On the Complexity of Computing the Hypervolume Indicator

N. Beume[*], C. M. Fonseca[†], M. López-Ibáñez[‡], L. Paquete[§], J. Vahrenhold

February 5, 2009

Abstract

The goal of multi-objective optimization is to find a set of best compromise solutions for typically conflicting objectives. Due to the complex nature of most real-life problems, only an approximation to such an optimal set can be obtained within reasonable (computing) time. To compare such approximations, and thereby the performance of multi-objective optimizers providing them, unary quality measures are usually applied. Among these, the *hypervolume indicator* (or *S-metric*) is of particular relevance due to its favorable properties. Moreover, this indicator has been successfully integrated into stochastic optimizers, such as evolutionary algorithms, where it serves as a guidance criterion for finding good approximations to the Pareto front.

Recent results show that computing the hypervolume indicator can be seen as solving a specialized version of Klee's Measure Problem. In general, Klee's Measure Problem can be solved with $\mathcal{O}(n \log n + n^{d/2} \log n)$ comparisons for an input instance of size n in d dimensions; as of this writing, it is unknown whether a lower bound higher than $\Omega(n \log n)$ can be proven.

In this article, we derive a lower bound of $\Omega(n \log n)$ for the complexity of computing the hypervolume indicator in any number of dimensions $d > 1$ by reducing the so-called UniformGap problem to it. For the three dimensional case, we also present a matching upper bound of $\mathcal{O}(n \log n)$ comparisons that is obtained by extending an algorithm for finding the maxima of a point set.

keywords Multi-objective optimization, performance assessment, complexity analysis, computational geometry

1 Motivation and Introduction to Multi-objective Optimization

In multi-objective optimization, the problem is to find best possible compromise solutions which cannot be improved according to one objective without deteriorating the others. This type of problems arises in all kind of industrial application areas ranging from production to service industries, entertainment, and many others. However, since many real-world problems cannot be expected to be solved to optimality (whether at all or within a reasonable amount of computing time), the goal is usually to obtain a good approximation to the optimal set of solutions within a reasonable amount of time. With this aim, many stochastic optimizers, such as multi-objective evolutionary algorithms [1, 2], have been proposed in the literature. To evaluate and compare the (sets of) compromise solutions suggested by

[*]Nicola Beume and Jan Vahrenhold are with Chair of Algorithm Engineering at the Faculty of Computer Science, Technische Universität Dortmund, 44221 Dortmund, Germany. E-mail: {nicola.beume,jan.vahrenhold}@tu-dortmund.de

[†]Carlos M. Fonseca is with the Department of Electronic Engineering and Informatics, Faculty of Science and Technology, Universidade do Algarve, 8005-139 Faro, Portugal, and CEG-IST – Centre for Management Studies, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2780-990 Porto Salvo, Portugal. E-mail: cmfonsec@ualg.pt

[‡]Manuel López-Ibáñez is with the Centre for Emergent Computing, School of the Built Environment, Napier University, EH10 5DT, Edinburgh, UK. E-Mail: m.lopez-ibanez@napier.ac.uk

[§]Luís Paquete is with the CISUC – Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra, 3030-290 Coimbra, Portugal. E-mail: paquete@dei.uc.pt

these optimizers, quality indicators have been developed. Of major importance among these is the hypervolume indicator whose computational complexity is analyzed in this work.

Without loss of generality, we consider maximization problems. A multi-objective optimization problem consists of d objective functions $f_1, \ldots, f_d$, which map an m-dimensional vector in the search space onto a d-dimensional vector in the objective space. Among all such d-dimensional objective vectors, a strict partial order can be defined as follows: a point $\mathbf{p} = (p_1, \ldots, p_d)$ *dominates* a point $\mathbf{q}$ iff $q_i \leq p_i$ holds for all $1 \leq i \leq d$ (i.e. $\mathbf{q} \preceq \mathbf{p}$) and $\mathbf{q} \neq \mathbf{p}$. Two distinct points are *incomparable* iff neither point dominates the other. Points that are not dominated within a set are the best ones, and are referred to as *non-dominated* or *maximal*. The elements of the search space that generate the non-dominated elements of the objective space form the *Pareto set* of the problem, and the set of the corresponding images in the objective space is called *Pareto front*.

Multi-objective optimizers generate approximations of the Pareto front. To assess the performance of different optimizers, their resulting approximations have to be compared. This may be performed by extending the Pareto dominance relation to sets of points (see e.g. Zitzler *et al.* [3]), but, in this case, good Pareto front approximations are often incomparable to one another. Therefore, many researchers have proposed *quality indicators* for the sets of compromise solutions generated by multi-objective optimizers that—according to several quality criteria—map such sets onto scalar values and thus allow for an easy comparison.

There is a general consensus about two (informal) criteria of quality: An approximation of the optimal set is good if (1) its points are 'close' to the Pareto front, (2) the points are 'well-distributed' along the whole Pareto front. Additionally, we find it worth mentioning that a good approximation shall contain 'many' non-dominated points. An in-depth overview of quality measures and their properties is given by Zitzler *et al.* [3].

The hypervolume indicator (or *S-metric*, *Lebesgue measure*), introduced by Zitzler and Thiele [4], is regarded as a rather fair measure since it respects all the aspects mentioned above and has favorable theoretical properties [3] giving it an outstanding importance among quality indicators. Formally, the hypervolume indicator may be defined as follows:

Definition 1 *Given a finite set $\mathcal{P}$ of points in the positive orthant $\mathbb{R}^d_{\geq 0}$, the hypervolume indicator is defined as the d-dimensional volume of the hole-free orthogonal polytope*

$$\Pi^d = \{\mathbf{x} \in \mathbb{R}^d_{\geq 0} : \mathbf{x} \preceq \mathbf{p} \text{ for some } \mathbf{p} \in \mathcal{P}\}.$$

This polytope corresponds to the space which is dominated by at least one point in the set $\mathcal{P}$.

