Computing 3-D Expected Hypervolume Improvement and Related Integrals in Asymptotically Optimal Time

Yang, K.; Emmerich, M.T.M.; Deutz, A.H.; Fonseca, C.M.; Trautmann, H.; Rudolph, G.; ... ; Grimme, C.

Computing 3-D Expected Hypervolume Improvement and Related Integrals in Asymptotically Optimal Time

Kaifeng Yang[1(✉)], Michael Emmerich[1], André Deutz[1], and Carlos M. Fonseca[2]

[1] LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
{k.yang,m.t.m.emmerich,a.h.deutz}@liacs.leidenuniv.nl
[2] CISUC, Department of Informatics Engineering, University of Coimbra, Polo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
cmfonsec@dei.uc.pt
http://moda.liacs.nl

Abstract. The Expected Hypervolume Improvement (EHVI) is a frequently used infill criterion in surrogate-assisted multi-criterion optimization. It needs to be frequently called during the execution of such algorithms. Despite recent advances in improving computational efficiency, its running time for three or more objectives has remained in $O(n^d)$ for $d \geq 3$, where d is the number of objective functions and n is the size of the incumbent Pareto-front approximation. This paper proposes a new integration scheme, which makes it possible to compute the EHVI in $\Theta(n \log n)$ optimal time for the important three-objective case ($d = 3$). The new scheme allows for a generalization to higher dimensions and for computing the Probability of Improvement (PoI) integral efficiently. It is shown, both theoretically and empirically, that the hidden constant in the asymptotic notation is small. Empirical speed comparisons were designed between the C++ implementations of the new algorithm (which will be in the public domain) and those recently published by competitors, on randomly-generated non-dominated fronts of size 10, 100, and 1000. The experiments include the analysis of batch computations, in which only the parameters of the probability distribution change but the incumbent Pareto-front approximation stays the same. Experimental results show that the new algorithm is always faster than the other algorithms, sometimes over 10^4 times faster.

Keywords: Expected hypervolume improvement · Time complexity · Surrogate-assisted multi-criterion optimization · Efficient global optimization · Probability of improvement

1 Introduction

Surrogate-assisted multi-criterion optimization (SAMCO) uses approximations to the objective functions in order to quickly assess the potential quality of a candidate solution. In the context of SAMCO, the *Expected Hypervolume Improvement* (EHVI) is frequently used as an infill or pre-selection criterion [1–8],

H. Trautmann et al. (Eds.): EMO 2017, LNCS 10173, pp. 685–700, 2017.
DOI: 10.1007/978-3-319-54157-0_46

and is a straightforward generalization of the single-objective *expected improvement* (EI). It is called multiple times because, in each iteration of SAMCO, either an optimum of the EI needs to be found, as in the case of *Bayesian global optimization*[1] [10], or, in *surrogate-assisted evolutionary algorithms*, it is used to pre-assess the quality of the individuals of a larger population.

Informally, the EHVI for a problem with d objectives is defined as follows [4]: Given an incumbent approximation to a Pareto front of size n, the EHVI is an integral that measures the expected increase of the *Hypervolume Indicator*, given a predictive multivariate Gaussian distribution on the d-dimensional objective space stemming, for instance, from a Gaussian process or Kriging Approximation, the outcome at a yet unevaluated design point. It, therefore, measures the expected quantity by which the Pareto-front approximation would improve, if that design point were evaluated. Both the mean values and variances of the objective function value predictions are considered.

Although the EHVI is interesting due to its theoretical properties, as shown by Wagner et al. [11], it has been critizised in the same paper for the computational effort required for its exact computation. This is a problem, as many SAMCO algorithms require a large number of EHVI computations to be performed in every iteration. Since the announcement of exact computation schemes [12], significant progress has been made, resulting in faster computation schemes [8,13,14]. Notably, in 2-D, the computation time could be improved from $O(n^3 \log n)$–using a straightforward integration scheme, with a grid partitioning and repeated hypervolume computations for each grid cell – to asymptotically optimal $\Theta(n \log n)$ in [13] using a stripe partitioning and a multi-layered integration scheme. However, despite significant efficiency improvements [8,14], the running time of the best-known algorithms for $d \geq 3$ has remained in $O(n^d)$.

This paper shows that the new integration technique proposed in [13] can be generalized to $d \geq 3$, yielding a new algorithm with asymptotically optimal time complexity in $\Theta(n \log n)$. This improves existing upper bounds by a factor of $O(n^2/\log n)$. Moreover, it is shown that similar approaches provide asymptotically optimal time algorithms for the *Probability of Improvement*, i.e. the probability that a point will be non-dominated, and the *Truncated Hypervolume Improvement* (TEHVI) [15].

The paper is structured as follows: Sect. 2 discusses the problem in the wider context of surrogate-assisted multi-criterion optimization, and Sect. 3 provides the necessary definitions. In Sect. 4 the new algorithm for 3-D EHVI calculation is discussed, including the analysis of its complexity. Experimental results on test data are reported in Sect. 5, including a speed comparison to other algorithms proposed in the literature [8,14]. Section 6 discusses how the new algorithm can be modified for solving the related problems of PoI and TEHVI. Section 7 draws the main conclusions and discusses future work.

[1] Also called Efficient Global Optimization [9].

2 Relevance and Related Work

In the context of single-objective optimization, the expected improvement was firstly proposed by Mockus et al. [10]. They assumed that the objective function, say $f : \mathbb{R}^q \to \mathbb{R}$, $q \geq 1$, is a realization of a Gaussian process with known (or estimated *a priori*) mean and covariance structure. Given that the values $f(x^{(i)})$ at sites $x^{(1)}, \ldots, x^{(t)}$ are known, e.g., from previous exact evaluations, the conditional mean $\mu(x)$ and standard deviations $\sigma(x)$ at a yet unevaluated point x can be computed. The 1-D Gaussian distribution with mean $\mu(x)$ and standard deviation $\sigma(x)$ assigns to $y \in \mathbb{R}$ the likelihood of the event $f(x) = y$. Consider the problem of maximizing $f(x)$, and a corresponding sequence of points $(x_i)_{i\in\mathbb{N}}$. Then, for a given point in time, t, we can express the improvement of a given $y \in \mathbb{R}$ as $I(y, f_{\max,t}) = \max\{0, y - f_{\max,t}\}$, where $f_{\max,t} = \max\{f(x_i)|i \in \{1, \ldots, t\}\}$. The expected improvement of x is then defined as $\mathrm{EI}(x) = \int_{y\in\mathbb{R}} I(y, f_{\max,t})\, \xi_{\mu(x),\sigma(x)}(y)\, \mathrm{d}y$, where $\xi_{\mu(x),\sigma(x)}$ is the probability density function of the 1-D Gaussian distribution conditioned on $(x_i, f(x_i))_{i\in\{1,\ldots,t\}}$.

