Fast Exact Computation of Expected HyperVolume Improvement

Guang Zhao[1]   **Raymundo Arróyave**[2]   **Xiaoning Qian**[1]

[1]Department of Electrical and Computer Engineering, Texas A&M University
[2]Department of Materials Science and Engineering, Texas A&M University

Abstract

In multi-objective Bayesian optimization and surrogate-based evolutionary algorithms, Expected HyperVolume Improvement (EHVI) is widely used as the acquisition function to guide the search approaching the Pareto front. This paper focuses on the exact calculation of EHVI given a nondominated set, for which the existing exact algorithms are complex and can be inefficient for problems with more than three objectives. Integrating with different decomposition algorithms, we propose a new method for calculating the integral in each decomposed high-dimensional box in constant time. We develop three new exact EHVI calculation algorithms based on three region decomposition methods. The first grid-based algorithm has a complexity of $O(m \cdot n^m)$ with n denoting the size of the nondominated set and m the number of objectives. The Walking Fish Group (WFG)-based algorithm has a worst-case complexity of $O(m \cdot 2^n)$ but has a better average performance. These two can be applied for problems with any m. The third CLM-based (Beume et al., 2009) algorithm is only for $m = 3$ and asymptotically optimal with complexity $\Theta(n \log n)$. Performance comparison results show that all our three algorithms are at least twice faster than the state-of-the-art algorithms with the same decomposition methods. When $m > 3$, our WFG-based algorithm can be over 10^2 faster than the corresponding existing algorithms. Our algorithm is demonstrated in an example involving efficient multi-objective material design with Bayesian optimization.

1 Introduction

With rapid advancements in sensing and computing, artificial intelligence (AI), especially machine learning (ML), has become an important tool in a wide variety of applications. With available "big" data, AI/ML can help derive accurate prediction for better decision making to maximize the gain with reduced time and cost (McAfee et al., 2012; Provost and Fawcett, 2013). However, in applications involving complex systems, for example, in multi-functional materials discovery (Cadman et al., 2013), it may not be possible to always have big data for effective ML. Often, model and data uncertainty, with limited training samples, have to be incorporated in a "smart" way for optimization to actively learn about the system and achieve the desirable goals. In such "data-poor" environments, Bayesian experimental design (Chaloner and Verdinelli, 1995) and reinforcement learning (Wiering and Van Otterlo, 2012) are adopted.

When simulations and experiments of a complex system under study are resource (cost and time) demanding, surrogate-based methods are adopted considering uncertainty for Bayesian experimental design. For example, Gaussian process models have been studied extensively as the surrogate models for the objective functions for decision making (Frazier et al., 2009). Based on that, the expected utility can be computed based on the application-specific acquisition function for Bayesian optimization to guide the search for decision making towards promising or less explored regions of the search space. We focus on Bayesian experimental design with multiple objective functions, for which Expected HyperVolume Improvement (EHVI) (Emmerich et al., 2006) has proved to be an effective acquisition function (Couckuyt et al., 2014). Moreover, EHVI is also used for preselection criteria for multi-objective evolutionary algorithms (Luo et al., 2015).

Computing EHVI can be computationally demanding, especially when the number of objective functions and the complexity of the Pareto front increases. This is not desirable in many Bayesian experimental design

problems. For example, for novel materials discovery, one of the major goals of applying Bayesian optimization is to reduce the time to run high-throughput simulations to identify candidate desirable materials. We need to guarantee that the computational time and complexity of Bayesian optimization is significantly better than those of high-throughput simulations to achieve our goals. Hence, efficiently computing EHVI for multi-objective Bayesian optimization is critical.

Usually EHVI is a multi-modal function in the search space. We need to search for the point with the maximum EHVI in each iteration of Bayesian optimization. As a result, EHVI needs to be calculated multiple times. Exact calculation for EHVI has been studied in the literature (Couckuyt et al., 2014; Hupkens et al., 2015; Emmerich et al., 2016; Yang et al., 2017). Existing exact EHVI calculation algorithms have two steps: at the first step, the integral region is decomposed into high-dimensional boxes; and then at the second step the integral in each box is computed and added together to compute EHVI. The integral region decomposition methods have been well studied using the algorithms calculating hypervolume indicator (Lacour et al., 2017). Existing EHVI calculation algorithms mostly tailor the integral computation in boxes to different decomposition methods and it is difficult to transfer across different decomposition methods (Hupkens et al., 2015; Yang et al., 2017). For two- and three-objective cases, Emmerich et al. (2016); Yang et al. (2017) have shown that EHVI can be calculated in asymptotically optimal time of $\Theta(n \log n)$ with n denoting the size of the nondominated set. However, these algorithms cannot be extended for more than three objectives. The existing algorithms are complicated and time-consuming when n is large with the number of objectives $m > 3$ (Couckuyt et al., 2014).

In this paper, we develop algorithms for the fast, exact computation of EHVI with different integral region decomposition methods. For any decomposition method given a nondominated set, we first transform the integral expression of EHVI and show that the transformed integrand has a closed form in each decomposed box and can be computed in constant time. We develop three different exact EHVI algorithms following this new scheme and name them after the decomposition methods. The first grid-based algorithm has a complexity of $O(m \cdot n^m)$, with n denoting the size of the nondominated set and m the number of objectives. The second WFG-based algorithm has a worst case complexity of $O(m \cdot 2^n)$ but experiments show it has a better average performance than grid-based algorithm. These two algorithms can be applied for problems with any m. A CLM-based algorithm is proposed merely for three objective problems with an

asymptotically optimal complexity of $\Theta(n \log n)$.