The dominated hypervolume is calculated with respect to a reference point $\mathbf{r}$ which, in the above definition, is chosen to coincide with the origin. This definition also assumes maximization of all objectives and strictly positive objective values. Whenever this is not the case, suitable affine transformations may be applied to each objective separately. Fig. 1 shows an example of such a polytope in two dimensions; the hypervolume indicator consists of the area of the shaded region. Note that the point depicted in light gray does not contribute to this area, as it is not a maximal element of $\mathcal{P}$. Since non-maximal (or dominated) points do not contribute to the value of the indicator, the set $\mathcal{P}$ is often assumed to coincide with the set of its maxima (or non-dominated elements).

Due to its properties, the hypervolume indicator has been applied in various empirical comparative studies. In addition, it has been used as a criterion for selecting non-dominated solutions for a size-limited archive [5], and has been integrated into multi-objective optimizers—mainly evolutionary multi-objective optimization algorithms (EMOA) [6–8]—as a single-objective substitute function to guide the optimization process. When the indicator is applied to update an archive, and especially when it is used within the selection operator of an EMOA, its calculation has to be performed several times per iteration. Thus, fast computation of the hypervolume indicator is essential for modern

EMOA and its computational complexity is currently an important research problem in the evolutionary multi-objective optimization field.

With the growing interest in the computation of the hypervolume indicator in the last few years, upper bounds on its asymptotic performance have been devised [9–16]. In this article, a lower bound of $\Omega(n \log n)$ comparisons for the computation of the hypervolume indicator in any number of dimensions $d > 1$ is proven by reduction from the UniformGap problem. In addition, an $\mathcal{O}(n \log n)$ time algorithm for the three-dimensional case is described. The combination of these results shows that the lower bound is tight for $d = 3$, and that the algorithm proposed is optimal.

In the following section, an upper bound is derived by considering the hypervolume indicator as a special case of Klee's Measure Problem. Section III presents the new lower bound and Section IV contains the description of an optimal algorithm for computing the hypervolume indicator in three dimensions. Concluding remarks are given in Section V.

2 An Upper Bound with Klee's Measure Problem

Klee's Measure Problem, or the problem of computing the length of the union of a collection of intervals on the real line, was formulated by Klee, who also showed that it can be solved in optimal $\mathcal{O}(n \log n)$ time [17]. Bentley [18] generalized this problem to $d \geq 2$ dimensions, and presented an upper bound of $\mathcal{O}(n^{d-1} \log n)$. Later, van Leeuwen and Wood [19] improved this result to $\mathcal{O}(n^{d-1})$ when $d \geq 3$. The fastest known algorithm to date for three or more dimensions is due to Overmars and Yap [20], and runs in $\mathcal{O}(n^{d/2} \log n)$ time. The d-dimensional version of Klee's Measure Problem is also known as the problem of computing the *measure of a union of hyper-rectangles* [21].

Fonseca *et al.* [14] and Beume [15] independently described the dominated hypervolume for a point set $\mathcal{P} \subset \mathbb{R}^d_{\geq 0}$ as a special case of Klee's Measure Problem. Indeed, the polytope Π^d is patterned by the collection of hyper-rectangles $\{R_{\mathbf{p}}\}_{\mathbf{p} \in \mathcal{P}}$ with $R_{\mathbf{p}} := \{\mathbf{x} \in \mathbb{R}^d_{\geq 0} : \mathbf{x} \preceq \mathbf{p}\}$ spanned by the points in $\mathcal{P}$ and the reference point $\mathbf{r} = \mathbf{0} \in \mathbb{R}^d$. This set of hyper-rectangles is a valid input for Klee's Measure Problem and the corresponding output is the desired hypervolume (see Fig. 2 for an example in two dimensions). This immediately establishes an upper bound of $\mathcal{O}(n \log n + n^{d/2} \log n)$ time, which is lower than the time complexity of various algorithms [11–13] proposed previously for the computation of the hypervolume indicator when $d \geq 3$ (the $n \log n$ term accounts for the special cases where $d \leq 2$). By simplifying Overmars and Yap's algorithm to take advantage of the fact that all rectangles are anchored at the same point (the reference point $\mathbf{r}$), Beume and Rudolph [16] obtained an upper bound of $\mathcal{O}(n \log n + n^{d/2})$, which is the best upper bound currently known for $d > 3$.

3 A Lower Bound with the UniformGap Problem

It seems natural that the computation of the hypervolume indicator may actually be easier than the general form of Klee's Measure Problem, since all rectangles are anchored at the same point, namely the reference point. In particular, the hypervolume indicator does not include the disjoint-interval case used by Fredman and Weide [22] to obtain a lower bound of $\Omega(n \log n)$ for Klee's Measure Problem. When $d = 1$, computing the hypervolume indicator requires only $n - 1 \in \mathcal{O}(n)$ comparisons, since this is equivalent to determining the single maximal element of P. However, Theorem 1 shows that the case $d = 1$ is the only case where the (known) lower bounds for Klee's Measure Problem and for the problem of computing the dominated hypervolume, the so-called DominatedHypervolume problem are different.

Theorem 1 *Solving the* DominatedHypervolume *problem for an n-element point set in $\mathbb{R}^d$, $d \geq 2$, has a time-complexity of $\Omega(n \log n)$.*

In the next subsections, we first explain the method used to derive this lower bound, and then we use this method to provide a proof for Theorem 1.

