculation and EHVIG is derived from EHVI. Therefore, it is necessary to introduce an efficient algorithm for 2-D EHVI calculation, as described in the recent work by Emmerich et al. [12].

The partitioning of the integration domain is done by the following steps: augment the current Pareto-front approximation $\mathcal{P} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\}$ by two more points $\mathbf{y}^{(0)} = (r_1, -\infty)^\top$ and $\mathbf{y}^{(n+1)} = (-\infty, r_2)^\top$; sort the points in $\mathcal{P}$ in ascending order by the second coordinate of the points. Then, the dominated space will be divided into $n+1$ disjoint rectangular stripes $S_1, \ldots, S_{n+1}$, and these stripes are defined by:

$$S_i = \left((y_1^{(i)}, -\infty)^\top, (y_1^{(i-1)}, y_2^{(i)})^\top\right) i = 1, \ldots, n+1 \tag{2-15}$$

See Fig. 2 for an illustration.

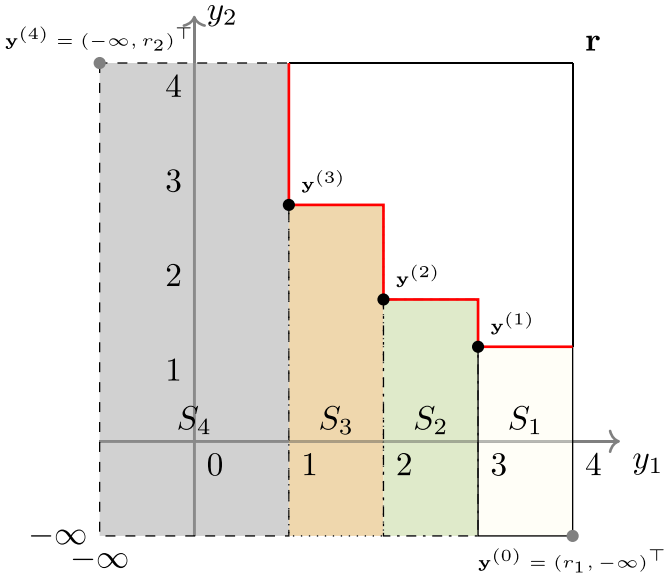

Fig. 2. Partitioning of the integration region into stripes.

For the convenience of computing EHVI, it is useful to define the function Δ, which is defined as:

Definition 2.5. (**Δ function (see also [12])**) *For a given vector of objective function values, $\mathbf{y} \in \mathbb{R}^d$, $\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r})$ is the subset of the vectors in $\mathbb{R}^d$ which are exclusively dominated by a vector $\mathbf{y}$ and not by elements in $\mathcal{P}$ and that dominate the reference point, in symbols*

$$\Delta(\mathcal{P}, \mathbf{y}, \mathbf{r}) = \lambda_d\{\mathbf{z} \in \mathbb{R} \mid \mathbf{y} \prec \mathbf{z} \text{ and } \mathbf{z} \prec \mathbf{r} \text{ and } \nexists \mathbf{q} \in \mathcal{P} : \mathbf{q} \prec \mathbf{z}\} \tag{2-16}$$

For the simplicity, the notation $\Delta(\mathbf{y})$ will be used to express $\Delta(\mathcal{P}, \mathbf{y}, \mathbf{r})$ in this paper.

Then, the hypervolume improvement of a point $\mathbf{y} \in \mathbb{R}^2$ can be expressed by:

$$\mathrm{HVI}(\mathcal{P}, \mathbf{y}, \mathbf{r}) = \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta(y_1, y_2)] \tag{2-17}$$

Here, $\Delta(y_1, y_2)$ is the part of the objective space that is dominated by $\mathbf{y} = (y_1, y_2)$. Recall the definition of EHVI, then the EHVI formula can be derived that consists of $n+1$ integrals:

$$\mathrm{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \int_{y_1=-\infty}^{\infty} \int_{y_2=-\infty}^{\infty} \sum_{i=1}^{n+1} \lambda_2[S_i \cap \Delta((y_1, y_2))] \cdot PDF_{\boldsymbol{\mu},\boldsymbol{\sigma}}(y_1, y_2) dy_1 dy_2 \tag{2-18}$$

It is observed that the intersection of S_i with $\Delta(y_1, y_2)$ is non-empty if and only if $\mathbf{y} = (y_1, y_2)$ dominates the upper right corner of S_i, and it is allowed to do the summation after integration since integration is a linear mapping, therefore:

$$\mathrm{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \sum_{i=1}^{n+1} \int_{y_1=-\infty}^{y_1^{(i-1)}} \int_{y_2=-\infty}^{y_2^{(i)}} \lambda_2[S_i \cap \Delta(y_1, y_2)] \cdot PDF_{\boldsymbol{\mu},\boldsymbol{\sigma}}(y_1, y_2) dy_1 dy_2 \tag{2-19}$$

After some basic derivations, the final expression for the 2-D EHVI is [12]

$$\begin{aligned}\mathrm{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = &\sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi\left(\frac{y_1^{(i)} - \mu_1}{\sigma_1}\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \\ &+ \sum_{i=1}^{n+1} \left(\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1)\right) \\ &\cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)\end{aligned} \tag{2-20}$$

3. Expected hypervolume improvement gradient

This section will mainly introduce the definition of EHVIG, how to calculate EHVIG and show some performance assessment between the exact calculation method and the numerical calculation method.

3.1. Definition

Considering the definition of the EHVI in Equation (2.5) and the efficient algorithm to calculate 2-D EHVI (minimization case), the EHVI is differentiable with respect to the predictive mean and its corresponding standard deviation provided, which is greater than zero. These two parameters, predictive mean and standard deviation, are again differentiable with respect to the input vector (or target point) in the search space. The EHVIG is the first order derivative of the EHVI with respect to a target point $\mathbf{x}$ under consideration in the search space. It is defined as:

Definition 3.1. (**Expected Hypervolume Improvement Gradient**)[4] *Given parameters of the multivariate predictive distribution$\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ at a target point $\mathbf{x}$ in the search space, the Pareto-front approximation $\mathcal{P}$, and a reference point $\mathbf{r}$, the expected hypervolume improvement gradient (EHVIG) at $\mathbf{x}$ is defined as:*

$$\begin{aligned}EHVIG(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) &= \frac{\partial\,(EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}))}{\partial \mathbf{x}} \\ &= \frac{\partial\left(\int_{\mathbb{R}^d} HVI(\mathcal{P}, \mathbf{y}) \cdot PDF_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y}\right)}{\partial \mathbf{x}}\end{aligned} \tag{3-1}$$

3.2. Formula derivation

According to the definition of EHVIG in Equation (3-1) and the efficient algorithm to calculate EHVI in Equation (2-20), we can substitute the Equation (2-20) into Equation (3-1), say that the formula of EHVIG for 2-D case can be expressed as:

$$\begin{aligned}EHVIG(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) &= \frac{\partial(EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}))}{\partial \mathbf{x}} \\ &= \frac{\partial\left(\sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi(\frac{y_1^{(i)} - \mu_1}{\sigma_1}) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)\right)}{\partial \mathbf{x}}\end{aligned} \tag{3-2}$$

$$+ \frac{\partial\left(\sum_{i=1}^{n+1} \left(\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1)\right) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2)\right)}{\partial \mathbf{x}} \tag{3-3}$$