The paper is structured as follows: Section 2 introduces the definition of EHVI and state-of-the-art exact calculation algorithms. Section 3 derives another EHVI integral expression and the closed form of the integral in each box. Section 4 details three decomposition methods. Finally, Section 5 presents the performance comparison of the proposed algorithms with state-of-the-art exact algorithms and the proposed algorithm application on Bayesian optimization for multi-objective material design.

2 Background

2.1 Multi-objective Bayesian Optimization

Formulate m-objective minimization problem as:

$$\min(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})), \boldsymbol{x} \in \mathcal{X}, \qquad (1)$$

where $\mathcal{X}$ is the feasible set or search space to explore. Denote the m-objective function image as $\mathcal{Y} \subset \mathbb{R}^m$. Usually there would not be a single solution optimizing every objective simultaneously. We aim to find the set of optimal trade-off solutions known as the Pareto front, which is defined as $\mathcal{Y}_p = \{\boldsymbol{y} \in \mathcal{Y} : \nexists \boldsymbol{y}' \text{ s.t. } \boldsymbol{y}' \prec \boldsymbol{y}\}$. Here, $\boldsymbol{y} \prec \boldsymbol{y}'$ reads as $\boldsymbol{y}$ dominates $\boldsymbol{y}'$ in the context of minimization, meaning that $\forall i \leq m, y_i \leq y_i'$ and $\exists j \leq m, y_j < y_j'$. A vector $\boldsymbol{y}$ is nondominated with respect to a set A if and only if $\boldsymbol{y}$ is not dominated by any vector in A. The nondominated set of A is defined as the subset that includes all nondominated vectors with respect to A. So the Pareto front $\mathcal{Y}_p$ is the non-dominated set of $\mathcal{Y}$. On the other hand, all the vectors not in the $\mathcal{Y}_p$ are dominated by at least one element in the $\mathcal{Y}_p$. So the Pareto front dominates the largest region of vectors in the objective space. In multi-objective optimization, we can use the dominated hypervolume as a metric guiding the search for the Pareto front as detailed below.

When the objective functions are unknown and expensive to evaluate, we wish to obtain an approximation to the Pareto front with the minimum number of required objective function evaluations. Bayesian Optimization (BO) is one solution to this problem. BO is a sequential sampling method that samples new points utilizing the information of previously observed data. In each sampling iteration, a probabilistic model and an acquisition function are calculated. The probabilistic model describes the distribution of objective functions. It starts with a prior belief of objective functions and updates the belief based on the observed data, following Bayes' rule. Usually the surrogate models are chosen as independent Gaussian process models for their good interpolation performance and easy computation

property (Emmerich and Klinkenberg, 2008; Picheny, 2015).

Given a Gaussian process prior, the posterior of the objective functions of a candidate point is still Gaussian. Hence, the computation of the acquisition function can be efficient. The acquisition function describes the utility of querying each candidate point given the probabilistic model, usually considering both the utility of exploration and exploitation. The maximum point of the acquisition function is chosen to evaluate in each iteration. Compared with the original objective functions, the acquisition function is much cheaper to evaluate and maximize. In multi-objective problems, a popular choice of the acquisition function is EHVI for its good benchmark performance (Couckuyt et al., 2014), which describes the expected gain in the dominated hypervolume. Its formal definition is given in the following subsection.

2.2 Expected HyperVolume Improvement

Assume $A \subset \mathcal{Y}$ is an objective vector set, the dominated hypervolume of A is defined as the hypervolume of the dominated region:

$$\mathcal{H}(A) = \text{Vol}\left(\{\boldsymbol{y} \in \mathbb{R}^m | \boldsymbol{y} \prec \boldsymbol{r} \text{ and } \exists\, a \in A : a \prec \boldsymbol{y}\}\right), \tag{2}$$

where $\boldsymbol{r}$ is a vector dominated by all the vectors in image $\mathcal{Y}$, called the reference point, which is introduced to bound the objective space, such that the dominated hypervolume is finite. We can also define the nondominated region as the whole region bounded by $\boldsymbol{r}$ minus the dominated region. From (2), it's easy to see that the dominated hypervolume of A equals to the dominated hypervolume of its nondominated set and therefore the dominated hypervolume of $\mathcal{Y}_p$ has the maximum dominated hypervolume, which equals to the dominated hypervolume of $\mathcal{Y}$. Now for BO the problem of approaching the Pareto front can be transformed to finding the set with the maximum dominated hypervolume.

With the definition of dominated hypervolume, the hypervolume improvement of a vector $\boldsymbol{y}$ with respect to A is defined as:

$$\mathcal{H}_I(\boldsymbol{y}, A) = \mathcal{H}(A \cup \{\boldsymbol{y}\}) - \mathcal{H}(A). \tag{3}$$

Notice that the hypervolume improvement can only be nonzero when $\boldsymbol{y}$ falls in the nondominated region of A.

In the context of BO, A is the observed set of the objective vectors and $\boldsymbol{y}$ is the random output of the probabilistic model corresponding to an candidate point. As a result, $\mathcal{H}_I(\boldsymbol{y}, A)$ is a random variable and EHVI is

defined as its expected value over distribution $P(\boldsymbol{y})$:

$$E\mathcal{H}_I(\boldsymbol{y}) = \mathbb{E}[\mathcal{H}_I(\boldsymbol{y}, A)] = \int_U \mathcal{H}_I(\boldsymbol{y}', A)P(\boldsymbol{y} = \boldsymbol{y}')\mathrm{d}\boldsymbol{y}', \tag{4}$$

which is an integral of hypervolume improvement over the nondominated region U since in the dominated region $\mathcal{H}_I(\boldsymbol{y}', A) = 0$. EHVI is the expected gain in dominated hypervolume of observing one candidate point. Hence, in a myopic sequential sampling method, we will choose the candidate point with the maximum EHVI as the next point to query.