3.1 Methods for Deriving Lower Bounds

The model of computation we are working in, the fixed-degree algebraic decision tree, is the standard model used in computational geometry (and algorithmic complexity) and is used to prove lower bounds for (geometric) decision problems. In a nutshell, an algebraic decision tree captures the behavior of a (loop-unrolled) algorithm that branches depending on the outcome of evaluations of bounded-degree polynomials. A lower bound on the complexity of a given problem can then be derived by establishing a lower bound on the depth of any such tree resembling any valid algorithm to solve this problem. This model is a generalization of the tree-based model used to establish an $\Omega(n \log n)$ lower bound for comparison-based sorting (as discussed by, e.g., Cormen *et al.* [23]). For a more in-depth exposition, we refer the reader to the textbook by Preparata and Shamos [21, Sec. 1.4].

Once a lower bound for some problem ProblemA has been established, a lower bound for a problem ProblemB can be derived from ProblemA's lower bound if we can prove that ProblemB can be used to solve any problem instance of ProblemA, or that ProblemA can be *reduced* to ProblemB, as illustrated in Fig. 3. More precisely, we need to establish a transformation $\tau : dom(\text{ProblemA}) \rightarrow dom(\text{ProblemB})$ and a transformation $\tau' : im(\text{ProblemB}) \rightarrow im(\text{ProblemA})$ where $dom(\cdot)$ denotes the set of all input instances and $im(\cdot)$ denotes the set of solutions to the given problem. Transformation τ is used to transform any input instance $\mathcal{A}$ for problem ProblemA into an input instance $\tau(\mathcal{A})$ for problem ProblemB, and transformation τ' is used to transform the result (of solving) ProblemB$(\tau(\mathcal{A}))$ into a valid solution for ProblemA.

For the correctness of the transformation we require that for any problem instance $\mathcal{A}$, we have ProblemA$(\mathcal{A}) = \tau'(\text{ProblemB}(\tau(\mathcal{A})))$, i.e., the result ProblemA$(\mathcal{A})$ obtained by running any algorithm for directly solving ProblemA for $\mathcal{A}$ has to be exactly the same as the solution that is obtained via the above transformation. To be able to obtain a meaningful lower bound for ProblemB, we also require that the asymptotic complexity $g(n)$ of both τ and τ' is strictly less than the lower bound for ProblemA (here, n is the input size). If this is the case, we can conclude that the lower bound for ProblemA is a lower bound for ProblemB as well—for more details, we again refer the reader to Preparata and Shamos [21, Sec. 1.4].

If $g(n) \in \mathcal{O}(n)$ the above transformation is called a *linear-time reduction* from ProblemA to ProblemB.

3.2 A Proof for Theorem 1

Proof. Based upon the approach presented in the previous subsection, the lower bound for the DominatedHypervolume problem is established by a linear-time reduction from the UniformGap problem. The latter problem is to decide for a given n-element point set on the real line whether the points are uniformly spaced, and has been shown to exhibit an $\Omega(n \log n)$ lower bound–see, e.g. Preparata and Shamos [21]. To prove the claimed lower bound for DominatedHypervolume, we need to establish that every problem instance of UniformGap can be transformed (in linear time) into an instance of DominatedHypervolume and that the result of solving the DominatedHypervolume problem for this particular instance can be used to obtain the correct answer for UniformGap problem for the given input instance.

Let $\mathcal{P} := \{x^{(1)}, \ldots, x^{(n)}\}$ be any (unordered) set of points on the real line. To solve UniformGap$(\mathcal{P})$, we first construct a set $\mathcal{P}'$ of two-dimensional points from $\mathcal{P}$ using the embedding $\mathbf{p}^{(i)} \mapsto (x^{(i)}, -x^{(i)})$. In linear time, we then translate the embedded point set such that all points have strictly positive coordinates. More precisely, we compute $m := (-\min\{p_1 \mid \mathbf{p} \in \mathcal{P}'\} + \delta, -\min\{p_2 \mid \mathbf{p} \in \mathcal{P}'\} + \delta)$, where $\delta > 0$ is a small positive constant, and let $\mathcal{Q}$ be the Minkowski sum of $\mathcal{P}'$ and $\{m\}$.

All points of $\mathcal{Q}$ lie on a diagonal line in the first quadrant (Fig. 4, top). We now run any algorithm for solving the DominatedHypervolume$(\mathcal{Q}, \mathbf{r})$ with the origin as the reference point $\mathbf{r}$ and obtain some real number a that gives the area of the dominated hypervolume.

To obtain the answer for UniformGap$(\mathcal{P})$, we first observe that the volume a of the dominated area can be written as the sum $a = a_1 + a_2 + a_3$ of the volumes of three disjoint subareas (Fig. 4, top). The volumes of two of these areas are independent of whether or not the points in $\mathcal{P}$ are equally spaced. More precisely, we have $a_1 = p_1^{\min} \cdot p_2^{\min}$ and $a_2 = (p_1^{\max} - p_1^{\min}) \cdot p_2^{\max}$, where $\mathbf{p}^{\min}$ is the point with minimal first coordinate and $\mathbf{p}^{\max}$ is the point with maximal one. Both $\mathbf{p}^{\min}$ and $\mathbf{p}^{\max}$ can be determined from $\mathcal{P}$ in linear time.