The EI is widely used to pre-assess the quality of solutions in evolutionary and deterministic optimization with expensive black box evaluations. It was popularized by Jones et al. [9] as an infill criterion in the so-called Efficient Global Optimization (EGO) algorithm. In each iteration EGO evaluates the design point with maximal EI. Its convergence properties are discussed in [16], where a proof of global convergence under mild assumptions on the global covariance and the smoothness of the function is given. Roughly speaking, global convergence is due to the fact that EI rewards high variance and also high mean values.

Various generalizations of EI to multi-criterion optimization have been discussed in the literature, e.g., [7,17–20]. See also [11] for an overview. In the case of multiple objectives, it is possible to consider a Gaussian process model for each objective function separately and independently, resulting in a multivariate distribution with d mean values $\mu_i(x)$ and standard deviation $\sigma_i(x)$. A key question when generalizing the expected improvement is how to define improvement of a given Pareto-front approximation. In indicator-based multi-criterion optimization, the performance of a Pareto-front approximation is assessed by a unary indicator, typically the *Hypervolume Indicator*, which allows for a simple generalization of the *Expected Improvement*–the EHVI. Given $(x_i, \mathbf{f}(x_i))$, $i = 1, \cdots, t$, the incumbent Pareto-front approximation becomes $\mathbf{P}_{\max,t}$ = Non-Dominated subset of $\{\mathbf{f}(x_i)|i \in \{1, ..., t\}\}$ and its hypervolume replaces $f_{\max,t}$ in the definition of the EI. The EHVI was first proposed in [21], and since then it has been used in Evolutionary Algorithms for airfoil optimization [4] and quantum control [22], as well as in multi-criterion generalizations of Efficient Global Optimization for applications in fluid dynamics [23], event controllers in wastewater treatment [1], efficient algorithm tuning [5], electrical component design [8], and bio-fuel power-generation [3]. In all of these applications, the bi-objective EHVI was used. Due to its high computation time for problems with three and more objectives, using it as an infill criterion was not advised in such settings, and fast, but imprecise, alternatives were sought [24].

So far it has remained unknown whether the integration algorithms used in the literature achieved optimal performance. Hence, it is an important question to study whether, and to what extent, the computational efficiency of the exact computation of the EHVI can be further improved. In early work, the EHVI integral was computed by Monte Carlo simulation [21]. The first algorithm for the exact calculation of the 2-D EHVI was introduced by Emmerich et al. [12], with a computational complexity of $O(n^3 \log n)$, where n is the number of non-dominated points in the front. Couckuyt et al. [8] provided an exact EHVI calculation algorithm for $d > 2$, and, according to experimental data, this algorithm was typically much faster than the one introduced in [12], but it still had a high worst-case time complexity. Hupkens et al. [14] found algorithms for computing EHVI with the then lowest worst-case time complexity of $O(n^2)$ and $O(n^3)$, for two and three objectives, respectively. Recently, Emmerich et al. [13] proposed an asymptotically optimal algorithm for the bi-objective case, with a computational time complexity of $\Theta(n \log n)$. In 3-D, the time complexity of this problem and those of the related problems of computing the *Probability of Improvement* [8] and *Truncated Expected Hypervolume Improvement* [2] have so far remained cubic. The following sections outline algorithms that can solve these problems in optimal time $\Theta(n \log n)$.

3 Expected Hypervolume Improvement

This section formally introduces the *Expected Hypervolume Improvement* and related definitions. In the following, we consider problems with three objectives. Most of the discussion will be on mathematical objects defined in the objective space. We will denote the axes of the objective space spanned by $f_1, f_2, ..., f_d$, where $f_i : \mathbb{R}^q \to \mathbb{R}$ are the objective functions to be maximized. Points in the objective space will be denoted by $\mathbf{y}$, in case we are not interested in their preimage. We define the usual Pareto dominance order on the objective space, i.e. $\mathbf{y}^{(1)} \succ \mathbf{y}^{(2)}$, iff $\mathbf{y}^{(1)} \geq \mathbf{y}^{(2)}$ (componentwise) and $\mathbf{y}^{(1)} \neq \mathbf{y}^{(2)}$.

The *Hypervolume Indicator*, introduced in [25], is one of the most important unary indicators for evaluating the quality of a Pareto-front approximation. Its theoretical properties are discussed in [26,27]. Notably, it does not require the Pareto front to be known in advance, and its maximization leads to high quality and diverse Pareto-front approximation sets. The *Hypervolume Indicator* is defined as follows:

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

$$HV(\mathbf{P}) = \lambda_d(\cup_{\mathbf{y} \in \mathbf{P}}[\mathbf{r}, \mathbf{y}]) \tag{1}$$

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

The hypervolume measures the size of the dominated subspace bounded below by a reference point $\mathbf{r}$ – we consider maximization. This reference point needs to be provided by the user, and it should, if possible, be chosen such that it is dominated by all elements of the Pareto-front approximation sets $\mathbf{P}$ that might occur during the optimization process.

In order to generalize the *Expected Improvement* (EI) criterion, we will first generalize the concept of improvement to the multiobjective case:

Definition 2 (Hypervolume Improvement). *Given a finite collection of vectors $\mathbf{P} \subset \mathbb{R}^d$, the* Hypervolume Improvement *of a vector $\mathbf{y} \in \mathbb{R}^d$ is defined as:*

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

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

The *Expected Hypervolume Improvement* is based on the theory of the *Hypervolume Indicator*, and is a measure of how much hypervolume may improve, considering the uncertainty of the prediction. The prediction is a probability distribution that measures, for a given input vector in the decision space, the likelihood of outcomes in the objective space. Typically, Gaussian process models or Kriging models are used for prediction, delivering for each input vector an independent, multivariate probability distribution with a mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and a standard deviation vector $\boldsymbol{\sigma} \in \mathbb{R}^d$, each component of which corresponding to a particular objective function prediction.