2.3 Algorithms for Calculating EHVI

As shown in definition, EHVI is the integral of hypervolume improvement with respect to the posterior measure over the nondominated region of a nondominated set. Since the nondominated region can have a complex shape, most existing exact EHVI calculation algorithms adopt a two-step approach, in which the first step is to decompose the integral region into high-dimensional boxes based on the observed set, and the second step is to calculate the integral contribution in each box separately and then add all the contributions together. For the integral in the second step, the integrand, i.e. the hypervolume improvement, also has a complex form that depends on the shape of the nondominated region. The integrand is often partitioned into terms by region decomposition so that the integral of each term has a closed form over each box for efficient computation. The complexity of such two-step approaches mainly depends on m and n as larger m and n result in a more complex shape of nondominated regions.

Couckuyt et al. (2014) derived an algorithm (CDD13) capable of calculating the exact EHVI for any number of objectives. It obtains the nondominated region by subtracting the dominated region from the whole objective space and then recursively decomposes the dominated region into boxes following the Walking Fish Group (WFG) algorithm (While et al., 2012), which calculates the exact hypervolumes. To calculate the integral in each box, CDD13 algorithm again decomposes the integrand, namely the hypervolume improvement into contributing terms belonging to every decomposed box, so that each term has a closed form integral over box. This nested decomposition scheme limits its application when n and m are large and the paper did not provide a complexity bound for the algorithm.

Hupkens et al. (2015) proposed an algorithm (IRS) valid for calculating EHVI for three objectives in $O(n^3)$. IRS generates a grid structure based on the coordinates of the nondominated set, and decomposes

the integral region into the corresponding $O(n^3)$ hyperrectangular grid cells (boxes). Then the algorithm decomposes the integrand into 8 terms in 3d space. Thanks to the grid structure, the integral terms between some of different boxes share same multiplication factors, and the algorithm can calculate all the factors in advance within $O(n^3)$ time and store the calculation results in an array of size $O(n^2)$. However, since the integrand decomposition is complicated, the complexity of the algorithm has a large constant factor and it is hard to extend the algorithm for the problems with more than three objectives.

More recently, Yang et al. (2017) developed a more efficient algorithm (KMAC) calculating EHVI for three objectives with the asymptotically optimal time complexity $\Theta(n \log n)$. The algorithm decomposes the integral region into $2n+1$ boxes, following an asymptotically optimal algorithm for calculating hypervolume improvement (Emmerich and Fonseca, 2011). With the integrand partitioned the same way as in Couckuyt et al. (2014), the algorithm takes advantage of the integral region decomposition method computing all the contributing hypervolumes in constant time. However, this algorithm is only asymptotically optimal for computing a three-objective EHVI. Both the decomposition method and integral trick are difficult to be extended to problems with $m > 3$.

Finally, Feliot et al. (2017) wrote the formula for computing EHVI (4):

$$E\mathcal{H}_I(\boldsymbol{y}) = \int_U P(\boldsymbol{y} \prec \boldsymbol{y}')\mathrm{d}\boldsymbol{y}', \qquad (5)$$

where $P(\boldsymbol{y} \prec \boldsymbol{y}')$ is the probability that the random vector $\boldsymbol{y}$ is dominated by $\boldsymbol{y}'$. The integrand in this equation has a relatively simple form, which motivates our research. We will derive in the next section a closed-form expression. However, Feliot et al. (2017) did not provide an exact algorithm calculating the integral but used a Monte Carlo sampling method instead.

In this paper, motivated by the above methods, we derive a closed-form integral computation for exact computation of (5). With that, different decomposition methods can be integrated to derive more efficient exact EHVI calculating algorithms.

3   Closed-form Integral Computation

In this section, we first show the equivalence of (4) and (5). In (3), the hypervolume improvement is the hypervolume of the region that is dominated by $\boldsymbol{y}$ but not dominated by A. Hence, it can be expressed with

the indicator function $\mathbb{1}(\cdot)$ as :

$$\mathcal{H}_I(\boldsymbol{y}, A) = \int_U \mathbb{1}(\boldsymbol{y} \prec \boldsymbol{y}')\mathrm{d}\boldsymbol{y}'. \qquad (6)$$

Computing EHVI by (4) can be transformed into another form (Feliot et al., 2017):

$$
\begin{aligned}
E\mathcal{H}_I(\boldsymbol{y}) &= \mathbb{E}_{\boldsymbol{y}}\Big[\int_U \mathbb{1}(\boldsymbol{y} \prec \boldsymbol{y}')\mathrm{d}\boldsymbol{y}'\Big] \\
&= \int_U \mathbb{E}_{\boldsymbol{y}}[\mathbb{1}(\boldsymbol{y} \prec \boldsymbol{y}')]\mathrm{d}\boldsymbol{y}' \\
&= \int_U P(\boldsymbol{y} \prec \boldsymbol{y}')\mathrm{d}\boldsymbol{y}', \qquad (7)
\end{aligned}
$$

where in the second line we reverse the order of integrations based on Fubini's theorem, and the third line by the probability identity. This equation transfers the problem of computing EHVI into computing the integral of the cumulative probability $P(\boldsymbol{y} \prec \boldsymbol{y}')$ over the whole objective space, which is simpler as it does not depend on the shape of the nondominated region.