For the Terms (3-2) and (3-3), the prerequisite of calculating these two Terms is to calculate the gradient of the Ψ function and of the $\Phi(\frac{y-\mu}{\sigma})$ function. The final expressions for $\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial \mathbf{x}}$ and $\frac{\partial\Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}}$ are

[4] The prediction of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ depends on a Kriging model and a target point $\mathbf{x}$ in the search space. Explicitly, EHVIG is dependent on the target point $\mathbf{x}$.

shown in Equation (3-4) and Equation (3-5), respectively. For detailed proofs, please refer to the Appendix of this paper.

$$\frac{\partial \Psi(a,b,\mu,\sigma)}{\partial \mathbf{x}} = \left(\frac{b-a}{\sigma}\cdot\phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma})\right)\cdot\frac{\partial \mu}{\partial \mathbf{x}} + \phi(\frac{b-\mu}{\sigma})\cdot\left(1+\frac{(b-\mu)(b-a)}{\sigma^2}\right)\cdot\frac{\partial \sigma}{\partial \mathbf{x}} \tag{3-4}$$

$$\frac{\partial \Phi(\frac{y-\mu}{\sigma})}{\partial \mathbf{x}} = \phi(\frac{y-\mu}{\sigma})\cdot(\frac{\mu-y}{\sigma^2}\cdot\frac{\partial \sigma}{\partial \mathbf{x}} - \frac{1}{\sigma}\cdot\frac{\partial \mu}{\partial \mathbf{x}}) \tag{3-5}$$

By substituting Equations (3-4) and (3-5) into Term (3-2) and applying the chain rule, Term (3-2) can be expressed by:

$$\begin{aligned}
&\frac{\partial\left(\sum_{i=1}^{n+1}(y_1^{(i-1)}-y_1^{(i)})\cdot\Phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)\right)}{\partial \mathbf{x}}\\
&=\sum_{i=1}^{n+1}(y_1^{(i-1)}-y_1^{(i)})\cdot\frac{\partial\left(\Phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)\right)}{\partial \mathbf{x}}\\
&=\sum_{i=1}^{n+1}(y_1^{(i-1)}-y_1^{(i)})\cdot\left(\phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\cdot(\frac{\mu_1-y_1^{(i)}}{\sigma_1^2}\cdot\frac{\partial \sigma_1}{\partial \mathbf{x}}-\frac{1}{\sigma_1}\cdot\frac{\partial \mu_1}{\partial \mathbf{x}})\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)+\left(0-\Phi(\frac{(y_2^{(i)}-\mu_2)}{\sigma_2}\cdot\frac{\partial \mu_2}{\partial \mathbf{x}})+\phi(\frac{y_2^{(i)}-\mu_2}{\sigma_2})\cdot(1+0)\cdot\frac{\partial \sigma_2}{\partial \mathbf{x}}\right)\cdot\Phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\right)\\
&=\sum_{i=1}^{n+1}(y_1^{(i-1)}-y_1^{(i)})\cdot\left(\phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\cdot(\frac{\mu_1-y_1^{(i)}}{\sigma_1^2}\cdot\frac{\partial \sigma_1}{\partial \mathbf{x}}-\frac{1}{\sigma_1}\cdot\frac{\partial \mu_1}{\partial \mathbf{x}})\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)+\left(\phi(\frac{y_2^{(i)}-\mu_2}{\sigma_2})\cdot\frac{\partial \sigma_2}{\partial \mathbf{x}}-\Phi(\frac{y_2^{(i)}-\mu_2}{\sigma_2})\cdot\frac{\partial \mu_2}{\partial \mathbf{x}}\right)\cdot\Phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\right)
\end{aligned} \tag{3-6}$$

Similar to the derivation of Term (3-2), Term (3-3) can be expressed by:

$$\begin{aligned}
&\frac{\partial\left(\sum_{i=1}^{n+1}\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)-\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)\right)}{\partial \mathbf{x}}\\
&=\sum_{i=1}^{n+1}\left(\frac{\partial\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)-\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)}{\partial \mathbf{x}}\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)+\frac{\partial\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)}{\partial \mathbf{x}}\cdot\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)-\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)\right)\\
&=\sum_{i=1}^{n+1}\left(\frac{\partial\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)\right)}{\partial \mathbf{x}}\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)-\frac{\partial\left(\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)}{\partial \mathbf{x}}\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)+\frac{\partial\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)}{\partial \mathbf{x}}\cdot\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)-\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)\right)\\
&=\sum_{i=1}^{n+1}\left(\left(\phi(\frac{y_1^{(i-1)}-\mu_1}{\sigma_1})\cdot\frac{\partial \sigma_1}{\partial \mathbf{x}}-\Phi(\frac{y_1^{(i-1)}-\mu_1}{\sigma_1})\cdot\frac{\partial \mu_1}{\partial \mathbf{x}}\right)\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)-\left([\frac{y_1^{(i)}-y_1^{(i-1)}}{\sigma_1}\cdot\phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})-\Phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})]\cdot\frac{\partial \mu_1}{\partial \mathbf{x}}+[\phi(\frac{y_1^{(i)}-\mu_1}{\sigma_1})\cdot(1+\frac{(y_1^{(i)}-\mu_1)(y_1^{(i)}-y_1^{(i-1)})}{\sigma_1^2})]\cdot\frac{\partial \sigma_1}{\partial \mathbf{x}}\right)\cdot\Psi(y_2^{(i)},y_2^{(i)},\mu_2,\sigma_2)+\left(\phi(\frac{y_2^{(i)}-\mu_2}{\sigma_2})\cdot\frac{\partial \sigma_2}{\partial \mathbf{x}}-\Phi(\frac{y_2^{(i)}-\mu_2}{\sigma_2})\cdot\frac{\partial \mu_2}{\partial \mathbf{x}}\right)\cdot\left(\Psi(y_1^{(i-1)},y_1^{(i-1)},\mu_1,\sigma_1)-\Psi(y_1^{(i-1)},y_1^{(i)},\mu_1,\sigma_1)\right)\right)
\end{aligned} \tag{3-7}$$

Then, the EHVIG is the sum of Terms (3-6) and (3-7). In these two Terms, $\frac{\partial \mu_i}{\partial \mathbf{x}}$ and $\frac{\partial \sigma_i}{\partial \mathbf{x}}$ $(i=1,2)$ are the first order derivatives of the Kriging predictive means and standard deviations at a target point $\mathbf{x}$, respectively. These parameters can be precisely calculated by the following equations [34,35]:

$$\frac{\partial \mu}{\partial \mathbf{x}} = \frac{\partial \mathbf{c}^\top}{\partial \mathbf{x}}\mathbf{\Sigma}^{-1}\left(\mathbf{y}-\frac{\mathbf{1}_n^\top\Sigma^{-1}\mathbf{y}}{\mathbf{1}_n^\top\Sigma^{-1}\mathbf{1}_n}\mathbf{1}_n\right) \tag{3-8}$$

$$\frac{\partial \sigma}{\partial \mathbf{x}} = -\frac{1}{\sigma}\frac{\partial \mathbf{c}^\top}{\partial \mathbf{x}}\mathbf{\Sigma}^{-1}\left(\mathbf{c}-\frac{1-\mathbf{1}_n^\top\Sigma^{-1}\mathbf{c}}{\mathbf{1}_n^\top\Sigma^{-1}\mathbf{1}_n}\mathbf{1}_n\right) \tag{3-9}$$

$$\text{where}\frac{\partial \mathbf{c}}{\partial \mathbf{x}} = 2\,\text{diag}(\theta_1,\ldots,\theta_n)\cdot[R(\mathbf{x},\mathbf{x}_1)(\mathbf{x}_1,\mathbf{x}),\ldots,R(\mathbf{x},\mathbf{x}_n)(\mathbf{x}_n,\mathbf{x})] \tag{3-10}$$