Definition 3 (Expected Hypervolume Improvement). *Given a mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and a standard deviation vector $\boldsymbol{\sigma} \in \mathbb{R}^d$, an incumbent Pareto-front approximation $\mathbf{P} \subset \mathbb{R}^d$, and a reference point $\mathbf{r} \in \mathbb{R}^d$ the expected improvement is given by:*

$$EHVI(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{P}, \mathbf{r}) = \int_{\mathbf{y} \in \mathbb{R}^d} HVI(\mathbf{y}, \mathbf{P}, \mathbf{r}) \xi_{\boldsymbol{\mu}, \boldsymbol{\sigma}}(\mathbf{y}) \mathrm{d}y \tag{3}$$

Example 1. Figure 1(left) depicts the dominated hypervolume for a small approximation set $\mathbf{P} = (\mathbf{y}^{(1)} = (4, 4, 1),\ \mathbf{y}^{(2)} = (1, 2, 4),\ \mathbf{y}^{(3)} = (2, 1, 3))$. The volume of all slices is the 3-D *Hypervolume Indicator* of $\mathbf{P}$, with $\mathbf{r}$ being the origin of the coordinate system. The *Hypervolume Improvement* of $\mathbf{y}^{(+)} = (3, 3, 2)$ relative to $\mathbf{P}$ is given by the joint volume covered by the red slices. The *Expected Hypervolume Improvement* would average over different realizations of $\mathbf{y}^{(+)}$ following a 3-D normal distribution with mean vector (μ_1, μ_2, μ_3) and standard deviation σ_1, σ_2, and σ_3.

4 Asymptotically Optimal Algorithm for 3-D EHVI Calculation

For the convenience of illustration, the EHVI calculation in this section only involves the maximization case. The main idea of the proposed algorithm is

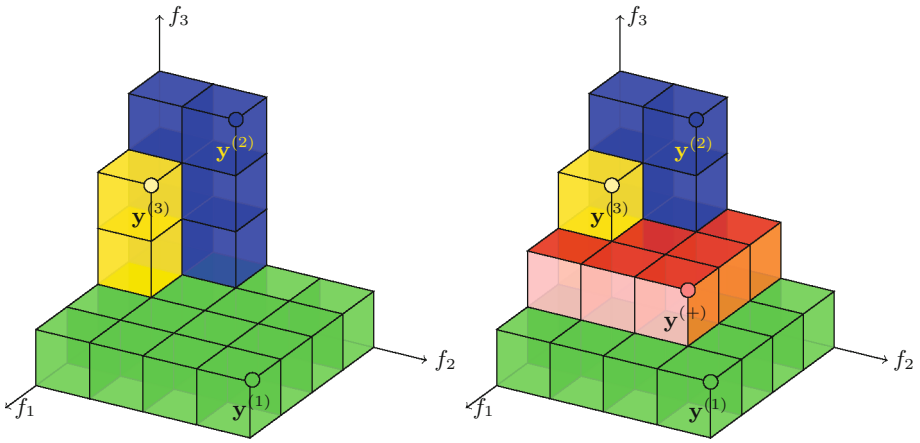

Fig. 1. The left figure shows a given 3-D Pareto-front approximation consisting of the points $\mathbf{y}^{(1)} = (4, 4, 1)$, $\mathbf{y}^{(2)} = (1, 2, 4)$, and $\mathbf{y}^{(3)} = (2, 1, 3)$. The right figure shows how the *Hypervolume Indicator* increases when the red point $\mathbf{y}^{(+)} = (3, 3, 2)$ is added. The volume of the increment (red blocks) is the *Hypervolume Improvement*. The *Expected Hypervolume Improvement* is the mean value of the *Hypervolume Improvement*, if $\mathbf{y}^{+}$ would be sampled from a 3-D normal distribution. (Color figure online)

separating the integration volume into integration slices, as few as possible, and then per integration slice compute all contributions to the EHVI integral that are related to the area of the bases of the integration slices.

4.1 Partitioning into $O(n)$ Integration Slices

Example 2. An illustration of integration slices is shown in Fig. 2. A Pareto front set is composed by 4 points ($\mathbf{y}^{(1)} = (1, 3, 4), \mathbf{y}^{(2)} = (4, 2, 3), \mathbf{y}^{(3)} = (2, 4, 2)$ and $\mathbf{y}^{(4)} = (3, 5, 1)$), and this Pareto front is shown in the left figure. The right figure illustrates the projection onto y_1y_2-plane with rectangle slices and $\mathbf{l}, \mathbf{u}$. The rectangular slices, which share the similar color but different opacity, represent integration slices with the same value of y_3 in their lower bound. The lower bound of the 3-D integration slice B_4 is $\mathbf{l}^{(4)} = (1, 2, 2)$, and the upper bound of the slice is $\mathbf{u}^{(4)} = (2, 4, \infty)$.

For maximization problems, the upper bound of each integration slice is always ∞ in the y_3 axis, therefore we can describe each integration slice by its lower bound ($\mathbf{l}$) and upper bound ($\mathbf{u}$) as follows.

$$B_i = (\mathbf{l}, \mathbf{u}) = \left(\begin{pmatrix} l_1^{(i)} \\ l_2^{(i)} \\ l_3^{(i)} \end{pmatrix}, \begin{pmatrix} u_1^{(i)} \\ u_2^{(i)} \\ \infty \end{pmatrix} \right), \qquad i = 1, \ldots, 2n + 1 \tag{4}$$

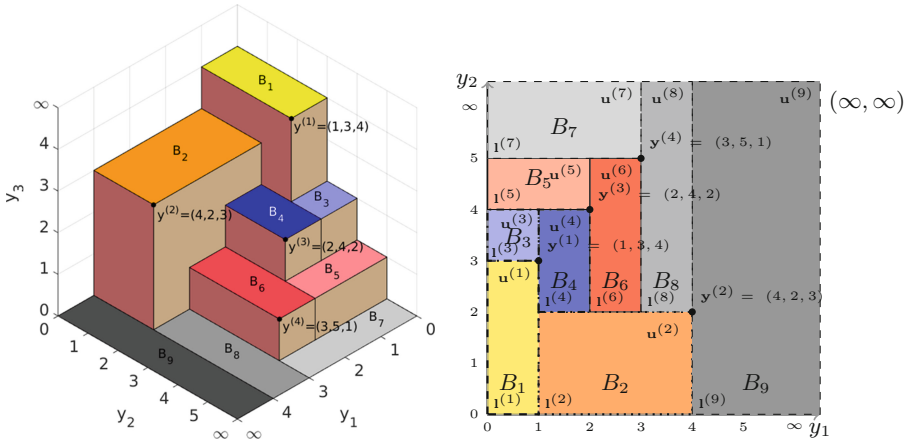

Fig. 2. Left: 3-D Pareto front. Right: 3-D Pareto front in 2-D, each slice can be described by lower bound and upper bound