With independent Gaussian process models as surrogate models for each objective in BO, the posterior output of each Gaussian process given the data can be considered as mutually independent Gaussian random variables $y_j \sim \mathcal{N}(\mu_j, \sigma_j^2), j \leq m$, with μ_j, σ_j^2 denoting the mean and covariance of j-th objective, respectively. Denote the probability distribution function of the standard normal distribution as $\phi(\cdot)$ and the corresponding cumulative distribution function as $\Phi(\cdot)$. Given a new candidate point, we have:

$$P(\boldsymbol{y} \prec \boldsymbol{y}') = \prod_{j=1}^m \Phi\Big(\frac{y_j' - \mu_j}{\sigma_j}\Big). \qquad (8)$$

As we mentioned in Section 2, the integral region U can be decomposed into a set of high-dimensional boxes. Here we derive the closed-form integral computation of $P(\boldsymbol{y} \prec \boldsymbol{y}')$ in each box. Let $box(\boldsymbol{l}, \boldsymbol{u})$ indicate the half open region $(l_1, u_1] \times (l_2, u_2] \times \cdots \times (l_m, u_m]$, with $\boldsymbol{l} = (l_1, l_2, \ldots, l_m)$ denoting the lower bound and $\boldsymbol{u} = (u_1, u_2, \ldots, u_m)$ denoting the upper bound. The integral of $P(\boldsymbol{y} \prec \boldsymbol{y}')$ in $box(\boldsymbol{l}, \boldsymbol{u})$ is:

$$
\begin{aligned}
\delta(box(\boldsymbol{u}, \boldsymbol{l})) &= \int_{box(\boldsymbol{u},\boldsymbol{l})} \prod_{j=1}^m \Phi\Big(\frac{y_j' - \mu_j}{\sigma_j}\Big)\mathrm{d}\boldsymbol{y}' \\
&= \prod_{j=1}^m \int_{l_j}^{u_j} \Phi\Big(\frac{y_j' - \mu_j}{\sigma_j}\Big)\mathrm{d}y_j'. \qquad (9)
\end{aligned}
$$

The integral of $\Phi(a+bx)$ has a closed form: $\int_{-\infty}^t \Phi(a+bx)\mathrm{d}x = \frac{1}{b}[(a+bt)\Phi(a+bt)+\phi(a+bt)]$ (Patel and Read, 1996). We then can write (9) as:

$$\delta(box(\boldsymbol{u}, \boldsymbol{l})) = \prod_{j=1}^m [\Psi(u_j, \mu_j, \sigma_j) - \Psi(l_j, \mu_j, \sigma_j)] \quad (10)$$

with $\Psi(a,\mu,\sigma) = \int_{-\infty}^{a} \Phi(\frac{y-\mu}{\sigma})\mathrm{d}y = (a-\mu)\Phi(\frac{a-\mu}{\sigma}) + \sigma\phi(\frac{a-\mu}{\sigma})$.

The integral in one box in (10) is not dependent on the other boxes or the nondominant set A. It can be calculated with the computation complexity in $O(m)$, which is constant with respect to n. What's more, computing this integral does not need the decomposed boxes to have certain structures, contrasting the requirement of the grid structure in Hupkens et al. (2015). Hence, we can decompose the integral region into boxes using any of the previously discussed decomposition methods in Section 2. It is worth mentioning that usually $\int_{box(\boldsymbol{u},\boldsymbol{l})} P(\boldsymbol{y} \prec \boldsymbol{y}')\mathrm{d}\boldsymbol{y}' \neq \int_{box(\boldsymbol{u},\boldsymbol{l})} \mathcal{H}_I(\boldsymbol{y}', A)P(\boldsymbol{y} = \boldsymbol{y}')\mathrm{d}\boldsymbol{y}'$. The two integrals are only equal when the integral region is the whole nondominated region U.

4 Decomposing Integral Regions

With the closed-form integral computation for each hyperrectangle, we now integrate it with different decomposition methods to propose algorithms, including 1) grid-based algorithm, 2) WFG-based algorithm, and 3) CLM-based algorithm to calculate EHVI exactly given the nondominated set.

4.1 Grid-based Decomposition

The grid-based algorithm decomposes the integral region into $O(n^m)$ m-dimensional disjoint boxes based on a grid structure, suggested by Hupkens et al. (2015). Denote the sorted list of jth coordinates of elements in nondominated set A as $a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(n)}$, and assume $a_j^{(0)} = -\infty$, $a_j^{(n+1)} = r_j$. We can then build the grid based on the sorted lists of m objectives and generate boxes (cells) bounded by the grid as shown in Fig. 1.

The nondominated region can be decomposed into the generated boxes with lower bound nondominated or belonging to A. Here is the proof. Given a $box(\boldsymbol{u},\boldsymbol{l})$ and A, if the $\boldsymbol{l}$ is not dominated or equal to any vectors in A, i.e. $\boldsymbol{l} \not\succeq \boldsymbol{a}, \forall \boldsymbol{a} \in A$, then $\exists j$, s.t. $l_j < a_j$ and since l_j, u_j and a_j are from the same discrete sorting list, we have $u_j \leq a_j$. As a result the whole box bounded by $\boldsymbol{l}$ and $\boldsymbol{u}$ is nondominated by A. On the other hand, if $\boldsymbol{l}$ is dominated by one vector $\boldsymbol{a} \in A$, then the whole box is dominated by $\boldsymbol{a}$. Therefore the union of boxes with lower bound nondominated or belonging to A equals to the nondominated set.