Lemma 1 *In the situation of Fig. 4 (top), the area a_3 is maximal if and only if the points in $\mathcal{P}$ are equally spaced.*

Proof. Let us assume that a_3 is maximal but that not all points in $\mathcal{P}$ are equally spaced. Then there exist three points $\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{p}''$ in $\mathcal{Q}$ that are consecutive in sorted x_1-order such that $|p_1' - p_1| \neq |p_1'' - p_1'|$ (note that for the purpose of this proof we do not need to actually find these points; it is sufficient to know that they exist). Without loss of generality, we have the situation depicted in Fig. 4 (bottom). The contribution of the point $\mathbf{p}'$ then is the area of the dark rectangle, or $|p_1' - p_1| \cdot |p_2' - p_2''|$. Since $\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{p}''$ lie on a line, the sum $|p_1' - p_1| + |p_2' - p_2''|$ and thus the perimeter of the dark rectangle is constant. For a given perimeter, a rectangle has maximal area if and only if it is a square. Thus, we can move $\mathbf{p}'$ such that $|p_1' - p_1| = |p_2' - p_2''|$, i.e., make $\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{p}''$ equally spaced, while increasing the area a_3. This is the desired contradiction. Conversely, we see that for an equally spaced set of points, every three consecutive points are equally spaced, so the local contribution of each point is a square. Again, trying to make any three consecutive points non-equally spaced results in a decrease of the contribution of the middle point and the claim follows.[1]

■

Continuing the proof of Theorem 1, Lemma 1 is used to provide the information needed to convert the answer for DominatedHypervolume$(\mathcal{Q})$ into an answer for UniformGap$(\mathcal{P})$. To this end, we compute the hypervolume $\hat{a}$ that the points in $\mathcal{Q}$ would dominate if they were equally spaced for some inter-point distance ε. Since the points $\mathbf{p}^{\min}$ and $\mathbf{p}^{\max}$ already have been found in linear time, we can immediately compute $\varepsilon := (p_1^{\max} - p_1^{\min})/(n-1)$. Furthermore, we have $\hat{a} = a_1 + a_2 + \hat{a}_3$ (note that a_1 and a_2 are independent of whether or not the points are equally spaced) where $\hat{a}_3 = \frac{1}{2}(p_1^{\max} - p_1^{\min})^2 - \frac{1}{2}\frac{(p_1^{\max} - p_1^{\min})^2}{n-1}$. The formula for $\hat{a}_3$ is easily verified to give the area of the isosceles right triangle spanned by $\mathbf{p}^{\max}$, $\mathbf{p}^{\min}$, and $(p_1^{\min}, p_2^{\max})$ minus $(n-1)$ times the area of an isosceles right triangle with leg-length ε.

To obtain the answer for UniformGap$(\mathcal{P})$, we simply check whether the hypervolume a reported by DominatedHypervolume$(\mathcal{Q})$ is strictly smaller than $\hat{a}$. If so, we know that $a_3 < \hat{a}_3$, and thus, by Lemma 1, the points are not equally spaced. Consequently the points are equally spaced if and only if $a = \hat{a}$.

Since both the transformation of the input and the transformation of the output of DominatedHypervolume$(\mathcal{Q})$ take linear time and since the algorithm given above solves the UniformGap problem for $\mathcal{P}$, we have established the claimed lower bound for the DominatedHypervolume problem. By embedding a two-dimensional point set into higher-dimensional space (setting each coordinate of higher dimensions to 1), we have also derived a lower bound for any dimension $d > 2$. ■

[1] An alternative proof may be constructed based on purely analytical arguments [24].

4 An Optimal Algorithm for the Three-Dimensional Case

In Theorem 1, a set of maximal (or non-dominated) points was constructed to prove the lower bound. This shows that even the knowledge that a particular input instance contains only non-dominated points does not help to accelerate the computation of the hypervolume indicator. On the other hand, the fact that dominated points do not contribute to the value of dominated hypervolume suggests that identifying them may be useful, if not necessary, in order to compute the indicator. Therefore, an algorithm for the maxima problem would seem to be a good starting point for the development of an algorithm for computing the hypervolume indicator.

In this section, we present an algorithm for the problem of computing the hypervolume indicator for a point set $\mathcal{P} \subset \mathbb{R}^3$, and the time complexity of this algorithm is analyzed to match the lower bound given by Theorem 1. Without loss of generality, we assume that all points have positive coordinates in all dimensions.

4.1 Description of the General Approach

The algorithm is a rather natural extension of Kung *et al.*'s algorithm for computing the set of maxima in three dimensions [25]. We recall that a point is maximal if and only if no other point has at least as large a coordinate in each dimension with one coordinate being strictly larger (cf. Section 1). This implies that processing the points in decreasing order with respect to the d-th dimension reduces the (static) d-dimensional problem to a sequence of $(d-1)$-dimensional sub-problems.[2]

This property makes the problem amenable to a *space-sweep*-based solution. Sweeping is a well-known paradigm in Computational Geometry: the d-dimensional problem instance is swept over with a $(d-1)$-dimensional hyperplane that moves along one coordinate axis. The hyperplane stops at a finite number of positions, usually defined by the objects in the problem instance, computes a solution to a properly defined $(d-1)$-dimensional sub-problem, and finally combines the solutions to the sub-problems into a solution for the global problem. The main issue (besides properly defining the sub-problems) is to efficiently maintain the objects relevant to the current sub-problem; usually they are maintained in a dynamic data structure, the *sweep structure*.

For the case of computing a dominated hypervolume in three dimensions, the sub-problems are given by hyperplanes parallel to the (x_1, x_2)-plane. These hyperplanes partition the dominated hypervolume into slices that are extruded rectilinear polygons–see Fig. 5, left. Restated in terms of dominated hypervolumes, the algorithm of Kung *et al.* is a space-sweep algorithm that processes the (three-dimensional) points in decreasing x_3-order (i.e. by decreasing x_3-coordinate) and keeps track of the (x_1, x_2)-projection of the boundary of the dominated hypervolume above the sweeping plane–see Fig. 5, right. This boundary is a monotone rectilinear polyline (monotonically increasing in x_1-direction and monotonically decreasing in x_2-direction) and thus the points can be maintained efficiently in increasing x_1-order by using a balanced binary search tree T as the sweep structure.