Compared to the final expression of 2-D EHVI in Equation (2-20), the final expression of 2-D EHVIG also consists of two terms, Term (3-6) and Term (3-7). Moreover, the number of the integration stripes both in EHVIG and EHVI is $n+1$, as we are using the EHVI partitioning method in EHVIG. Therefore, the computational complexity of 2-D EHVIG is equal to the complexity of EHVI, that is $O(n \log n)$. For the detailed proof of 2-D EHVI computational complexity, see Ref. [12]. Note, that this $O(n \log n)$ complexity does not include the time required for computing $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, which was discussed earlier and depends on the surrogate modelling approach.

3.3. Performance assessment

The performance assessment of the EHVIG will be illustrated by a single numerical experiment. The bi-criteria optimization problem is:

$y_1(\mathbf{x}) = \|\mathbf{x}-\mathbf{1}\| \to \min$, $y_2(\mathbf{x}) = \|\mathbf{x}+\mathbf{1}\| \to \min$, $\mathbf{x} \in [-1,6]\times[-1,6] \subset \mathbb{R}^2$ [12]. Fig. 3 shows the landscape of EHVIG, in which the evaluated points are marked by blue circles. The EHVIG calculated by the exact method,[5] which uses EHVIG formula in section 3.2, is indicated by black arrows in the left figure. The EHVIG calculated by the numerical method, which meshgrids a search space and calculate the difference

[5] The MATLAB source code for computing the EHVIG for 2-D case is available on moda.liacs.nl or on request from the authors.

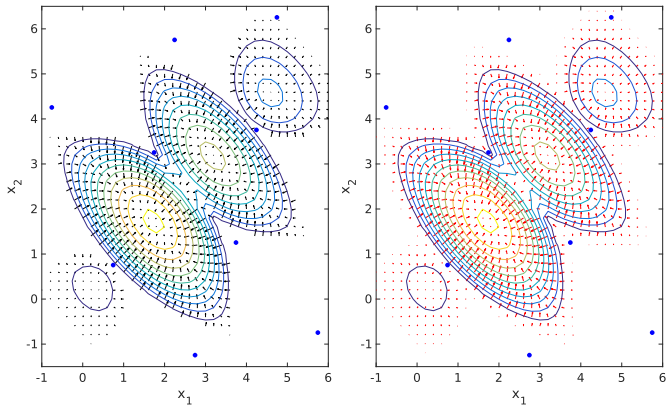

Fig. 3. The landscape of EHVIG. Left: computed using exact calculation algorithm, Right: computed using numerical calculation method.

of the EHVI values between numerical differential two points, is indicated by the red arrows in the right figure. The landscapes of EHVIG in both figures are very similar, however, there exist some slight differences between them, while very small and caused by numerical errors.

4. Multi-Objective Bayesian Global Optimization based on EHVIG

Similar to BGO, Multi-Objective Bayesian Global Optimization (MOBGO) is also based on Kriging theory. MOBGO assumes that d objective functions are mutually independent in an objective space. In MOBGO, the Kriging method or Gaussian process can approximate the objective functions and quantify the uncertainties of the prediction by using Kriging models, which are determined by the existing evaluation data $D = \left((\mathbf{x}^{(1)}, \mathbf{y}^{(1)} = Y(\mathbf{x}^{(1)})), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)} = Y(\mathbf{x}^{(\mu)}))\right)$. Each objective function at a given point $\mathbf{x}^{(t)}$ is approximated by a one-dimensional normal distribution, with mean μ and standard deviation σ. Then MOBGO can predict the multivariate outputs by means of an independent joint normal distribution with parameters $\mu_1, \ldots, \mu_d$ and $\sigma_1, \ldots, \sigma_d$ at the point $\mathbf{x}^{(t)}$.

These predictive means and standard deviations can be used to calculate infill criteria. An *infill criterion* measures how promising a new point is when compared to a current Pareto-front approximation. With the assistance of a single objective optimization algorithm, the 'optimal' solution $\mathbf{x}^*$ can be found according to the score of the infill criterion. This score of the infill criterion is calculated by the predictions of the Kriging models, instead of by the direct evaluations of the objective functions. Subsequently, the algorithm evaluates the 'optimal' solution $\mathbf{x}^*$, and both the dataset D and the Pareto-front approximation set $\boldsymbol{\mathcal{P}}$ are updated.

The basic structure of the MOBGO algorithm is shown in Algorithm 1. Note that only one criterion C is chosen in a certain MOBGO, and this criterion defines the variations of MOBGO in Algorithm 1 line 8. Some common infill criteria are: *Probability of Improvement* (PoI), EHVI and *Hypervolume Improvement* (HVI)·In this paper, the infill criterion is EHVI. Here, *opt* is a search algorithm which finds the optimal solution x^* by maximizing the EHVI.

4.1. Gradient ascent algorithms

Previously, the optimizer *opt* in Algorithm 1 was chosen as CMA-ES [36], which is a state-of-the-art heuristic global optimization algorithm. Since the formula of 2-D EHVIG is derived in this paper, a gradient ascent algorithm can replace CMA-ES to speed up the process of finding a promising point x^*.

Many gradient ascent algorithms (GAAs) exist. The conjugate gradient algorithm is one of the most efficient algorithms among them. However, it cannot solve the constrained problems, and this is the reason why we exclude it in this paper. For the other GAAs, the general formula for computing the next solution is:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + s \cdot \nabla F(\mathbf{x}^{(t)}) \tag{4-1}$$

where $\mathbf{x}^{(t)}$ is the current solution, $\mathbf{x}^{(t+1)}$ is the updated solution, s is the stepsize, and $\nabla F(\cdot)$ is the gradient of the objective functions or of the infill criterion. In this paper, ∇F is EHVIG.

Another important aspect is that the starting point is crucial to the performance of GAAs. To improve the probability of finding the globally optimal location, CMA-ES was used to initialize the starting points in this paper. The structure of gradient ascent based search algorithm is shown in Algorithm 2.

4.2. EHVIG as a stopping criterion for CMA-ES

Traditionally, when EAs are searching for the promising point $\mathbf{x}^*$, convergence velocity and some other statistical criteria are used to determine whether the EAs should stop/restart or not. These criteria can balance the quality of the performance and efficiency of the execution time to some degree, but not optimally. Because all these criteria are blind to whether an individual is already the optimal or not.

Considering that the gradient of the promising point in the search space should be the zero vector and EHVIG can be exactly calculated, EHVIG can be used as a stopping/restart criterion in EAs when they are searching for the optimal point with the EHVI as the infill criterion. Theoretically speaking, the EHVI should be maximized during the procedure. Therefore, for this method it is necessary to use, for instance, information about the second derivative of the EHVI at this point, in order to determine the optimality and the type of optimality. However, this is omitted due to the complexities. The structure of CMA-ES assisted by EHVIG is shown in Algorithm 3.