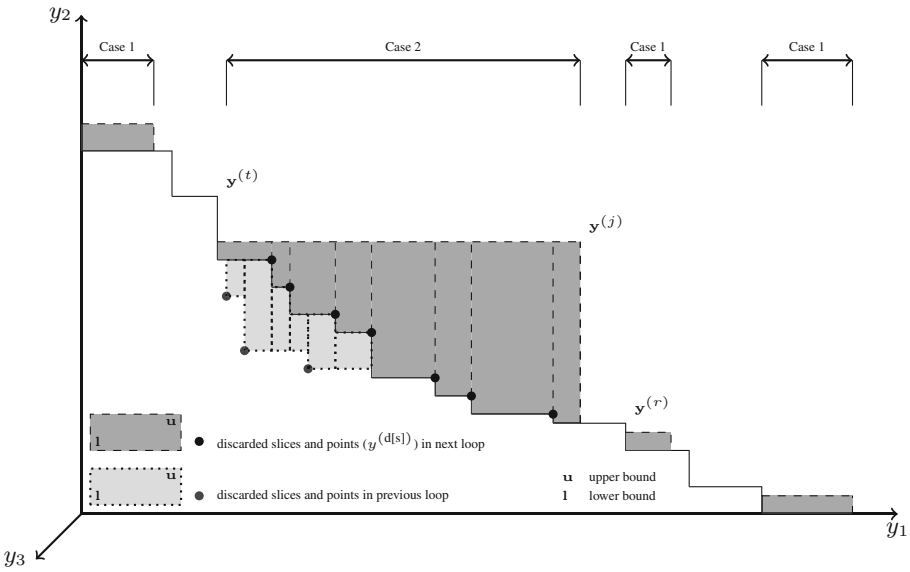

Fig. 3. Boundary search for slices in 3-D case

Algorithm 1 describes how to obtain the slices $B_1, \ldots, B_i, \ldots, B_{2n+1}$ with the corresponding lower and upper bounds ($\mathbf{l}^{(i)}$ and $\mathbf{u}^{(i)}$) and compute the integrals for them. The partitioning algorithm is similar to the sweep line algorithm described in [28]. The basic idea of this algorithm is using an AVL tree, structured by y_1 and y_2 coordinates, to process the points in a descending order of the y_3 coordinate. For each such point, say $\mathbf{y}^{(i)}$, add this point to the AVL tree

and find all the points $(\mathbf{y}^{(\mathrm{d}[1])}, \ldots, \mathbf{y}^{(\mathrm{d}[\mathrm{s}])})$ which are dominated by it in the y_1y_2-plane and discard them from the AVL tree. See Fig. 3 for describing one such iteration. In each iteration, $s+1$ slices are created by coordinates of the points $\mathbf{y}^{(\mathrm{t})}, \mathbf{y}^{(\mathrm{d}[1])}, \ldots, \mathbf{y}^{(\mathrm{d}[\mathrm{s}])}, \mathbf{y}^{(r)}$, and $\mathbf{y}^{(i)}$ as illustrated in Fig. 3. The integration of the slices (calculation_$3D$) will be described in Subsect. 4.2.

Here, the number of the integration slices is $2n+1$ when all points are in general position. (Otherwise $2n+1$ provides an upper bound for the obtained number of slices.) The reason is as follows: In the algorithm each point $\mathbf{y}^{(i)}, i = 1, \ldots, n$ creates a slice, say slice $A^{(i)}$, when it is created and a slice, say slice $B^{(i)}$, when it is discarded from the AVL tree due to domination by another point, say $\mathbf{y}^{(s)}$, in the y_1y_2-plane. The two slices are defined as follows $A^{(i)} = ((y_1^{(t)}, y_2^{(l2)}, y_3^{(i)}), (y_1^{(u1)}, y_2^{(i)}, \infty))$ whereas $y_2^{(l2)}$ is either $y_2^{(r)}$ if no points are dominated by $\mathbf{y}^{(i)}$ in the y_1y_2-plane or $y_2^{(d[1])}$, otherwise. Moreover, $B^{(i)} = ((y_1^{(i)}, y_2^{(r)}, y_3^{(s)}), (y_1^{(u)}, y_2^{(s)}, \infty))$, and $\mathbf{y}^{(u)}$ denotes either the right neighbour among the newly dominated points in the y_1y_2-plane, or $\mathbf{y}^{(s)}$ if $\mathbf{y}^{(i)}$ is the rightmost point among all newly dominated points. This way each slice can be attributed to exactly one point in $\mathbf{P}$, except for one slice that is created in the final iteration. In the final iteration one additional point $\mathbf{y}^{(n+1)} = (\infty, \infty, \infty)$ is added in the y_1y_2-plane. This point leads to the creation of a slice when it is added, but because it is never discarded it adds only a single slice. In total, therefore $2n+1$ slices are created.

4.2 Computing Slice-Based Parts of the Integral

Given a partitioning of the non-dominated space into integration slices $B_1, \ldots, B_i, \ldots, B_{2n+1}$ the part of the integral related to each of the integration slices can be computed separately. To see how this can be done, the *Hypervolume Improvement* of a point $\mathbf{y} \in \mathbb{R}^3$ is rewritten as:

$$\mathrm{HVI}_3(\mathbf{y}, \mathbf{P}, \mathbf{r}) = \sum_{i=1}^{2n+1} \lambda_3[B_i \cap \Delta(\mathbf{y})] \tag{5}$$

where $\Delta\mathbf{y}$ is the part of the objective space that is dominated by $\mathbf{y}$. The HVI expression in the definition of EHVI in Eq. (3) can be replaced by HVI_3 in Eq. (5):

$$\mathrm{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{P}, \mathbf{r}) = \sum_{i=1}^{2n+1} \int_{y_1=l_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[B_i \cap \Delta(\mathbf{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{6}$$