On a high level, the proposed algorithm works as follows: Whenever the sweeping hyperplane encounters a point $\mathbf{p}$, the projection of this point is inserted into T. Since the projection of $\mathbf{p}$ can dominate the projection of other points inserted before (see Fig. 5, right), the algorithm has to scan the data structure from the insertion point to find (and remove) all dominated projections. Finally, the (two-dimensional) area A of the slice starting at $x_3 = p_3$ has to be computed. When the sweeping hyperplane encounters the next point $\mathbf{q}$, the volume of this slice can be computed by $A \cdot (p_3 - q_3)$ (note that the algorithm sweeps downwards), and this value is added to the global volume computed so far.

[2] Depending on the application at hand, the input set may contain duplicates which do not dominate each other by definition. The algorithm presented in this paper does not depend on all coordinate values being distinct, and, as such, handles duplicates transparently.

4.2 Detailed Description

To simplify the description, we also add two dummy points $(0, \infty)$ and $(\infty, 0)$ to T. Since all points $\mathbf{p} \in \mathcal{P}$ have positive (yet finite) coordinates, this ensures that each such point $\mathbf{p}$ inserted into T will have a successor $succ_1(\mathbf{p})$ and a predecessor $pred_1(\mathbf{p})$ï¿½in T with respect to the increasing order on the x_1-coordinate.

A description of our algorithm is given as Algorithm 1. Let us assume that the (x_1, x_2)-projection of the boundary of the dominated hypervolume above the sweeping plane is maintained using a binary search tree T on the x_1-coordinates.

The algorithm then processes the next point $\mathbf{p}^{(i)}$ (in decreasing x_3-order) by first locating the successor $\mathbf{q} := succ_1(\mathbf{p}^{(i)})$ of $\mathbf{p}$ in T (Step 3a). If the x_2-coordinate of $\mathbf{q}$ is larger than the x_2-coordinate of $\mathbf{p}^{(i)}$, $\mathbf{q}$ dominates $\mathbf{p}^{(i)}$ in all dimensions, and nothing needs to be done (see the point $\mathbf{p}'$ in Fig. 6, top). If, on the other hand, the x_2-coordinate of $\mathbf{q}$ is smaller than the x_2-coordinate of $\mathbf{p}^{(i)}$, $\mathbf{p}^{(i)}$ is added to the set of maximal points (see the point $\mathbf{p}$ in Fig. 6, top). This update also affects the boundary of the dominated hypervolume, and the algorithm reflects this update by deleting all points between $succ_2(\mathbf{p}^{(i)})$ and $\mathbf{q}\,(= succ_1(\mathbf{p}^{(i)}))$ (Step 3b,iii). The point $succ_2(\mathbf{p}^{(i)})$ can be found by going backwards in T from $\mathbf{q}$ and exploiting the fact that the x_1-order of the points in T corresponds to their reverse x_2-order (see Fig. 6, bottom).

Our algorithm augments the above approach by simultaneously maintaining the *volume* V of the dominated hypervolume seen so far. To do so, the algorithm maintains the area A in the (x_1, x_2)-projection that is dominated by the points currently stored in T and the last point $\mathbf{z}$ added to T. Whenever a new point $\mathbf{p}^{(i)}$ is identified as a non-dominated point, the dominated volume seen so far is increased by $A \cdot (z_3 - p_3^{(i)})$ (Step 3b,i), and $\mathbf{z}$ is updated to $\mathbf{p}^{(i)}$ (Step 3b,ii). Then, A is updated to reflect the changed boundary stored in T (Step 3b,iii + iv). At the end of the algorithm, we have to add the volume of the slice between the last maximal point and the (x_1, x_2) plane, i.e. the volume $(A \cdot z_3)$ (see Step 4).

4.3 Analysis

The efficiency of the proposed algorithm depends on the efficiency of the implementation of the sweep structure T. If we implement T using a balanced binary search tree, e.g. a red-black tree or an AVL-tree [23], the cost for a single update or search operation including *pred* and *succ* is logarithmic in the number of elements currently stored in T. Since each point is added to T at most once, T cannot contain more than n elements, and thus the cost for each update or search operation is in $\mathcal{O}(\log n)$.

The cost of updating A (Step 3(b)iii(A) and 3(b)iv) is linear in the number of updates to T, since all relevant volumes can be computed in constant time relative to $(succ_2(\mathbf{p}^{(i)})_1, succ_1(\mathbf{p}^{(i)})_2)$—see Fig. 6, bottom.

The running time of the algorithm now is easily seen to be in $\mathcal{O}(n \log n)$, since each point can be added to (and removed from) T at most once. All updates to T have logarithmic cost and each associated update to A and to V can be done in constant time per point. Thus, the global cost of all updates and of the initial sorting step is $\mathcal{O}(n \log n)$, which proves Theorem 2.

Theorem 2 *Computing the hypervolume dominated by a set of n points in $\mathbb{R}^3$ can be done in optimal $\mathcal{O}(n \log n)$ time.*

The presented algorithm can be generalized to higher dimensions resulting in an upper bound of $\mathcal{O}(n^{d-2} \log n)$ for $d > 2$ (see Fonseca *et al.* [14]), though, for $d \geq 4$, a better bound can be obtained using the algorithm by Beume and Rudolph [16] with currently lowest worst-case complexity.

5 Concluding Remarks

The efficient computation of the hypervolume indicator (or S-metric) is of particular relevance, especially for its online application within multi-objective optimizers.