5. Empirical experiments

The benchmarks were well-known test problems: BK1 [37], SSFYY1 [38], ZDT1, ZDT2, ZDT3 [39], the generalized Schaffer problem [40] with different parameter settings for γ (γ in GSP and GSP12 were 0.4 and 1.2, respectively), and three proportional-integral-derivative (PID) parameter tuning problems [41–43].

5.1. Test problem 1 - Robust PID parameter tuning

A PID controller is a control loop feedback mechanism, and it is widely applied in industrial control applications. The structure of the feedback controller is shown in Fig. 4, where $R(s)$ is the reference input signal, $E(s)$ represents error signal, $C(s)$ is the transfer function of the controller, $U(s)$ is control signal, $P(s)$ stands for controlled plant, $\triangle P(s)$ is the plant perturbation, $d(t)$ is the external disturbance and $Y(s)$ is the output of the system. For the PID controller, three parameters are part of $C(s)$: proportionality $B2$, integral $B1$ and derivative $B0$, and the transfer function of the PID controller for a continuous system can be defined as: $C(s) = \frac{B_2 s^2 + B_1 s + B_0}{s}$. The basic idea of a PID controller is to attempt to minimize an error ($E(s)$) by adjusting the process control inputs.

The benchmark for PID parameter tuning is taken from Ref. [41,44]. The transfer function of the plant is given as follows:

$$P(S) = \begin{pmatrix} \frac{-33.98}{(98.02s+1)(0.42s+1)} & \frac{32.63}{(99.6s+1)(0.35s+1)} \\ \frac{-18.85}{(75.43s+1)(0.30s+1)} & \frac{34.84}{(110.5s+1)(0.03s+1)} \end{pmatrix} \tag{5-1}$$

Two criteria were used in this paper: balanced performance criterion $J_\infty = (J_a^2 + J_b^2)^{1/2}$ [45] and interval squared error $J_2 = \int_0^\infty e^\top(t)e(t)dt$. For J_∞, J_a and J_b are defined as follows: $J_a^2 = \|W_1(s)T(s)\|_\infty$, $J_b^2 =$

Algorithm 1 MOBGO algorithm.

Input: Objective functions $\mathbf{y}$, initialization size μ, termination criterion T_c
Output: Pareto-front approximation $\mathcal{P}$

Initialize μ points $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$;
Evaluate the initial set of μ points: $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$;
Store $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(\mu)})$ and $(\mathbf{y}^{(1)} = \mathbf{y}(\mathbf{x}^{(1)}), \ldots, \mathbf{y}^{(\mu)} = \mathbf{y}(\mathbf{x}^{(\mu)}))$ in D: $D = ((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}))$;
Compute the non-dominated subset of D and store it in $\mathcal{P}$;
$g = \mu$;
while $g <= T_c$ **do**
  Train Kriging models $M_1, \cdots, M_d$ based on D ;
  Use an optimizer (opt) to find the promising point $\mathbf{x}^*$ based on surrogate models M, with the infill criterion C;
  Update D: $D = D \cup (\mathbf{x}^*, \mathbf{y}(\mathbf{x}^*))$;
  Update $\mathcal{P}$ as the non-dominated subset of D;
  $g = g + 1$;
Return $\mathcal{P}$

Algorithm 2 Gradient ascent based search algorithm.

Input: Kriging Models $M_1, \cdots, M_d$, Pareto-front approximation $\mathcal{P}$, reference point $\mathbf{r}$, number of clusters N_c
Output: Optimal solution $\mathbf{x}^*$

Initialize λ points using CMA-ES with 15 iterations;
Cluster λ points into N_c clusters $G_1, \cdots, G_{N_c}$;
for $i = 1$ *to* $i <= N_c$ **do**
  Update starting point $\mathbf{x}^s$, $\mathbf{x}^s = mean(G_i)$;
  Calculate the promising point $\mathbf{x}^{*i}$ using simple gradient ascent algorithm and the starting point $\mathbf{x}^s$;
  Calculate the corresponding EHVI value $EHVI^i$
Find the promising point $\mathbf{x}^*$ among $\mathbf{x}^{*1}, \cdots, \mathbf{x}^{*N_c}$;
Return $\mathbf{x}^*$;

$\|W_2(s)S(s)\|_\infty$. Here, $W_1(s)$ is the assumed boundary of plant perturbation $\triangle P(s)$, $W_2(s)$ is a stable weighting function matrix and they are defined in Ref. [45]:

$$W_1(s) = \frac{100s+1}{s+1000} \times I_{2\times 2}, \tag{5-2}$$

$$W_2(s) = \frac{s+1000}{1000s+1} \times I_{2\times 2}. \tag{5-3}$$

$T(s)$ and $S(s)$ are the sensitivity and complementary sensitivity functions of the system, respectively, and they can be calculated by:

$$S(s) = (I + P(s)C(s))^{-1}, \tag{5-4}$$

$$T(s) = P(s)C(s)(I + P(s)C(s))^{-1}. \tag{5-5}$$

5.2. Test problem 2 - PID parameter tuning problem

This benchmark on PID parameter tuning is taken from Ref. [43]. The three parameters for the PID controller are: proportionality K_p, integral K_i and derivative K_d. The transfer function of PID controller for a continuous system can be defined as: $Y(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$. The process of PID controller can be described as follows: when a setpoint is set or $E(s)$ exists, $E(s)$ will be calculated by the difference between the setpoint and actual output, and a PID controller will generate a new control signal ($U(s)$) based on $E(s)$. Then the new control signal $U(s)$ is applied to the plant model, and the new actual output and $E(s)$ are generated again. The structure of a PID control is shown in Fig. 5.

The chosen transfer functions modelling the plant in this paper are:

$$G_1(s) = \frac{25.2s^2 + 21.2s + 3}{s^5 + 16.58s^4 + 25.41s^3 + 17.18s^2 + 11.70s + 1} [42] \tag{5-6}$$

Algorithm 3 CMA-ES assisted by EHVIG.

Input: Kriging Models $M_1, \cdots, M_d$, Pareto-front approximation $\mathcal{P}$, reference point $\mathbf{r}$, restart number N_r
Output: Optimal solution $\mathbf{x}^*$

for $i = 1$ *to* $i <= N_r$ **do**
  Initialize parameters in CMA-ES;
  $flag = 1$;
  for $flag \geq \epsilon$ **do**
    Get offspring by normal CMA-ES with default parameters;
    Select the best individual $\mathbf{x}^{*i}$ from the offspring, whose EHVI value is maximum;
    Predict the mean value $\boldsymbol{\mu}^*$ and the standard deviation $\boldsymbol{\sigma}^*$ at $\mathbf{x}^{*i}$;
    $flag = Sum(|EHVIG(\mathbf{x}^{*i}, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^*, \mathcal{P}, \mathbf{r})|)$;
Find the promising point $\mathbf{x}^*$ among $\mathbf{x}^{*1}, \cdots, \mathbf{x}^{*N_r}$;
Return $\mathbf{x}^*$;

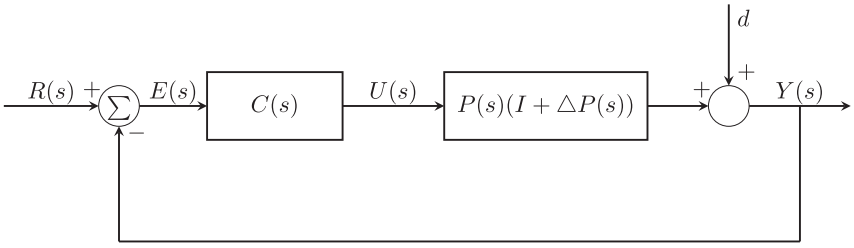

Fig. 4. Feedback control system with plant perturbation and external disturbance.