In Eq. (6), the summation can be done after the integration, because integration is a linear mapping. Then we can divide the integration interval $\int_{y_1=l_1^{(i)}}^{\infty}$ and $\int_{y_2=l_2^{(i)}}^{\infty}$ into $(\int_{y_1=l_1^{(i)}}^{u_1^{(i)}} + \int_{y_1=u_1^{(i)}}^{\infty})$ and $(\int_{y_2=l_2^{(i)}}^{u_2^{(i)}} + \int_{y_2=u_2^{(i)}}^{\infty})$, respectively. Based on this division, Eq. (6) can be expressed by:

$$\text{Eq.6} = \sum_{i=1}^{2n+1} \int_{y_1=l_1^{(i)}}^{u_1^{(i)}} \int_{y_2=l_2^{(i)}}^{u_2^{(i)}} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[B_i \cap \Delta(\mathbf{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{7}$$

$$+ \sum_{i=1}^{2n+1} \int_{y_1=l_1^{(i)}}^{u_1^{(i)}} \int_{y_2=u_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[B_i \cap \Delta(\mathbf{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{8}$$

$$+ \sum_{i=1}^{2n+1} \int_{y_1=u_1^{(i)}}^{\infty} \int_{y_2=l_2^{(i)}}^{u_2^{(i)}} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[B_i \cap \Delta(\mathbf{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{9}$$

$$+ \sum_{i=1}^{2n+1} \int_{y_1=u_1^{(i)}}^{\infty} \int_{y_2=u_2^{(i)}}^{\infty} \int_{y_3=l_3^{(i)}}^{\infty} \lambda_3[B_i \cap \Delta(\mathbf{y})] \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{y}) d\mathbf{y} \tag{10}$$

The idea of this decomposition is that the improvement for $\int_{y_1=u_1^{(i)}}^{\infty}(\cdots)$ and $\int_{y_2=u_2^{(i)}}^{\infty}(\cdots)$ are constant, that is: $\int_{y_k=u_k^{(i)}}^{\infty} \lambda_1[B_i \cap \Delta(y_k)] \cdot \xi_{\mu_k,\sigma_k}(y_k)dy_k = \int_{y_k=u_k^{(i)}}^{\infty}(u_k^{(i)} - l_k^{(i)}) \cdot \xi_{\mu_k,\sigma_k}(y_k)dy_k = (u_k^{(i)} - l_k^{(i)}) \cdot (1 - \Phi(\frac{u_k^{(i)}-\mu_k}{\sigma_k})) =: \vartheta(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k)$, where $k = 1, 2$ and $\lambda_1[B_i \cap \Delta(y_k)]$ is the *Hypervolume Improvement* in dimension k, i.e., a 1-D *Hypervolume Improvement*. We introduce the following abbreviations: $\lambda_1[B_i \cap \Delta(y_k)] = |[l_k^{(i)}, u_k^{(i)}] \cap [l_k^{(i)}, y_k]| = \min\{u_k^{(i)}, y_k\} - l_k^{(i)} =: \ell_{l_k^{(i)}}^{(u_k^{(i)}, y_k)}, \xi_i = \xi_{\mu_i,\sigma_i}(y_i)$, where $i = 1, 2, 3$. Based on this abbreviation, Eq. (7) can be written as

$$\begin{aligned}
\text{Eq.7} &= \sum_{i=1}^{2n+1} \int_{l_1^{(i)}}^{u_1^{(i)}} \ell_{l_1^{(i)}}^{(u_1^{(i)}, y_1)} \xi_1 dy_1 \int_{l_2^{(i)}}^{u_2^{(i)}} \ell_{l_2^{(i)}}^{(u_2^{(i)}, y_2)} \xi_2 dy_2 \int_{l_3^{(i)}}^{\infty} \ell_{l_3^{(i)}}^{(u_3^{(i)}, y_3)} \xi_3 dy_3 \\
&= \sum_{i=1}^{2n+1} \left(\int_{l_1^{(i)}}^{\infty} \ell_{l_1^{(i)}}^{(u_1^{(i)}, y_1)} \xi_1 dy_1 - \int_{u_1^{(i)}}^{\infty} \ell_{l_1^{(i)}}^{(u_1^{(i)}, y_1)} \xi_1 dy_1 \right) \cdot \\
&\quad \left(\int_{l_2^{(i)}}^{\infty} \ell_{l_2^{(i)}}^{(u_2^{(i)}, y_2)} \xi_2 dy_2 - \int_{u_2^{(i)}}^{\infty} \ell_{l_2^{(i)}}^{(u_2^{(i)}, y_2)} \xi_2 y_2 \right) \cdot \int_{l_3^{(i)}}^{\infty} \ell_{l_3^{(i)}}^{(u_3^{(i)}, y_3)} \xi_3 dy_3 \\
&= \sum_{i=1}^{2n+1} \left(\int_{l_1^{(i)}}^{\infty} (y_1 - l_1^{(i)}) \xi_1 dy_1 - \int_{u_1^{(i)}}^{\infty} (u_1^{(i)} - l_1^{(i)}) \xi_1 dy_1 \right) \cdot \\
&\quad \left(\int_{l_2^{(i)}}^{\infty} (y_2 - l_2^{(i)}) \xi_2 dy_2 - \int_{u_2^{(i)}}^{\infty} (u_1^{(i)} - l_2^{(i)}) \xi_2 dy_2 \right) \cdot \int_{l_3^{(i)}}^{\infty} (y_3 - l_3^{(i)}) \xi_3 dy_3 \\
&= \sum_{i=1}^{2n+1} \left(\Psi_\infty(l_1^{(i)}, l_1^{(i)}, \mu_1, \sigma_1) - \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1) \right) \cdot \\
&\quad \left(\Psi_\infty(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) - \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2) \right) \cdot \Psi_\infty(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3)
\end{aligned} \tag{11}$$

The Ψ_∞ functions that are used here, are integrals of the 1-D cumulative Gaussian distribution function (Φ) and the 1-D Gaussian probability density function ξ, these two functions are discussed in the appendix. Similar to the derivation of Eqs. (7), (8), (9) and (10) can be written as follows:

$$\text{Eq.8} = \sum_{i=1}^{2n+1} \Big(\Psi_\infty(l_1^{(i)}, l_1^{(i)}, \mu_1, \sigma_1) - \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1)\Big) \cdot \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2) \cdot \Psi_\infty(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \tag{12}$$

$$\text{Eq.9} = \sum_{i=1}^{2n+1} \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1) \cdot \Big(\Psi_\infty(l_2^{(i)}, l_2^{(i)}, \mu_2, \sigma_2) - \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2)\Big) \cdot \Psi_\infty(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \tag{13}$$

$$\text{Eq.10} = \sum_{i=1}^{2n+1} \vartheta(l_1^{(i)}, u_1^{(i)}, \sigma_1, \mu_1) \cdot \vartheta(l_2^{(i)}, u_2^{(i)}, \sigma_2, \mu_2) \cdot \Psi_\infty(l_3^{(i)}, l_3^{(i)}, \mu_3, \sigma_3) \tag{14}$$

The final EHVI formula is the sum of Eqs. (11), (12), (13) and (14). The Pseudo code of the proposed algorithm is shown in Algorithm 1.