In this article, the computational complexity of the hypervolume indicator was analyzed by relating it to problems from computational geometry. By casting the hypervolume indicator as a special case of Klee's Measure Problem in d dimensions, the existence of algorithms with lower worst-case complexity than those currently used by practitioners in the field [9–13] was readily established. In particular, the hypervolume indicator may be computed in $\mathcal{O}(n \log n + n^{d/2})$ time [16].

A lower bound of $\Omega(n \log n)$ for this problem was obtained by reduction from the geometric problem of deciding whether n points are equally spaced on a line. The proof exploits the fact that the maximal value of the indicator for a finite set of points located on a certain linear Pareto front is achieved only when the distance between consecutive points is constant.

Finally, a dedicated algorithm for the case of $d = 3$ was developed based upon an algorithm for the problem of identifying the maximal elements of a set. The relation between the two problems arises due to the fact that only non-dominated (or maximal) points contribute to the value of the indicator. As the obtained upper bound of $\mathcal{O}(n \log n)$ matches the proved lower bound, the proposed algorithm is asymptotically optimal, which improves over the previously best result. In addition, its conceptual simplicity makes it possible to implement it efficiently, without hiding large constants in the $\mathcal{O}$-notation [14]. However, it is still possible that, in practice, other algorithm implementations may run faster on given input data sets.

The improvement of the current lower and upper bounds when $d > 3$ remains an open problem. Future research shall deal with the development of more efficient algorithms for the hypervolume indicator in an arbitrary number of dimensions by further exploiting the relationship with known geometrical problems and taking advantage of existing results and insights from computational geometry. Another direction for future work is the empirical evaluation of those algorithms in comparison to existing ones.

References

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[2] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, New York, 2002.

[3] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[4] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - A comparative case study," in *Proc. of PPSN V, Fifth International Conference on Parallel Problem Solving from Nature*. Springer Verlag, Berlin, 1998, pp. 292–301.

[5] J. Knowles and D. Corne, "Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100–116, April 2003.

[6] S. Huband, P. Hingston, L. While, and L. Barone, "An evolution strategy with probabilistic mutation for multi-objective optimisation," in *Proc. of the 2003 Congress on Evolutionary Computation (CEC'03)*, vol. 4. IEEE Press, 2003, pp. 2284–2291.

[7] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. of PPSN VIII, Eight International Conference on Parallel Problem Solving from Nature*. Springer Verlag, Berlin, 2004, pp. 832–842.

[8] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Proc. of EMO 2005, Third International Conference on Evolutionary Multi-Criterion Optimization*. Springer Verlag, Berlin, 2005, pp. 62–76.

[9] E. Zitzler, *Hypervolume Metric Calculation*. Computer Engineering and Networks Laboratory (TIK), Zürich, 2001, ftp://ftp.tik.ee.ethz.ch/pub/people/zitzler/hypervol.c.

[10] J. D. Knowles, "Local Search and Hybrid Evolutionary Algorithms for Pareto Optimization," Ph.D. dissertation, University of Reading, Reading, UK, January 2002.

[11] M. Fleischer, "The measure of pareto optima: Applications to multi-objective metaheuristics," in *Proc. of EMO 2003, Second International Conference on Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 2632. Springer Verlag, Berlin, 2003, pp. 519–533.

[12] L. While, "A new analysis of the LebMeasure algorithm for calculating hypervolume," in *Proc. of EMO 2005, Third International Conference on Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 3410. Springer Verlag, Berlin, 2005, pp. 326–340.

[13] L. While, L. Bradstreet, L. Barone, and P. Hingston, "Heuristics for optimising the calculation of the hypervolume for multi-objective optimisation problems," in *Proc. of the 2005 Congress on Evolutionary Computation (CEC'05)*, vol. 3. IEEE, 2005, pp. 2225–2232.

[14] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *Proc. of the 2006 Congress on Evolutionary Computation (CEC'06)*. IEEE Press, 2006, pp. 1157–1163.

[15] N. Beume, "Hypervolumen-basierte Selektion in einem evolutionären Algorithmus zur Mehrzieloptimierung," University of Dortmund, Department of Computer Science, Diploma Thesis, Algorithm Engineering Report TR06-2-002, 2006.

[16] N. Beume and G. Rudolph, "Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem," in *Proceedings of the Second IASTED Conference on Computational Intelligence*, B. Kovalerchuk, Ed. ACTA Press, Anaheim, 2006, pp. 231–236.

[17] V. Klee, "Can the measure of $\bigcup_1^n [a_i, b_i]$ be computed in less than $O(n \log n)$ steps?" *American Mathematics Monthly*, vol. 84, no. 4, pp. 284–285, Apr. 1977.

[18] J. L. Bentley, "Algorithms for Klee's rectangle problems," 1977, unpublished notes, Carnegie Mellon University, Computer Science Department. Referenced in [19].

[19] J. van Leeuwen and D. Wood, "The measure problem for rectangular ranges in d-space," *Journal of Algorithms*, vol. 2, no. 3, pp. 282–300, 1981.

[20] M. H. Overmars and C.-K. Yap, "New upper bounds in Klee's measure problem," *SIAM Journal on Computing*, vol. 20, no. 6, pp. 1034–1045, 1991.

[21] F. P. Preparata and M. I. Shamos, *Computational Geometry. An Introduction*, 2nd ed. Springer Verlag, Berlin, 1988.

[22] M. L. Fredman and B. Weide, "On the complexity of computing the measure of $\bigcup[a_i, b_i]$," *Communications of the ACM*, vol. 21, no. 7, pp. 540–544, July 1978.

[23] T. H. Cormen, C. E. Leierson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, Cambridge, 2001.