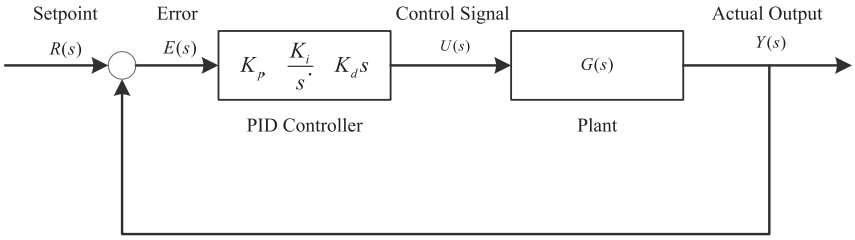

Fig. 5. The structure of PID control.

$$G_2(s) = \frac{4.228}{(s+0.5)(s^2+1.64s+8.456)}[43] \quad (5\text{-}7)$$

The step response of these two plants is analyzed with the criteria of *settling time* (t_s) and *percentage overshoot* (*PO*). *Settling time* (t_s) is defined as time elapsed from the application of an ideal instantaneous step input to the time, at which the output has entered error band with 2% in this paper, while *percentage overshoot* (*PO*) refers to the percentage of an output exceeding its final steady-state value.

5.3. Experimental settings

All the benchmarks in this paper were employed by using different search strategies in MOBGO and some well-known evolutionary multi-objective optimization algorithms (EMOAs), including NSGA-II [46], SMS-EMOA [23] and MOEA/D [47]. All the parameter settings for both MOBGO based algorithms and EMOAs are shown in Table 2. Here, the termination criterion T_c refers to the number of function evaluations. The hyperparameter θ for the correlation function in Equation (2-3) was optimized by the simplex search method of Lagarias et al. (*fminsearch*) [48], with the parameter of 1000 for the max evaluations of the likelihood functions and the search space of $0 \le \theta \le 10^{10}$. In EMOAs, the unmentioned parameters in Table 2 were set as the default.

The selection of a reference point is tricky. The Pareto-front approximation set will focus on the extreme points if a large reference point is selected, and it will concentrate on a knee point if a small reference is set. All the reference points in this paper were chosen according to the suggested reference points in the referenced articles, as indicated in Table 1.

Each trail was repeated for ten times. All the experiments were finished on the same computer: Intel(R) i7-3770 CPU @ 3.40 GHz, RAM 16 GB. The operating system was Ubuntu 16.04 LTS (64 bit), and the platform was MATLAB 8.4.0.150421 (R2014b), 64 bit.

In GAAs, one of the main tasks is how to control the stepsize *s*. A major concern of this paper is to demonstrate GAAs can be utilized with the assistance of EHVIG in MOBGO, instead of proposing a good GAA, a fixed stepsize control strategy and the simplest parameters were applied in GAA. The parameters of the GAA in Alg. 3 are: stepsize $s = 0.01$ and the max iteration number is 1000.

5.4. Results

Table 3 shows the final experimental results. The performances of each algorithm are evaluated by HV and execution time. The highest value of HV on each test problem is indicated in bold, and the smallest value of the standard deviation of HV is also shown in bold. For the execution time (ET, unit: minutes), both the least execution time and smallest standard deviation of time, among Alg. 1, Alg. 2 and Alg. 3 are indicated in bold. If an execution time of an algorithm is less than 1 min, we didn't calculate the standard deviation of the execution times and we use '-' to express this.

Here, Alg. 4 (original CMA-ES with no restart mechanism and with a max iteration of 15) is a control group for Alg. 3 to test whether the GAA works as predicted or not. Since there is no new mechanism added to Alg. 4 and max iteration is too small, the performance of Alg. 4 is indeed worse than the other three algorithms. Hence, there is no need to compare the execution time of Alg. 4 with the other algorithms.

Compared to MOBGO based algorithms, EMOAs perform worse concerning HV values in all test problems. This result is expected and reasonable because EMOAs need a large mount of function evaluations and can not generate a good Pareto-front approximation set when the number of function evaluations is only 200. Therefore, the analyses of the experimental results focus on MOBGO based algorithms.

From Table 3, it can be seen that Alg. 3, using GAA to searching for an optimal point and CMA-ES for the initialization of the starting points, can improve the final performance a little bit, compared to Alg. 4. However, it can not outperform the original CMA-ES (Alg. 1). One potential reason is related to the starting points in the GAA, that is: GAA is very sensitive to the starting point, and the starting points generated by CMA-ES with 15 iterations are located at the optimal local area. Another potential reason is that the parameters of the stepsize and the max iteration number in GAA are not well set, as no parameter tuning was done.

Compared to the original CMA-ES (Alg. 1), Alg. 2 (CMA-ES using EHVIG as the stopping criterion) outperforms Alg. 1 on BK1, SSFYY1, GSP, and GSP12. Among these four test problems, the execution time of Alg. 2 is much faster than Alg. 1 in the case of the SSFYY1 and GSP problems. When applying EHVIG as a stopping criterion in Alg. 2, algorithm CMA-ES can terminate the loop earlier when the EHVIG of one individual is a zero vector, and therefore some execution time can be saved. In other words, while the original CMA-ES does not know

Table 1
Reference points.

	BK1	SSFYY1	ZDT1	ZDT2	ZDT3	GSP	GSP12	Robust_PID	G1	G2
r	(60, 60)	(5, 5)	(11, 11)	(11, 11)	(11, 11)	(5, 5)	(5, 5)	(30, 2)	(20, 20)	(20, 20)

Table 2
Parameter settings.

MOBGOs	ϵ	N_r	Stopping Criterion	Max Iter.	GAA	N_c	μ	T_c	ref.
Alg. 1	/	3	Default	2000	No	/	30	200	[6]
Alg. 2	10^{-5}	3	EHVIG	2000	No	/	30	200	
Alg. 3	/	0	Default	15	Yes	4	30	200	
Alg. 4	/	0	Default	15	No	/	30	200	
Alg. 5	10^{-5}	3	EHVIG projection	2000	No	/	30	200	
EMOAs	μ	N_r	Stopping Criterion	Max Iter.	λ	p_c	p_m	T_c	ref.
NSGA-II	30	Default	Default	200	30	0.9	$1/N$	200	[46]
SMS-EMOA	30	Default	Default	200	/	0.9	$1/N$	200	[23]
MOEA/D	30	Default	Default	200	/	/	/	200	[47]

whether a current individual is already the optimal solution or not, EHVIG can be used as a criterion to check for this. For the BK1, GSP12 and PID problems, Alg. 2 needs more time, but the performance of Alg. 2 is better than Alg. 1.