During the EHVI calculation, as the y_1y_2-projections are mutually non-dominated, the points are also sorted by the y_2 coordinate in the AVL tree, identifying a neighbouring point or a discard point takes time $O(\log n)$. Then the EHVI for these integration slices will be calculated by the *calculation_3d* function in Algorithm 1 (line 13 and 23), which is the summation of Eqs. (11), (12), (13) and (14) with the parameters of $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and B_{2n+1}. The EHVI computational complexity for each slice is $O(1)$. Moreover, the dominated points ($\mathbf{y}^{(\mathrm{d[s]})}$) will be removed from the AVL tree, and the new points ($\mathbf{y}^{(j)}$) will be inserted in the AVL tree. Since the points that are dominated by the new point $\mathbf{y}^{(j)}$ will be deleted at the end of the current loop, they will not occur again in the later computations. Hence the total number of open slices does not exceed $2n + 1$, as mentioned before, and the total computation costs $O(n \log n)$.

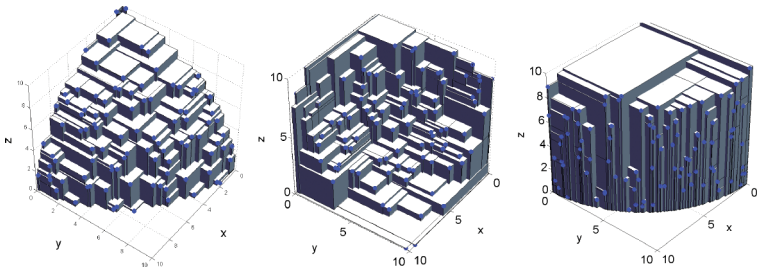

Fig. 4. Randomly generated fronts of type convexSpherical, concaveSpherical, and cliff3D from [28] with $|P| = 100$ (left, middle and right).

Algorithm 1. EHVI calculation algorithm in 3-D

Input: $(\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(n)})$: mutually non-dominated $\mathbb{R}^3$-points sorted by third coordinate (y_3) in descending order
Output: EHVI value

$\mathbf{y}^{(n+1)} = (\infty, \infty, r_3)$;
Initialize AVL tree T for 3-D points
Insert $\mathbf{y}^{(1)}$, $(\infty, r_2, \infty)^T$ and $(r_1, \infty, \infty)^T$ into T;
Initialize the number of integration slices $n_b = 1$;
Initialize $EHVI = 0$;
for $i = 2$ **to** $n + 1$ **do** /* Main loop */
  Retrieve the following information from tree T:
    r: index of the successor of $\mathbf{y}^{(i)}$ in x-coordinate (right neighbour);
    t: index of the successor of $\mathbf{y}^{(i)}$ in y-coordinate (left neighbour);
    d[1], $\cdots$, d[s]: indices of points dominated by $\mathbf{y}^{(i)}$ in y_1y_2-plane, sorted ascendingly in the first coordinate(y_1);
  $B_{n_b}.l_3 = y_3^{(i)}, \quad B_{n_b}.u_2 = y_2^{(i)}, \quad B_{n_b}.u_3 = \infty$;
  if $s == 0$ **then** /* Case 1 */
    $B_{n_b}.l_1 = y_1^{(t)}, \quad B_{n_b}.l_2 = y_2^{(r)}, \quad B_{n_b}.u_1 = y_1^{(i)}$;
    $EHVI = EHVI+$ calculation_3d$(\boldsymbol{\mu}, \boldsymbol{\sigma}, B_{n_b})$;
    $n_b = n_b + 1$;
  else /* Case 2 */
    for $j = 1$ **to** $s + 1$ **do**
      if $j == 1$ **then**
        $B_{n_b}.l_1 = y_1^{(t)}, \quad B_{n_b}.l_2 = y_2^{(\mathrm{d[1]})}, \quad B_{n_b}.u_1 = y_1^{(d[1])}$;
      else if $j == s + 1$ **then**
        $B_{n_b}.l_1 = y_1^{(\mathrm{d[s]})}, \quad B_{n_b}.l_2 = y_2^{(r)}, \quad B_{n_b}.u_1 = y_1^{(i)}$;
      else
        $B_{n_b}.l_1 = y_1^{(\mathrm{d[j-1]})}, \quad B_{n_b}.l_2 = y_2^{(\mathrm{d[j]})}, \quad B_{n_b}.u_1 = y_1^{(\mathrm{d[j]})}$;
      $EHVI = EHVI+$ calculation_3d$(\boldsymbol{\mu}, \boldsymbol{\sigma}, B_{n_b})$;
      $n_b = n_b + 1$;
  Discard $\mathbf{y}^{(\mathrm{d[1]})}, \cdots, \mathbf{y}^{(\mathrm{d[s]})}$ from tree T;
  Insert $\mathbf{y}^{(i)}$ in tree T.

The C++ source-code and MATLAB .mex file for computing the EHVI is available on http://moda.liacs.nl or on request from the authors.

5 Empirical Comparison

Three algorithms, IRS_fast [14], CDD13 [8] and KMAC, which is short for the authors given name, in this paper are compared via the same benchmarks. The test benchmarks from Emmerich and Fonseca [28] are used to generated the Pareto fronts. The Pareto fronts and evaluated points are randomly generated based on convexSpherical, concaveSpherical, and cliff3D functions.