[24] M. Emmerich, A. Deutz, and N. Beume, "Gradient-based/Evolutionary Relay Hybrid for Computing Pareto Front Approximations Maximizing the S-Metric," in *Proc. 4th International Workshop on Hybrid Metaheuristics (HM 2007)*, ser. LNCS 4771, T. Bartz-Beielstein *et al.*, Eds. Springer, 2007, pp. 140–157.

[25] H. T. Kung, F. Luccio, and F. P. Preparata, "On finding the maxima of a set of vectors," *Journal of the ACM*, vol. 22, no. 4, pp. 469–476, Oct. 1975.

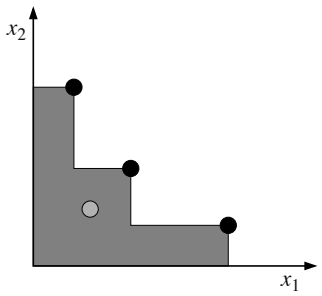

Figure 1: A set of points in the positive quadrant and the corresponding hole-free orthogonal polytope with the origin as the reference point. Maximal points are depicted black, non-maximal gray.

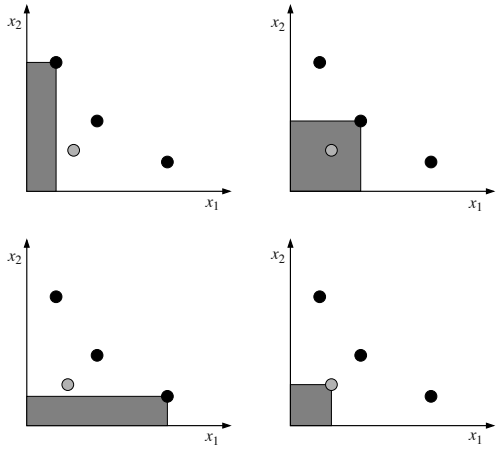

Figure 2: The hypervolume indicator as a special case of Klee's Measure Problem. The dominated hypervolume of the points is divided into rectangles spanned by a point and the reference point $\mathbf{r}$.

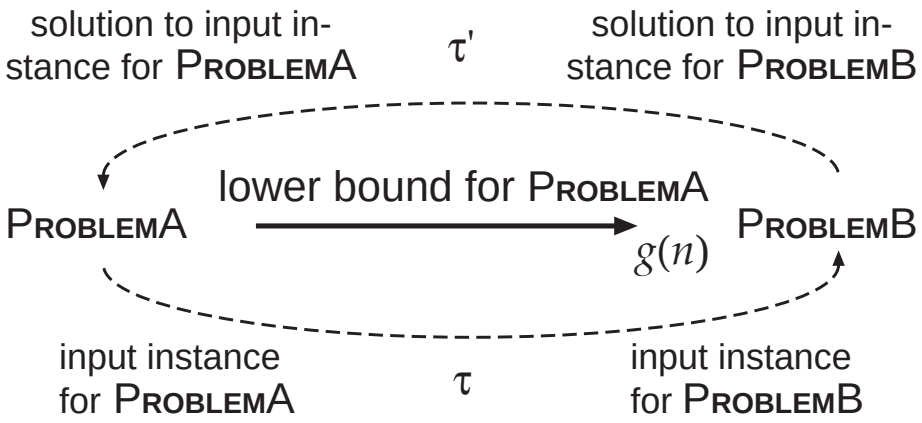

Figure 3: Transferring a lower bound by reduction from ProblemA to ProblemB.

Algorithm 1 Algorithm for computing the hypervolume V dominated by a set of n points in $\mathbb{R}^3$.

1. (*Initialize the algorithm*) Sort the points in decreasing x_3-order and let $(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(n)})$ be the resulting sequence of points. Initialize the search structure T by inserting two sentinel elements $(\infty, 0)$ and $(0, \infty)$ and set the volume V computed so far to 0.

2. (*Process the first point*) Store $\mathbf{p}^{(1)}$ in T and set the area A of the cross-section of the dominated hypervolume and the sweeping plane to $(p_1^{(1)} \cdot p_2^{(1)})$. Set $\mathbf{z}$, the lowest maximal point in the x_3-order seen so far, to $\mathbf{p}^{(1)}$.

3. (*Process all other points*) Process $\mathbf{p}^{(2)}$ to $\mathbf{p}^{(n)}$ in decreasing x_3-order. For each point $\mathbf{p}^{(i)}$ do the following:

   (a) Search T to find the point $\mathbf{q}$ that is immediately right of $\mathbf{p}^{(i)}$ (next higher x_1-value), i.e. $\mathbf{q} := succ_1(\mathbf{p}^{(i)})$.

   (b) If $\mathbf{p}^{(i)}$ is not dominated by $\mathbf{q}$ (i.e. if $q_2 < p_2^{(i)}$), update T and the variables A, V, and $\mathbf{z}$ as follows:

      i. (*Update* V) Since $\mathbf{p}^{(i)}$ is maximal, increase V by the volume of the slice between $\mathbf{z}$ (the last maximal point seen so far) and $\mathbf{p}^{(i)}$, i.e. set $V := V + A \cdot (z_3 - p_3^{(i)})$.

      ii. (*Update* $\mathbf{z}$) Set $\mathbf{z}$ to $\mathbf{p}^{(i)}$.

      iii. (*Process points dominated by* $\mathbf{p}^{(i)}$) Starting from $pred_1(\mathbf{q})$, search backwards in T until the first point $\mathbf{t}$ in T with $t_2 > p_2^{(i)}$, i.e. $\mathbf{t} = succ_2(\mathbf{p}^{(i)})$, is found (see Fig. 6, bottom). For each point $\mathbf{s}$ with $s_2 \leq p_2^{(i)}$ at which this search stops do the following:

         A. (*Update* A) Decrease A by the relative contribution of $\mathbf{s}$, i.e. set $A := A - (s_1 - (pred_1(\mathbf{s}))_1) \cdot (s_2 - q_2)$ (the dark rectangles in Fig. 6, bottom).