The relationship between lower bounds $\boldsymbol{l}$ and A can be identified in constant time, by comparing l_m with $H(l_1, l_2, \ldots, l_{m-1})$, an array indexed by the first $m-1$ coordinates which stores the lowest value of the m-th coordinate given $\boldsymbol{l} \succeq A$. Array H stores values of all possible combinations of the first $m-1$ coor-

Figure 1: Illustration of grid-based decomposition method: U (yellow region) denotes the nondominated region; D denotes the dominated region. The grid coordinates are indicated by the sorted lists $a_1^{(i)}$ and $a_2^{(i)}$ for the first and second objectives.

dinates of lower bounds, and the number of combinations, also the size of array, is n^{m-1}. The array can be built by going through all vectors $\boldsymbol{a} \in A$ in the descending order of the m-th coordinate, and for each $\boldsymbol{a}$, setting or updating $H(l_1, l_2, \ldots, l_{m-1}) = a_m$ for all $(l_1, l_2, \ldots, l_{m-1}) \succeq (a_1, a_2, \ldots, a_{m-1})$. The complexity of building this array is $O(n^m)$. After that the complexity of identifying nondominated boxes and calculating the corresponding integrals is $O(m \cdot n^m)$. Hence the total complexity of the grid-based algorithm is also $O(m \cdot n^m)$.
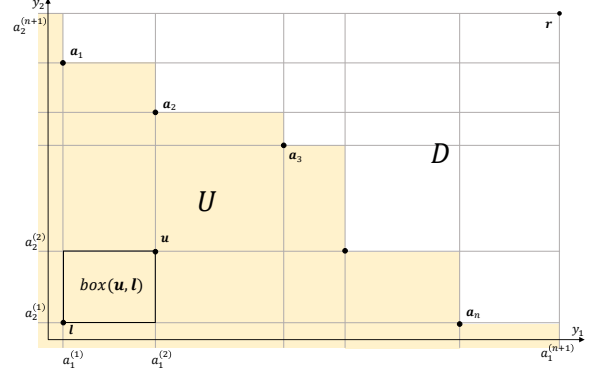
4.2 WFG-based Decomposition

The grid-based algorithm is intuitive and easy to implement, but the number of boxes can be large when m is high. This is undesirable since the algorithm running time depends mostly on the number of boxes. In fact, the grid-based decomposition can be inefficient since some boxes can be combined into a larger box. To decompose the integral region into as few boxes as possible, we can borrow the idea in the exact algorithms for calculating the dominated hypervolume based on decomposition (While et al., 2006, 2012; Russo and Francisco, 2014; Jaszkiewicz, 2018; Lacour et al., 2017). Among them we choose to extend the Walking Fish Group (WFG) algorithm (While et al., 2012) to calculate EHVI since it's one of the fastest algorithms for calculating hypervolume (Jaszkiewicz, 2018).

WFG calculates the dominated hypervolume of a nondominated set A as a summation of hypervolume improvements (exclusive hypervolume):

$$\mathcal{H}(A) = \sum_{i=1}^{n} \mathcal{H}_I(\boldsymbol{a}_i, \{\boldsymbol{a}_{i+1}, \ldots, \boldsymbol{a}_n\}). \qquad (11)$$
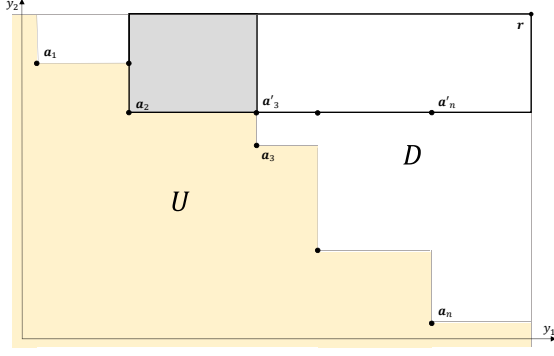
Figure 2: One step in the WFG-based algorithm to calculate $\mathcal{H}_I(\boldsymbol{a}_2, \{\boldsymbol{a}_3, \ldots, \boldsymbol{a}_n\}) = \mathcal{H}(\{\boldsymbol{a}_2\}) - \mathcal{H}(\{\boldsymbol{a}'_3, \ldots, \boldsymbol{a}'_n\})$, denoted in the shaded region.

$\mathcal{H}_I(\cdot, \cdot)$ is defined in (3) and we define $\mathcal{H}_I(\boldsymbol{a}, \emptyset) = \mathcal{H}(\boldsymbol{a})$. In (11), if we calculate the hypervolume improvements by the definition, we still need to calculate $\mathcal{H}(A)$ since $\mathcal{H}_I(\boldsymbol{a}_1, \{\boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}) = \mathcal{H}(A) - \mathcal{H}(\{\boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\})$. Fortunately, the hypervolume improvement for a vector $\boldsymbol{a}$ with respect to S is equal to the hypervolume of $\boldsymbol{a}$ minus the hypervolume of another set S':

$$\mathcal{H}_I(\boldsymbol{a}, S) = \mathcal{H}(\{\boldsymbol{a}\}) - \mathcal{H}(S'), \qquad (12)$$