On the ZDT series of problems, however, the performance of Alg. 2 is worse than Alg. 1. An explanation of this phenomenon is that the optimal solutions for the ZDT series of problems are located on the boundary of the search space. According to the definition of the gradient, EHVIG would be infeasible at these boundaries, and thus EHVIG would mislead CMA-ES in search of the optimal solution. A remedy to improve the performance of Alg. 2 is to apply the projection of EHVIG to check whether an individual is optimal or not on the boundaries,

Table 3
Experimental results.

			Alg. 1	Alg. 2	Alg. 3	Alg. 4	Alg. 5	MOEA/D	NSGA-II	SMS-EMOA
BK1	ET	mean	**6.2817**	13.4433	8.0933	< 1	/	< 1	< 1	< 1
		std.	**0.6480**	1.0280	0.8803	–	/	–	–	–
	HV	mean	3175.7582	**3175.9683**	3166.4668	3133.8960	/	3116.9535	2724.6070	2802.5662
		std.	0.3620	**0.2940**	3.6840	6.0266	/	16.3347	184.2896	203.0055
SSFYY1	ET	mean	13.1067	**4.7667**	7.2550	< 1	/	< 1	< 1	< 1
		std.	5.4001	**0.3306**	0.3705	–	/	–	–	–
	HV	mean	20.7096	**20.7098**	20.5474	20.0187	/	20.3103	15.3387	15.9825
		std.	0.0069	**0.0035**	0.0361	0.1284	/	0.0884	4.5380	1.8841
ZDT1	ET	mean	82.9317	76.9400	**15.0667**	6.6133	34.8383	< 1	< 1	< 1
		std.	38.5988	12.1167	**8.2437**	4.2966	14.7293	–	–	–
	HV	mean	120.6491	120.6488	120.6275	120.6268	**120.6498**	117.7104	115.2243	115.1850
		std.	0.0055	**0.0052**	0.0066	0.0069	0.0063	0.9239	0.6335	0.3815
ZDT2	ET	mean	40.3233	39.6800	**6.8889**	2.0983	33.8407	< 1	< 1	< 1
		std.	7.1394	6.1038	**0.1332**	0.1628	2.3391	–	–	–
	HV	mean	120.3025	120.2965	120.1151	119.2155	**120.3159**	113.2975	94.0143	91.4064
		std.	0.0130	**0.0067**	0.3474	2.9890	0.0127	4.4121	8.8958	3.8514
ZDT3	ET	mean	53.6267	45.9850	**8.5450**	2.8550	13.3067	< 1	< 1	< 1
		std.	8.5955	8.8638	**0.5120**	0.4217	9.0423	–	–	–
	HV	mean	**128.7486**	128.4772	127.7556	127.4168	128.6857	118.9928	104.2878	109.3214
		std.	**0.0079**	0.7747	1.2385	1.2383	0.1029	2.1111	6.2398	7.4549
GSP	ET	mean	46.4850	**7.5017**	13.3167	< 1	/	< 1	< 1	< 1
		std.	40.2517	**0.3572**	0.7771	–	/	–	–	–
	HV	mean	24.9066	**24.9066**	24.9055	24.9050	/	24.6423	24.6399	24.7962
		std.	0.0001	**0.0000**	0.0001	0.0001	/	0.0685	0.1256	0.0454
GSP12	ET	mean	20.3167	20.6650	**13.7200**	4.6867	/	< 1	< 1	< 1
		std.	**0.4215**	0.7123	0.4407	0.1403	/	–	–	–
	HV	mean	24.3914	**24.3930**	24.3883	24.3848	/	24.0930	22.2701	22.5135
		std.	0.0034	0.0019	**0.0016**	0.0013	/	0.2284	0.4931	0.4208
PID	ET	mean	129.8650	137.3000	**38.9667**	Failed	/	< 1	< 1	< 1
		std.	16.8889	13.0257	**5.3610**	Failed	/	–	–	–
	HV	mean	52.0297	**52.8901**	42.6178	Failed	/	27.8485	27.7715	25.6507
		std.	2.1025	**1.7566**	2.6531	Failed	/	0.1057	0.2081	3.5012
G1	ET	mean	38.9667	41.3889	**19.1167**	Failed	/	< 1	< 1	< 1
		std.	5.3610	**3.7302** 1	4.6981	Failed	/	–	–	–
	HV	mean	335.0914	**375.4543**	352.8091	Failed	/	233.8351	228.1784	182.7330
		std.	**6.3093**	15.2715	25.6120	Failed	/	50.1712	48.8630	77.8912
G2	ET	mean	19.7000	36.2500	**17.0167**	Failed	/	< 1	< 1	< 1
		std.	4.7891	**0.7507**	2.3632	Failed	/	–	–	–
	HV	mean	299.5460	**302.8426**	229.5980	Failed	/	180.9543	168.7046	177.4528
		std.	**3.0077**	4.0221	5.8811	Failed	/	29.2686	90.6174	74.8918

instead of EHVIG. Here, the projection of EHVIG is the orthogonal projection of EHVIG onto the active constraint boundary. Since we are only dealing with box constraints, all the components of the gradient that correspond to active boundaries in the same dimension are set to zero. In Table 3, compared to Alg. 2 in the ZDT series of problems, Alg. 5 is assisted by the projection of EHVIG and can reach Pareto-front approximations closer to the true ones with less execution time. For ZDT1 and ZDT2 problems, the average HV values of Alg. 5 are even better than Alg. 1 with less execution time.

The PID parameter (Robust_PID, G1, G2) tuning problems are much more complex than the other benchmark problems in this paper. Moreover, there is no effective optimizer in the control group (Alg. 4), as the only optimizer in Alg. 4 is CMA-ES and the maximum iteration number of CMA-ES is only 15. The Pareto-front approximation sets can not converge during the main loop of MOBGO, thus we used the word 'Failed' in Table 3 to express the failure of Alg. 4 on these problems. In contrast, Alg. 1 and Alg. 3 produce feasible solutions on these three problems, Alg. 2 outperforms the other MOBGO based algorithms.

Compared to the performance of Alg. 1, Alg. 2 and its extension (Alg. 5 for the ZDT series of problems) outperform Alg. 1 with respect to the HV value and execution time on simple test problems. For the more complex problems, Alg. 2 consumes more execution time than Alg. 1, but can generate better Pareto-front approximation sets than Alg. 1.

6. Conclusions and future work

This paper introduced an efficient algorithm to exactly calculate the 2-D EHVIG and applied EHVIG in MOGBO using two different strategies in the process of searching for the optimal solution: using EHVIG as a stopping criterion in the original CMA-ES and applying it in a GAA (CMA-ES used here to initialize the starting points).

The empirical, experimental results show that MOBGO based algorithms perform much better than EMOAs, when a small amount of evaluations is considered. Among the different strategies of the optimizer in MOBGO, the GAA is much faster than original CMA-ES, but it has an obvious drawback: it gets easily stuck at stationary points, that are local optima or saddle points. Compared to the original CMA-ES, the GAA fails to outperform CMA-ES in most test problems because it is very easy to get stuck at stationary points and the parameters of GAA are not well tuned in this paper.