Table 1. Empirical comparisons of strategies for 3-D EHVI calculation

Type	$\|P\|$	Batch Size	Time Average (s)		
			CDD13 [8]	IRS_fast [14]	KMAC
CONVEX	10	1	0.13785	0.00037	**0.00005**
CONVEX	10	10	0.14090	0.00056	**0.00021**
CONVEX	10	100	0.16500	0.00304	**0.00095**
CONVEX	10	1000	0.69104	0.02778	**0.00754**
CONVEX	100	1	13.97556	0.05337	**0.00038**
CONVEX	100	10	17.05551	0.13730	**0.00099**
CONVEX	100	100	45.90095	0.93196	**0.00831**
CONVEX	100	1000	422.31263	8.38585	**0.06462**
CONVEX	1000	1	>3 h	94.72402	**0.00390**
CONVEX	1000	10	>3 h	155.77306	**0.01067**
CONVEX	1000	100	>3 h	795.11319	**0.06517**
CONVEX	1000	1000	>3 h	2838.31854	**0.53801**
CONCAVE	10	1	0.11209	0.00026	**0.00007**
CONCAVE	10	10	0.12790	0.00054	**0.00014**
CONCAVE	10	100	0.14002	0.00294	**0.00077**
CONCAVE	10	1000	0.36697	0.02597	**0.00840**
CONCAVE	100	1	10.62329	0.04895	**0.00031**
CONCAVE	100	10	12.63582	0.12927	**0.00146**
CONCAVE	100	100	27.51827	0.85124	**0.00768**
CONCAVE	100	1000	314.32314	7.67280	**0.06285**
CONCAVE	1000	1	>3 h	91.51055	**0.00332**
CONCAVE	1000	10	>3 h	149.58491	**0.01079**
CONCAVE	1000	100	>3 h	744.46691	**0.06696**
CONCAVE	1000	1000	>3 h	2499.29737	**0.50981**
CLIFF3D	10	1	0.12514	0.00026	**0.00007**
CLIFF3D	10	10	0.13222	0.00055	**0.00013**
CLIFF3D	10	100	0.14432	0.00278	**0.00075**
CLIFF3D	10	1000	0.44964	0.02725	**0.00761**
CLIFF3D	100	1	10.90605	0.04730	**0.00029**
CLIFF3D	100	10	12.85031	0.12709	**0.00112**
CLIFF3D	100	100	44.79395	0.80735	**0.00689**
CLIFF3D	100	1000	679.51368	7.46205	**0.06099**
CLIFF3D	1000	1	>3 h	136.37944	**0.00344**
CLIFF3D	1000	10	>3 h	165.34537	**0.01007**
CLIFF3D	1000	100	>3 h	731.03794	**0.06480**
CLIFF3D	1000	1000	>3 h	2543.16864	**0.51032**

The parameters: $\boldsymbol{\sigma} = (2.5, 2.5, 2.5)$, $\boldsymbol{\mu} = (10, 10, 10)$ were used in the experiments. Pareto front sizes $|P| \in \{10, 100, 1000\}$ and the number of predictions (candidate points) or Batch Size $\in \{1, 10, 100, 1000\}$ are used together with $\boldsymbol{\sigma}$ and $\boldsymbol{\mu}$. Ten trials were randomly generated by the same parameters, and average runtimes (10 runs) for the whole 10 trails with the same parameters were computed. The data for 3-D case with $|P| = 100$ are visualized in Fig. 4, and these figures are originally from [28]. All the experiments were run on the same hardware: Intel(R) Xeon(R) CPU E5-2667 v2 3.30 GHz, RAM 48 GB. The operating system was Ubuntu 12.04 LTS (64 bit), and the compiler was gcc 4.9.2 with compiler flag -Ofast, except for SUMO code, MATLAB 8.4.0.150421 (R2014b), 64 bit. The experiments were set to halt if the algorithms would not finish the EHVI computation within 3 hours. The results are shown in Table 1. While the speed-up gained by batch processing in IRS_fast is considerable, in KMAC the idea is not the case and we just use repeated computation. The results show that the proposed algorithm, KMAC, is the fastest one for all the test problems. Empirical comparisons on randomly generated Pareto fronts of different shape show that the new algorithm is by a factor of 7 to 3.9×10^4 faster than previously published implementations.

6 Related Problems

It is now straightforward to compute other integrals in a similar manner. For instance the *Probability of Improvement* [8,21], that is the probability that a point is non-dominated w.r.t. $\mathbf{P}$, can be computed simply by integration in parts over the $2n + 1$ integration slices, which yields an algorithm with running time in $O(n \log n)$ algorithm for $d = 3$:

$$\mathrm{PoI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{P}) = \sum_{i=1}^{2n+1} \prod_{j=1}^{d} \Phi\left(\frac{u_j^{(i)} - \mu_j}{\sigma_j}\right) - \Phi\left(\frac{l_j^{(i)} - \mu_j}{\sigma_j}\right) \tag{15}$$

Moreover, we can compute the *Truncated Expected Hypervolume Improvement* (TEHVI) [2], by replacing the formulas of Ψ_∞, Φ, and ξ by corresponding formulas of the truncated Gaussian distribution (see [2] for details) and discard irrelevant integration slices that do not intersect with the reference slice [2].

7 Conclusions and Future Research

In this paper, an asymptotically optimal algorithm with a computational complexity of $O(n \log n)$, for the 3-D EHVI exact calculation, was proposed. Compared to [14], the computational complexity is improved by the factor $n^2 / \log n$. This meets the lower bound for the time complexity of the 3-D EHVI computation for $d = 3$, shown by reduction to the *Hypervolume Indicator* problem, see [14]. Thus the algorithm is asymptotically optimal and the time complexity of 3-D EHVI computation is in $\Theta(n \log n)$. As opposed to previous techniques, which

required grid decomposition of the non-dominated subspace into $O(n^3)$ integration slices, the new integration technique can make use of efficient partitioning of the dominated space into only $2n+1$ axis-aligned integration slices. In practice, the new computation scheme will be of great advantage for making the EHVI and related integrals applicable in multiobjective optimization with three objectives, especially in Bayesian Optimization and surrogate-assisted multi-criterion evolutionary algorithms. Empirical comparisons are in line with this theoretical improvement and show the proposed algorithm is by a factor of 7 to 3.9×10^4 faster than previous existing implementations.

It is expected that this technique can be generalized to higher dimensional objective spaces, using for instance the partitioning technique for the non-dominated space discussed in [29]. The time complexity per slice is expected to be $O(2^{d-1})$.

Acknowledgements. Kaifeng Yang acknowledges financial support from the China Scholarship Council (CSC), CSC No. 201306370037. Carlos M. Fonseca was supported by national funds through the Portuguese Foundation for Science and Technology (FCT), and by the European Regional Development Fund (FEDER) through COMPETE 2020 – Operational Program for Competitiveness and Internationalization (POCI).