where $S' = \{\text{limit}(\boldsymbol{s}, \boldsymbol{a}) | \boldsymbol{s} \in S\}$ and $\text{limit}(\boldsymbol{s}, \boldsymbol{a}) = (\max(s_1, a_1), \ldots, \max(s_m, a_m))$. The first term $\mathcal{H}(\{\boldsymbol{a}\})$ is just the hypervolume of $box(\boldsymbol{a}, \boldsymbol{r})$ and the second term $\mathcal{H}(S')$ can be calculated recursively by (11). By this recursive procedure, WFG transforms the dominated hypervolume into summation and difference of a series of boxes. Figure 2 illustrates the calculation of $\mathcal{H}_I(\boldsymbol{a}_2, S = \{\boldsymbol{a}_3, \ldots, \boldsymbol{a}_n\}) = \mathcal{H}(\{\boldsymbol{a}_2\}) - \mathcal{H}(S' = \{\boldsymbol{a}'_3, \ldots, \boldsymbol{a}'_n\})$: the exclusive hypervolume of $\boldsymbol{a}_2$ is illustrated as the grey region. Note that, $\boldsymbol{a}'_3$ dominates all other vectors in S' so that other vectors make no contribution to the hypervolume and can be disregarded. Before each recursion of calculating $\mathcal{H}(S')$, we can take only the nondominated set of S' into the following calculation, which makes WFG more efficient. It's shown in While et al. (2012) that in most cases the nondominated set will lose over 50% vectors after being limited by one of its vectors. It's worth mentioning that in high dimension cases, the exclusive hypervolumes are not always boxes, so the hypervolume improvement has to be calculated by (12).

To apply WFG to calculate EHVI, we can extend WFG to calculate the integral of integrand (8) over the dominated region D and then subtract it from the integral of the whole region bounded by $\boldsymbol{r}$, which can be expressed by $\delta(box(-\infty, \boldsymbol{r}))$ and calculated by (10). The dominated region can be decomposed into a series of boxes by following the same adding and subtracting

rules in WFG for each box, for which we can compute the integral (9) for each box. The complexity of the WFG-based algorithm is proportional to the number of decomposed boxes. In the worst case, the number of boxes can be $2^n - 1$, so the WFG-based algorithm has a worst case complexity $O(m \cdot 2^n)$. However, in most real-world cases the average complexity is much smaller, which will be demonstrated in Section 5.

4.3 CLM-based Decomposition

We now derive another decomposition algorithm specially designed for three-objective EHVI calculation based on the CLM algorithm (Beume et al., 2009). The CLM algorithm is a space sweeping algorithm along the y_3 axis, which achieves asymptotically optimal complexity $\Theta(n \log n)$. It decomposes the 3d volume into slices with the volume of each slice being calculated by multiplying the height along the y_3 axis by the 2d area on the (y_1, y_2) plane. The algorithm complexity is enhanced by taking advantage of a balanced binary search tree to store the nondominated projection on each plane. Such a tree structure can avoid repeated dominance checks and the recalculation of dominated area on the (y_1, y_2) plane.

We can use this algorithm to calculate the integral of integrand (8) in the dominated region D and subtract it from the integral of the whole region to obtain EHVI, just as in our WFG-based algorithm. To calculate the integral, we only need to replace the volume calculation in the space sweeping algorithm with (9). Since (9) also can be calculated in constant time, the CLM-based EHVI calculation will have the same asymptotically optimal algorithm complexity $\Theta(n \log n)$ as in Beume et al. (2009). The existing KMAC algorithm (Yang et al., 2017) uses a similar decomposition method extended from the CLM algorithm, also with the asymptotically optimal complexity. The performance comparison between KMAC and our CLM-based algorithm is found in the next section.

5 Experiments

Experiments are performed to compare different algorithms calculating exact EHVI, including IRS (Hupkens et al., 2015), CDD13 (Couckuyt et al., 2014), KMAC (Yang et al., 2017) and our three algorithms with grid-based, WFG-based, and AVL-based decomposition algorithms.[1] Based on the decomposition method, we can pair these algorithms as IRS and

---

[1] IRS and KMAC are coded in C++ and obtained from http://liacs.leidenuniv.nl/~csmoda/index.php?page=code. We implemented the other algorithms in C/C++.

grid-based algorithm, CDD13 and WFG-based algorithm, KMAC and CLM-based algorithm. In the experiments, all the objectives are maximized, and the nondominated sets for EHVI calculation are based on the benchmark from `http://www.wfg.csse.uwa.edu.au/hypervolume/`. Each nondominated set A is randomly generated, initially A is empty, and sequentially a vector $\boldsymbol{a}$ randomly samples from box $[0.1, 10]^m$, $\boldsymbol{a}$ will be added to A if $A \cup \boldsymbol{a}$ is still a nondominated set. The distribution of candidate points is assumed to be independent Gaussian with mean $\mu_i = 10$ and variance $\sigma_i = 2.5$, for $i \leq m$.

The first experiment compares the performance of the proposed grid-based, WFG-based, and CLM-based algorithms with IRS, CDD13, and KMAC on the benchmark datasets with the nondominated set of size $n \in \{10, 50, 100, 150, 200, 250, 300\}$ when $m = 3$. For each data size n, the algorithms are applied on 10 different random nondominated sets, and perform 5 runs for each nondominated set. All algorithms were validated by producing the same results for the same nondominated sets. The average running time over different data size is shown in Fig. 3. The existing algorithms are marked with circles, and our algorithms with triangles. The algorithms with the same decomposition method are in the same color. From the figure, we can see that all of our proposed algorithms consistently perform better than the corresponding state-of-the-art algorithms with the same decomposition method. Among them, IRS and grid-based algorithms only have a constant difference in running time (ours is at least twice faster) because they have the same $O(n^3)$ complexity but IRS has a larger constant factor since it needs to calculate eight integral contribution terms for integral in each box compared with our proposed algorithm calculate a closed form expression. A similar trend can be observed between KMAC and our CLM-based algorithm, both have the same $\Theta(n \log n)$ complexity. Our WFG-based algorithm has a smaller overall complexity than CDD13 because for CDD13 calculating integral in each box requires summation over all boxes with time complexity higher than constant. Our WFG-based algorithm performs better than the grid-based algorithm, demonstrating that our WFG algorithm decomposes the integral regions into fewer boxes than the grid-based decomposition does.