Another strategy proposed in this paper, is taking EHVIG as the stopping criterion in CMA-ES. The experimental results show that this method can improve the quality of the final Pareto front and reduce some execution time, compared to the original CMA-ES on problems whose optimal points are not at the boundaries in the search space. This strategy does not work on the ZDT series of problems because EHVIG cannot be calculated at the boundaries of the search space. However, a useful remedy to these problems is the projection of EHVIG.

Considering the good performance of the second strategy, for the optimizer in MOBGO, it is recommended to use EHVIG as a stopping criterion in EAs (like CMA-ES, GA). For future works, extending EHVIG from the 2-D case to higher dimensional cases is highly recommended, and the GAA in MOBGO based algorithms using EHVIG should be improved.

Acknowledgements

Kaifeng Yang acknowledges financial support from the China Scholarship Council (CSC), CSC No. 201306370037.

Appendix

$$1.\phi'(x) = -x\phi(x) \tag{A-1}$$

$$2.\Phi'(x) = \phi(x) \tag{A-2}$$

$$3.\frac{\partial\Phi(\frac{y-\mu}{\sigma})}{\partial\mathbf{x}} = \phi(\frac{y-\mu}{\sigma})\cdot(\frac{\mu-y}{\sigma^2}\cdot\frac{\partial\sigma}{\partial\mathbf{x}} - \frac{1}{\sigma}\cdot\frac{\partial\mu}{\partial\mathbf{x}}) \tag{A-3}$$

Using the chain rule and quotient rule, considering that y does not depend on x, we get the statement in (A-3):

$$\frac{\partial\Phi(\frac{y-\mu}{\sigma})}{\partial\mathbf{x}} = \phi(\frac{y-\mu}{\sigma})\cdot\frac{\partial(\frac{y-\mu}{\sigma})}{\partial\mathbf{x}} = \phi(\frac{y-\mu}{\sigma})\cdot\frac{(\frac{\partial y}{\partial\mathbf{x}} - \frac{\partial\mu}{\partial\mathbf{x}})\sigma - (y-\mu)\frac{\partial\sigma}{\partial\mathbf{x}}}{\sigma^2}$$

After tidying up, we get statement in (A-3):

$$\frac{\partial\Phi(\frac{y-\mu}{\sigma})}{\partial\mathbf{x}} = \phi(\frac{y-\mu}{\sigma})\cdot(\frac{\mu-y}{\sigma^2}\cdot\frac{\partial\sigma}{\partial\mathbf{x}} - \frac{1}{\sigma}\cdot\frac{\partial\mu}{\partial\mathbf{x}})$$

4.

$$\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mathbf{x}} = \left(\frac{b-a}{\sigma}\cdot\phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma})\right)\cdot\frac{\partial\mu}{\partial\mathbf{x}} + \phi(\frac{b-\mu}{\sigma})\cdot\left(1 + \frac{(b-\mu)(b-a)}{\sigma^2}\right)\cdot\frac{\partial\sigma}{\partial\mathbf{x}} \tag{A-4}$$

Using the product rule and considering a and b do not depend on $\mathbf{x}$, we get the statement:

$$\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mathbf{x}} = \frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mu}\cdot\frac{\partial\mu}{\partial\mathbf{x}} + \frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\sigma}\cdot\frac{\partial\sigma}{\partial\mathbf{x}} \tag{A-5}$$

Substituting Equation (2-14) into $\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mu}$ and $\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\sigma}$, using the chain rule, quotient rule, and product rule, the expressions for $\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mu}$ and $\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\sigma}$ are:

$$\frac{\partial\Psi(a,b,\mu,\sigma)}{\partial\mu} = \frac{\partial[\sigma\cdot\phi(\frac{b-\mu}{\sigma}) + (a-\mu)\cdot\Phi(\frac{b-\mu}{\sigma})]}{\partial\mu}$$

$$= \sigma\cdot\frac{\partial\phi(\frac{b-\mu}{\sigma})}{\partial\mu} + (-1)\cdot\Phi(\frac{b-\mu}{\sigma}) + (a-\mu)\cdot\frac{\partial\Phi(\frac{b-\mu}{\sigma})}{\partial\mu}$$

$$= \frac{b-\mu}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) + [-\frac{a-\mu}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma})]$$
$$= \frac{b-a}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma}) \tag{A-6}$$

$$\frac{\partial \Psi(a,b,\mu,\sigma)}{\partial \sigma} = \frac{\partial[\sigma \cdot \phi(\frac{b-\mu}{\sigma}) + (a-\mu) \cdot \Phi(\frac{b-\mu}{\sigma})]}{\partial \sigma}$$
$$= \phi(\frac{b-\mu}{\sigma}) + \sigma \cdot \frac{\partial \phi(\frac{b-\mu}{\sigma})}{\partial \sigma} + (a-\mu) \cdot \frac{\partial \Phi(\frac{b-\mu}{\sigma})}{\partial \sigma}$$
$$= \phi(\frac{b-\mu}{\sigma}) + (\frac{b-\mu}{\sigma})^2 \cdot \phi(\frac{b-\mu}{\sigma}) + \left(-\frac{(a-\mu)\cdot(b-\mu)}{\sigma^2} \cdot \phi(\frac{b-\mu}{\sigma})\right)$$
$$= \phi(\frac{b-\mu}{\sigma}) + \frac{(b-\mu)\cdot(b-a)}{\sigma^2} \cdot \phi(\frac{b-\mu}{\sigma})$$
$$= \phi(\frac{b-\mu}{\sigma}) \left(1 + \frac{(b-\mu)\cdot(b-a)}{\sigma^2}\right) \tag{A-7}$$

After substituting Equations (A-6) and (A-7) into (A-5), we get formula in (A-4):

$$\frac{\partial \Psi(a,b,\mu,\sigma)}{\partial \mathbf{x}} = \left(\frac{b-a}{\sigma} \cdot \phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{b-\mu}{\sigma})\right) \cdot \frac{\partial \mu}{\partial \mathbf{x}} + \phi(\frac{b-\mu}{\sigma}) \cdot \left(1 + \frac{(b-\mu)(b-a)}{\sigma^2}\right) \cdot \frac{\partial \sigma}{\partial \mathbf{x}} \tag{A-8}$$

References

[1] Y. Jin, Surrogate-assisted evolutionary computation: recent advances and future challenges, Swarm Evol. Comput. 1 (2) (2011) 61–70.

[2] J. Močkus, On Bayesian Methods for Seeking the Extremum, Springer Berlin Heidelberg, Berlin, Heidelberg, 1975, pp. 400–404. https://doi.org/10.1007/3-540-07165-2_55.

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

[4] M. Emmerich, Single-and Multi-objective Evolutionary Design Optimization Assisted by Gaussian Random Field Metamodels, Ph.D. thesis, Fachbereich Informatik, Chair of Systems Analysis, University of Dortmund, October 2005.

[5] I. Couckuyt, D. Deschrijver, T. Dhaene, Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization, J. Global Optim. 60 (3) (2014) 575–594.

[6] K. Yang, D. Gaida, T. Bäck, M. Emmerich, Expected hypervolume improvement algorithm for PID controller tuning and the multiobjective dynamical control of a biogas plant, in: 2015 IEEE Congress on Evolutionary Computation (CEC), 2015, pp. 1934–1942, https://doi.org/10.1109/CEC.2015.7257122.

[7] C. Luo, K. Shimoyama, S. Obayashi, Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1187–1194, https://doi.org/10.1109/CEC.2014.6900299.