Appendix

Definition 4 (Ψ_∞ function (see also [14])). *Let $\phi(s) = 1/\sqrt{2\pi}e^{-\frac{1}{2}s^2}(s \in \mathbb{R})$ denote the probability density function (PDF) of the standard normal distribution. Moreover, let $\Phi(s) = \frac{1}{2}\left(1 + erf\left(\frac{s}{\sqrt{2}}\right)\right)$ denote its cumulative probability distribution function (CDF), and erf is Gaussian error function. The general normal distribution with mean μ and standard deviation σ has as PDF, $\xi_{\mu,\sigma}(s) = \phi_{\mu,\sigma}(s) = \frac{1}{\sigma}\phi(\frac{s-\mu}{\sigma})$ and its CDF is $\Phi_{\mu,\sigma}(s) = \Phi(\frac{s-\mu}{\sigma})$. Then the function $\Psi_\infty(a, b, \mu, \sigma)$ is defined as:*

$$\begin{aligned}\Psi_\infty(a, b, \mu, \sigma) &= \int_b^\infty (z-a)\frac{1}{\sigma}\phi\left(\frac{z-\mu}{\sigma}\right)dz \\ &= \sigma\phi\left(\frac{b-\mu}{\sigma}\right) + (\mu - a)\left[1 - \Phi\left(\frac{b-\mu}{\sigma}\right)\right]\end{aligned}$$

References

1. Zaefferer, M., Bartz-Beielstein, T., Naujoks, B., Wagner, T., Emmerich, M.: A case study on multi-criteria optimization of an event detection software under limited budgets. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 756–770. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37140-0_56

2. Yang, K., Deutz, A., Yang, Z., Bäck, T., Emmerich, M.: Truncated expected hypervolume improvement: exact computation and application. In: IEEE Congress on Evolutionary Computation (CEC). IEEE (2016)
3. Yang K, Gaida D, Bäck T, Emmerich M.: 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), pp. 1934–1942, May 2015
4. Michael, T.M., Giannakoglou, K.C., Naujoks, B.: Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. IEEE Trans. Evol. Comput. **10**(4), 421–439 (2006)
5. Koch, P., Wagner, T., Emmerich, M.T., Bäck, T., Konen, W.: Efficient multicriteria optimization on noisy machine learning problems. Appl. Soft Comput. **29**, 357–370 (2015)
6. Shimoyama, K., Jeong, S., Obayashi, S.: Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems. In: IEEE Congress on Evolutionary Computation (CEC), pp. 658–665. IEEE (2013)
7. Shimoyama, K., Sato, K., Jeong, S., Obayashi, S.: Comparison of the criteria for updating kriging response surface models in multi-objective optimization. In: IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2012)
8. Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. J. Global Optim. **60**(3), 575–594 (2014)
9. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**(4), 455–492 (1998)
10. Mockus, J., Tiešis, V., Žilinskas, A.: The application of Bayesian methods for seeking the extremum. In: Towards Global Optimization, vol. 2, pp. 117–131. North-Holland, Amsterdam (1978)
11. Wagner, T., Emmerich, M., Deutz, A., Ponweiser, W.: On expected-improvement criteria for model-based multi-objective optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6238, pp. 718–727. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15844-5_72
12. Emmerich, M.T., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: monotonicity properties and exact computation. In: IEEE Congress on Evolutionary Computation (CEC), pp. 2147–2154. IEEE (2011)
13. Emmerich, M., Yang, K., Deutz, A., Wang, H., Fonseca, C.M.: A multicriteria generalization of bayesian global optimization. In: Pardalos, P.M., Zhigljavsky, A., Žilinskas, J. (eds.) Advances in Stochastic and Deterministic Global Optimization. SOIA, vol. 107, pp. 229–242. Springer, Cham (2016). doi:10.1007/978-3-319-29975-4_12
14. Hupkens, I., Deutz, A., Yang, K., Emmerich, M.: Faster exact algorithms for computing expected hypervolume improvement. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) EMO 2015. LNCS, vol. 9019, pp. 65–79. Springer, Cham (2015). doi:10.1007/978-3-319-15892-1_5
15. Yang, K., Li, L., Deutz, A., Bäck, T., Emmerich, M.: Preference-based multiobjective optimization using truncated expected hypervolume improvement. In: 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. IEEE (2016)
16. Vazquez, E., Bect, J.: Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. J. Stat. Plan. Infer. **140**(11), 3088–3095 (2010)

17. Knowles, J., Hughes, E.J.: Multiobjective optimization on a budget of 250 evaluations. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 176–190. Springer, Heidelberg (2005). doi:10.1007/978-3-540-31880-4_13
18. Keane, A.J.: Statistical improvement criteria for use in multiobjective design optimization. AIAA J. **44**(4), 879–891 (2006)
19. Shimoyama, K., Sato, K., Jeong, S., Obayashi, S.: Updating kriging surrogate models based on the hypervolume indicator in multi-objective optimization. J. Mech. Des. **135**(9), 094503–094503-7 (2013)
20. Svenson, J., Santner, T.: Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. Comput. Stat. Data Anal. **94**, 250–264 (2016)
21. Emmerich, M.T.M.: Single-and multi-objective evolutionary design optimization assisted by Gaussian random field metamodels. Ph.D. thesis, FB Informatik, University of Dortmund, ELDORADO, Dortmund, 10 (2005)
22. Shir, O.M., Emmerich, M., Bäckck, T., Vrakking, M.J.: The application of evolutionary multi-criteria optimization to dynamic molecular alignment. In: IEEE Congress on Evolutionary Computation, pp. 4108–4115. IEEE (2007)
23. Łaniewski-Wołłk, P, Obayashi S, Jeong S.: Development of expected improvement for multi-objective problems. In: Proceedings of 42nd Fluid Dynamics Conference/Aerospace Numerical, Simulation Symposium (CD ROM), Varna, Bulgaria (2010)
24. Luo, C., Shimoyama, K., Obayashi, S.: Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1187–1194. IEEE (2014)
25. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
26. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.D.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)
27. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Hypervolume-based multiobjective optimization: theoretical foundations and practical implications. Theor. Comput. Sci. **425**, 75–103 (2012)
28. Emmerich, M.T.M., Fonseca, C.M.: Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results. In: Takahashi, R.H.C., Deb, K., Wanner, E.F., Greco, S. (eds.) EMO 2011. LNCS, vol. 6576, pp. 121–135. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19893-9_9
29. Lacour, R., Klamroth, K., Fonseca, C.M.: A box decomposition algorithm to compute the hypervolume indicator. Comput. Oper. Res. **79**, 347–360 (2016)