In the second set of experiments, we show how the computation time varies with m. Since IRS, KMAC and our CLM-based algorithms are only applicable with $m = 3$, we only compare our grid-based and WFG-based algorithms with CCD13 with the benchmark examples when $n = 10$ and $m \in \{3, 4, 5, 6, 7, 8\}$. Again for each case we perform 5 runs of each algorithm on 10 different random nondominated sets
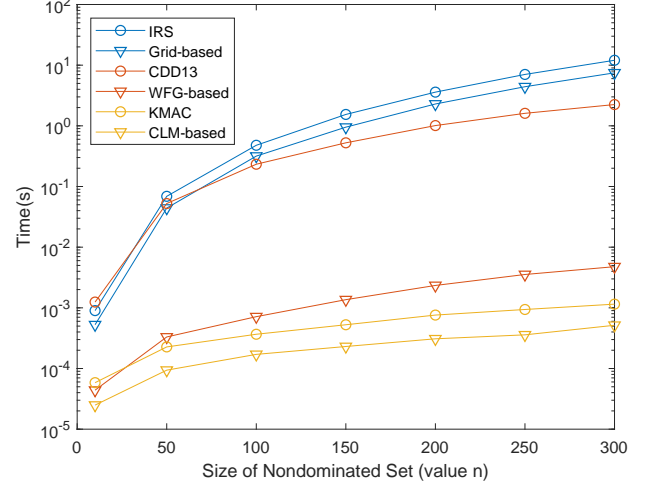

Figure 3: Performance comparison for BO with respect to different n for three-objective benchmark data ($m = 3$). Reported computation time is averaged over five runs of 10 different benchmark nondominated sets.

and all the algorithm give same results for the same nondominated sets. Figure 4 illustrates the increasing trend of the computation time with increasing m. When $m = 8$, the grid-based algorithm takes much longer time (> 500 seconds) and is not included in the plot for better visualization. From the figure, we can see the complexity of the grid-based algorithm is nearly exponential to the number of objectives, confirming the derived complexity $O(m \cdot n^m)$. The running time of our WFG-based algorithm and CDD13 is also nearly exponential to m with a smaller bases ($< n$) when m is small. But when m approaches $n = 10$, the running time starts to saturate because the number of decomposed boxes by WFG algorithm is bounded by the worst case complexity $O(2^n)$.

Last but not least, we have applied our CLM-based algorithm for implementing Bayesian experimental design to search for desirable NiTi Shape Memory Alloys (SMAs) (Cox et al., 2017; Yu et al., 2015). SMAs are capable of undergoing reversible shape chages that make them ideal actuators. However, their properties are extremely sensitive to minute changes in their microstructure and composition and this opens up a very large design space that is extremely costly to explore solely through traditional experimental means.

For our problem, we would like to search for NiTi SMAs in the materials design space $\mathcal{X}$ determined by the Ni concentration and precipitate Volume Fraction (VF). A Finite Element-based micromechanical model (Cox et al., 2017) is used to evaluate the material properties, from which we choose three objectives to optimize: 1) the austenitic finish temperature being close to 30°C, i.e. $\min |A_f - 30|$, 2)
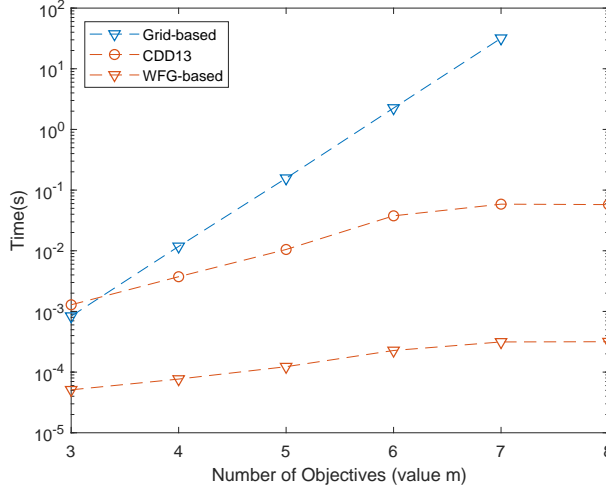
Figure 4: Performance comparison with different m between CCD13 and our grid-based and WFG-based algorithms based on the benchmark datasets of $n = 10$. Reported computation time is averaged over five runs of 10 different benchmark nondominated sets.

the difference of austenitic finish temperature and martensitic start temperature being close to 40°C, i.e. $\min |A_f - M_s - 40|$, and 3) maximizing transformation strain. Here the martensitic temperatures are the ones that correspond to the reversible shape change upon cooling and the austenitic temperatures correspond to the reverse transformation upon heating.

Independent Gaussian Processes with squared exponential kernel functions are adopted to model these objective functions. The hyperparameters of the Gaussian Processes are optimized by the quasi-Newton method. The initial set is generated by random sampling. Subsequently in each BO iteration, an exhaustive search over all unexplored points is performed to find the next query point optimizing EHVI. With IRS, such an exhaustive search can cost more than one hour in each iteration, which is comparable to the running time of the micromechanical model. The expensive time cost for BO itself makes this impractical. On the other hand, with our new CLM-based algorithm the exhaustive search only takes a fraction of a second, orders of magnitude faster than the actual micromechanical simulations, reducing significantly the cost associated to the optimal exploration of expensive design spaces.

We carried out 10 BO runs with the initial set of 20 samples and then sequential Bayesian experimental design with 100 following BO iterations. Performance of the sampling runs from Bayesian experimental design is assessed by the dominated hypervolume and compared against 10 runs with 120 random sampling evaluations. The performance comparison is shown in Fig.