[8] K. Shimoyama, S. Jeong, S. Obayashi, Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems, in: 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2013, pp. 658–665.

[9] M. Emmerich, K.C. Giannakoglou, B. Naujoks, Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels, IEEE Trans. Evol. Comput. 10 (4) (2006) 421–439.

[10] M. Emmerich, A.H. Deutz, J.W. Klinkenberg, Hypervolume-based expected improvement: monotonicity properties and exact computation, in: 2011 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2011, pp. 2147–2154.

[11] I. Hupkens, A. Deutz, K. Yang, M. Emmerich, Faster exact algorithms for computing expected hypervolume improvement, in: A. Gaspar-Cunha, C. Henggeler Antunes, C.C. Coello (Eds.), International Conference on Evolutionary Multi-criterion Optimization, Springer, Cham, 2015, pp. 65–79.

[12] M. Emmerich, K. Yang, A. Deutz, H. Wang, C.M. 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.

[13] K. Yang, M. Emmerich, A. Deutz, C.M. Fonseca, Computing 3-D Expected Hypervolume Improvement and Related Integrals in Asymptotically Optimal Time, Springer International Publishing, Cham, 2017, pp. 685–700. https://doi.org/10.1007/978-3-319-54157-0_46.

[14] E. Vazquez, J. Bect, Convergence properties of the expected improvement algorithm with fixed mean and covariance functions, J. Stat. Plann. Inference 140 (11) (2010) 3088–3095.

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

[16] J. Mockus, Bayesian Approach to Global Optimization: Theory and Applications, vol. 37, Springer Science & Business Media, 2012.

[17] A. Torn, A. Žilinskas, Global Optimization, Springer-Verlag New York, Inc., 1989.

[18] R. Li, M.T.M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, T. Back, J. Dijkstra, J.H.C. Reiber, Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 2764–2771, https://doi.org/10.1109/CEC.2008.4631169.

[19] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. (1989) 409–423.

[20] T. Chugh, Handling Expensive Multiobjective Optimization Problems with Evolutionary Algorithms, Ph.D. thesis, Faculty of Information Technology, University of Jyväskylä, June 2017.

[21] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms a comparative case study, in: A.E. Eiben, T. Bäck, M. Schoenauer, H.-P. Schwefel (Eds.), International Conference on Parallel Problem Solving from Nature–PPSN V, Springer, Berlin, Heidelberg, 1998, pp. 292–301.

[22] M. Emmerich, N. Beume, B. Naujoks, An EMO algorithm using the hypervolume measure as selection criterion, in: C.A.C. Coello, A.H. Aguirre, E. Zitzler (Eds.), International Conference on Evolutionary Multi-criterion Optimization, Springer, Berlin, Heidelberg, 2005, pp. 62–76.

[23] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669.

[24] J.E. Dennis, V. Torczon, Managing approximation models in optimization, in: N. Alexandrov, M.Y. Hussaini (Eds.), Multidisciplinary Design Optimization: State-of-the-art, SIAM, Philadelphia, PA, 1997, pp. 330–347.

[25] M. Fleischer, The measure of Pareto optima applications to multi-objective metaheuristics, in: C.M. Fonseca, P.J. Fleming, E. Zitzler, L. Thiele, K. Deb (Eds.), International Conference on Evolutionary Multi-criterion Optimization, Springer, Berlin, Heidelberg, 2003, pp. 519–533.

[26] T. Wagner, M. Emmerich, A. Deutz, W. Ponweiser, On expected-improvement criteria for model-based multi-objective optimization, in: R. Schaefer, C. Cotta, J. Kołodziej, G. Rudolph (Eds.), International Conference on Parallel Problem Solving from Nature–PPSN XI, Springer, Berlin, Heidelberg, 2010, pp. 718–727.

[27] H.J. Kushner, A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise, J. Basic Eng. 86 (1) (1964) 97–106.

[28] K. Yang, A. Deutz, Z. Yang, T. Back, M. Emmerich, Truncated expected hypervolume improvement: exact computation and application, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 4350–4357, https://doi.org/10.1109/CEC.2016.7744343.

[29] K. Yang, L. Li, A. Deutz, T. Bäck, M. Emmerich, Preference-based multiobjective optimization using truncated expected hypervolume improvement, in: 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE, 2016, pp. 276–281, https://doi.org/10.1109/FSKD.2016.7603186.

[30] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.

[31] N. Beume, C.M. Fonseca, M. López-Ibáñez, L. Paquete, J. Vahrenhold, On the complexity of computing the hypervolume indicator, IEEE Trans. Evol. Comput. 13 (5) (2009) 1075–1082.

[32] T.M. Chan, Klee's measure problem made easy, in: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2013, pp. 410–419.

[33] M.T.M. Emmerich, C.M. Fonseca, Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results, in: R.H.C. Takahashi, K. Deb, E.F. Wanner, S. Greco (Eds.), International Conference on Evolutionary Multi-criterion Optimization, Springer, Berlin, Heidelberg, 2011, pp. 121–135.

[34] H. B. Nielsen, S. N. Lophaven, J. Sŕndergaard, DACE, a MATLAB Kriging Toolbox, Informatics and Mathematical Modelling. Lyngby–Denmark: Technical University of Denmark, DTU.

[35] H. Wang, M. Emmerich, T. Back, Balancing risk and expected gain in kriging-based global optimization, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 719–727.

[36] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18.
[37] T.T. Binh, U. Korn, An evolution strategy for the multiobjective optimization, in: The Second International Conference on Genetic Algorithms (Mendel 96), Brno, Czech Republic, 1996, pp. 23–28.
[38] M.-B. Shim, M.-W. Suh, T. Furukawa, G. Yagawa, S. Yoshimura, Pareto-based continuous evolutionary algorithms for multiobjective optimization, Eng. Comput. 19 (1) (2002) 22–48.
[39] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[40] M.T. 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.
[41] B. Chen, Y. Cheng, A structure-specified H^{∞} optimal control design for practical applications: a genetic approach, IEEE Trans. Contr. Syst. Technol. 6 (6) (1998) 707–718.
[42] I. Chiha, N. Liouane, P. Borne, Tuning PID controller using multiobjective ant colony optimization, Appl. Comp. Intell. Soft Comput. 2012 (2012) 11–18. https://doi.org/10.1155/2012/536326.
[43] M.S. Saad, H. Jamaluddin, I.Z. Darus, PID controller tuning using evolutionary algorithms, WSEAS Trans. Syst. Control 7 (4) (2012) 139–149.
[44] S. Zhao, M.W. Iruthayarajan, S. Baskar, P.N. Suganthan, Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization, Inf. Sci. 181 (16) (2011) 3323–3335.
[45] S. Ho, S. Ho, M. Hung, L. Shu, H. Huang, Designing structure-specified mixed H_2H^{∞} optimal controllers using an intelligent genetic algorithm IGA, IEEE Trans. Contr. Syst. Technol. 13 (6) (2005) 1119–1124.
[46] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.-P. Schwefel (Eds.), International Conference on Parallel Problem Solving from Nature–PPSN VI, Springer, Berlin, Heidelberg, 2000, pp. 849–858.
[47] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[48] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, SIAM J. Optim. 9 (1) (1998) 112–147. https://doi.org/10.1137/S1052623496303470.