         B. (*Update* T) Since $\mathbf{s}$ is dominated by $\mathbf{p}^{(i)}$, remove $\mathbf{s}$ from T.

      iv. (*Update* A) Increase A by the relative contribution of $\mathbf{p}^{(i)}$, i.e. set $A := A + (p_1^{(i)} - t_1) \cdot (p_2^{(i)} - q_2)$ (the light rectangle in Fig. 6, bottom).

      v. (*Update* T) Since $\mathbf{p}^{(i)}$ is maximal, store $\mathbf{p}^{(i)}$ in T.

4. (*Computing the volume dominated by the last maximal point*) Increase V by the volume of the slice between the last maximal point $\mathbf{z}$ and the (x_1, x_2)-plane, i.e. set $V := V + A \cdot z_3$.

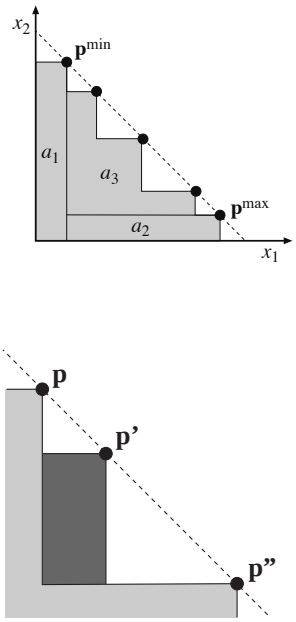

Figure 4: Top: Partitioning of the dominated hypervolume in three parts. Bottom: Three consecutive points that are not equally distributed. The dark gray area is maximal in case $\mathbf{p}'$ lies in the middle of $\mathbf{p}$ and $\mathbf{p}''$ and spans a square.

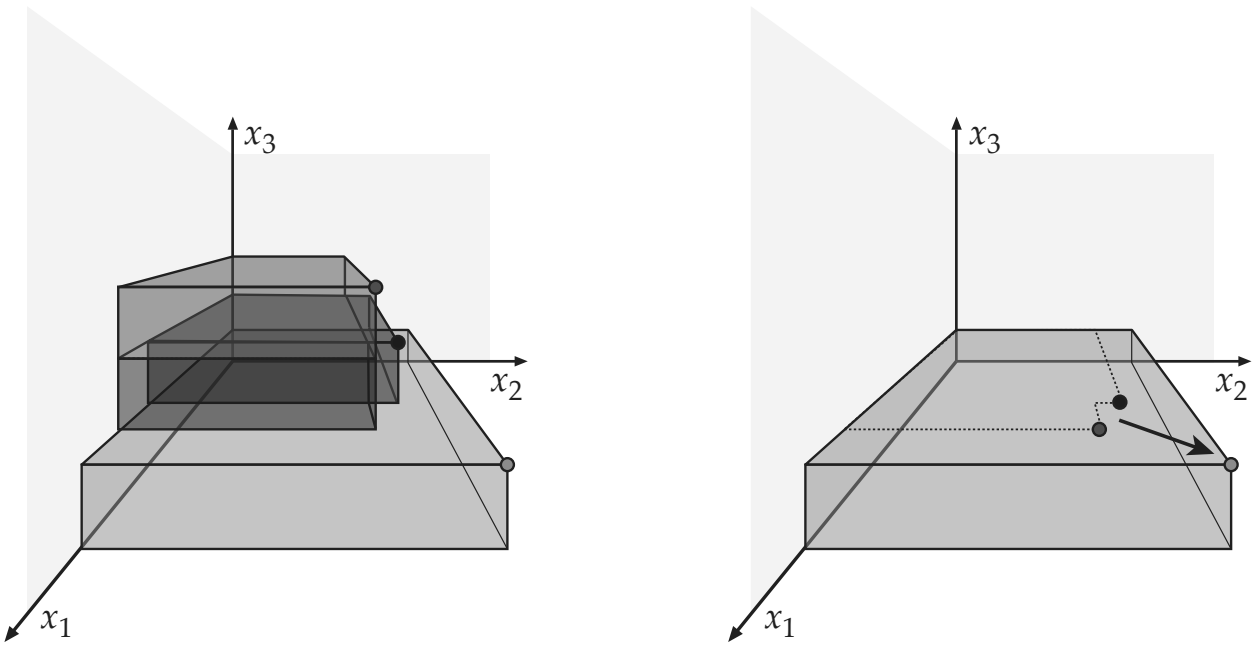

Figure 5: Computing the dominated hypervolume by sweeping in decreasing x_3-order. The dominated hypervolume is partitioned into slices (left) and each transition between two sub-problem involves updating a two-dimensional boundary (right).

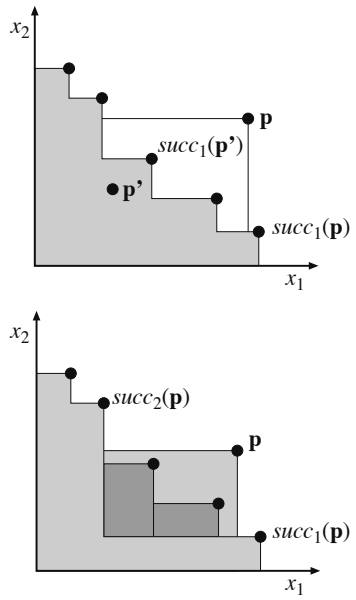

Figure 6: (x_1, x_2)-projection of the intersection of the dominated hypervolume and the sweepline. Classifying the next point during the sweep (top) and updating the projection and the area of the intersection (bottom).