5. From the figure, we can see in the first 20 evaluations, there are barely no difference between the two sampling methods, since both methods are actually getting random samples. But after 20 initial evaluations, Bayesian experimental design performs much better than random sampling. After 100 evaluations, most Bayesian optimization runs begin to saturate with the variance converging to zero, demonstrating that the procedure has identified the Pareto front.
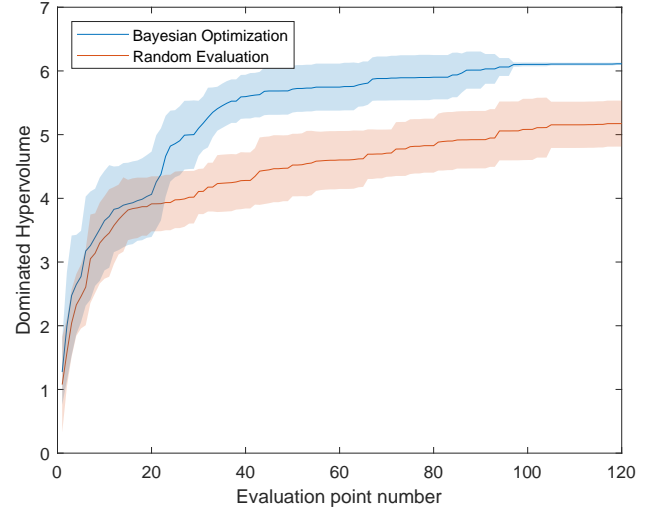

Figure 5: Dominated volume change (average shown as the lines and standard deviation as shaded regions) over the iterations by Bayesian Optimization and Random Sampling. Each Bayesian optimization starts with an initial set of 20 random samples.

6   Conclusions

We focus on improving the efficiency of calculating the exact EHVI, especially for cases with many objectives ($m \geq 3$). We have developed three fast exact EHVI calculation algorithms by integrating our new closed-form integral computation for hyperrectangles with three hypervolume decomposition methods. Experiments have shown that with the same decomposition method, our new algorithms are faster than state-of-the-art counterparts, also confirming the derived computation complexity. Moreover, our WFG-based algorithm can speed up the existing algorithm by more than two orders of magnitude when $m \geq 3$, enabling much more efficient Bayesian experimental design. In materials science, the need to carry out optimal sequential experiments over multi-objective spaces is necessary and approaches such as the ones demonstrated here empower autonomous materials discovery as demonstrated in our Shape Memory Alloy case study. This contribution, however, has a much wider applicability, beyond materials science.

Bibliography

Nicola Beume, Carlos M. Fonseca, Manuel López-Ibáñez, Luís Paquete, and Jan Vahrenhold. On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, 2009.

Joseph E. Cadman, Shiwei Zhou, Yuhang Chen, and Qing Li. On design of multi-functional microstructural materials. *Journal of Materials Science*, 48(1): 51–66, 2013.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

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

A Cox, B Franco, S Wang, T Baxevanis, I Karaman, and DC Lagoudas. Predictive modeling of the constitutive response of precipitation hardened ni-rich niti. *Shape Memory and Superelasticity*, 3(1):9–23, 2017.

Michael T. M. Emmerich and Carlos M. Fonseca. Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 121–135. Springer, 2011.

Michael T. M. Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Rapport technique, Leiden University*, 34, 2008.

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

Michael T. M. Emmerich, Kaifeng Yang, André Deutz, Hao Wang, and Carlos M. Fonseca. A multicriteria generalization of bayesian global optimization. In *Advances in Stochastic and Deterministic Global Optimization*, pages 229–242. Springer, 2016.

Paul Feliot, Julien Bect, and Emmanuel Vazquez. A bayesian approach to constrained single-and multiobjective optimization. *Journal of Global Optimization*, 67(1-2):97–133, 2017.

Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

Iris Hupkens, André Deutz, Kaifeng Yang, and Michael Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 65–79. Springer, 2015.

Andrzej Jaszkiewicz. Improved quick hypervolume algorithm. *Computers & Operations Research*, 90:72–83, 2018.

Renaud Lacour, Kathrin Klamroth, and Carlos M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347–360, 2017.

Chang Luo, Koji Shimoyama, and Shigeru Obayashi. A study on many-objective optimization using the kriging-surrogate-based evolutionary algorithm maximizing expected hypervolume improvement. *Mathematical Problems in Engineering*, 2015, 2015.

Andrew McAfee, Erik Brynjolfsson, Thomas H. Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard business review*, 90(10):60–68, 2012.

Jagdish K. Patel and Campbell B. Read. *Handbook of the normal distribution*, volume 150. CRC Press, 1996.

Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

Foster Provost and Tom Fawcett. Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59, 2013.

Luís M. S. Russo and Alexandre P. Francisco. Quick hypervolume. *IEEE Transactions on Evolutionary Computation*, 18(4):481–502, 2014.

Lyndon While, Philip Hingston, Luigi Barone, and Simon Huband. A faster algorithm for calculating hypervolume. *IEEE transactions on evolutionary computation*, 10(1):29–38, 2006.

Lyndon While, Lucas Bradstreet, and Luigi Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1): 86–95, 2012.

Marco Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12:51, 2012.

Kaifeng Yang, Michael T. M. Emmerich, André Deutz, and Carlos M. Fonseca. Computing 3-d expected hypervolume improvement and related integrals in asymptotically optimal time. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 685–700. Springer, 2017.

Chao Yu, Guozheng Kang, and Qianhua Kan. A micromechanical constitutive model for anisotropic cyclic deformation of super-elastic niti shape memory alloy single crystals. *Journal of the Mechanics and Physics of Solids*, 82:97–136, 2015.