THE HYPERVOLUME INDICATOR HESSIAN MATRIX:
ANALYTICAL EXPRESSION, COMPUTATIONAL TIME
COMPLEXITY, AND SPARSITY

A PREPRINT

André H. Deutz
Leiden Institute of Advanced Computer Science
Leiden University
2333CA Leiden
The Netherlands

Michael T.M. Emmerich
Leiden Institute of Advanced Computer Science
Leiden University
2333CA Leiden
The Netherlands

Hao Wang
Leiden Institute of Advanced Computer Science
and applied Quantum algorithms (aQa)
Leiden University
2333CA Leiden
The Netherlands

January 3, 2023

ABSTRACT

The problem of approximating the Pareto front of a multiobjective optimization problem can be
reformulated as the problem of finding a set that maximizes the hypervolume indicator. This paper
establishes the analytical expression of the Hessian matrix of the mapping from a (fixed size)
collection of \( n \) points in the \( d \)-dimensional decision space (or \( m \) dimensional objective space) to
the scalar hypervolume indicator value. To define the Hessian matrix, the input set is vectorized,
and the matrix is derived by analytical differentiation of the mapping from a vectorized set to the
hypervolume indicator. The Hessian matrix plays a crucial role in second-order methods, such as
the Newton-Raphson optimization method, and it can be used for the verification of local optimal
sets. So far, the full analytical expression was only established and analyzed for the relatively simple
bi-objective case. This paper will derive the full expression for arbitrary dimensions (\( m \geq 2 \) objective
functions). For the practically important three-dimensional case, we also provide an asymptotically
efficient algorithm with time complexity in \( O(n \log n) \) for the exact computation of the Hessian
Matrix’ non-zero entries. We establish a sharp bound of \( 12m - 6 \) for the number of non-zero entries.
Also, for the general \( m \)-dimensional case, a compact recursive analytical expression is established,
and its algorithmic implementation is discussed. Also, for the general case, some sparsity results can
be established; these results are implied by the recursive expression. To validate and illustrate the
analytically derived algorithms and results, we provide a few numerical examples using Python and
Mathematica implementations. Open-source implementations of the algorithms and testing data are
made available as a supplement to this paper.

1 Introduction

In this paper, we delve into continuous \( m \)-dimensional multi-objective optimization problems (MOPs), where multiple
objective functions, e.g., \( \mathbf{f} = (f_1, \ldots, f_m) : \mathcal{X} \subseteq \mathbb{R}^d \to \mathbb{R}^m \) are subject to minimization. Also, we assume \( \mathbf{f} \) is at least
twice continuously differentiable. When solving such problems, it is a common strategy to approximate the Pareto
front for \( m \)-objective functions mapping from a continuous decision space \( \mathbb{R}^d \) to the \( \mathbb{R} \) (or as a vector-valued function

Table 1: Basic notation used throughout the paper. HVI stands for “Hypervolume Indicator”. $B(\mathbb{R}^m)$ denotes the Borel sets on $\mathbb{R}^m$.

| Symbol | Domain or Signature | Description |
|--------|---------------------|-------------|
| $m$    | $\mathbb{N}$        | number of objective functions |
| $d$    | $\mathbb{N}$        | number of decision variables |
| $n$    | $\mathbb{N}$        | number of points in the approximation set |
| $f = (f_1, \ldots, f_m)$ | $\mathbb{R}^d \to \mathbb{R}^m$ | vector-valued objective function |
| $X = (x^{(1)}_1, \ldots, x^{(n)}_n)\top$ | $\mathbb{R}^{nd}$ | concatenation of $n$ points in the decision space |
| $Y = (y^{(1)}_1, \ldots, y^{(n)}_n)\top$ | $\mathbb{R}^{nm}$ | concatenation of $n$ points in the objective space |
| $\lambda_m$ | $B(\mathbb{R}^m)$ $\to \mathbb{R}_{\geq 0}$ | $m$-dimensional Lebesgue measure |
| $HV$ | $B(\mathbb{R}^n)$ $\to \mathbb{R}_{\geq 0}$ | HVI for subsets of the objective space |
| $H$ | $\mathbb{R}^{nm} \to \mathbb{R}_{\geq 0}$ | HVI supported on the product of $n$ objective spaces |
| $H_F$ | $\mathbb{R}^{nd} \to \mathbb{R}_{\geq 0}$ | HVI supported on the product of $n$ decision spaces |

From $\mathbb{R}^d$ to $\mathbb{R}^m$. MOPs can be accomplished by means of a finite set of points that distributes across the at most $m - 1$-dimensional manifold of the Pareto front. The hypervolume indicator of a set of points is the $m$ dimensional Lebesgue measure of the space that is jointly dominated by a set of objective function vectors in $\mathbb{R}^m$ and bound from above by a reference point $r \in \mathbb{R}^m$. More precisely, for minimization problems, the hypervolume indicator (HV) [ZT98, ZTL03] is defined as the Lebesgue measure of the compact set dominated by a Pareto approximation set $Y \subset \mathbb{R}^m$ and cut from above by a reference point $r$:

$$HV(Y; r) = \lambda_m \left\{ p : \exists y \in Y (y \prec p \wedge p \prec r) \right\},$$

where $\lambda_m$ denotes the Lebesgue measure on measurable space $(\mathbb{R}^m, B(\mathbb{R}^m))$ with $B(\mathbb{R}^m)$ being the Borel sets of $\mathbb{R}^m$. We will omit the reference point for simplicity henceforth. HV is Pareto compliant, i.e., for all $Y \prec Y'$, $HV(Y') > HV(Y)$, and is extensively used to assess the quality of approximation sets to the Pareto front, e.g., in SMS-EMOA [BNE07] and multiobjective Bayesian optimization [EYD16]. Being a set function, it is cumbersome to define the derivative of HV\(^1\). Therefore, we follow the generic set-based approach for MOPs [ED12], which considers a finite set of objective points (of size $n$) as a single point in $\mathbb{R}^{nm}$, obtained via the following concatenation map (and its inverse):

$$concat : (\mathbb{R}^m)^n \to \mathbb{R}^{nm}, \quad Y \mapsto \left( y^{(1)}_1, \ldots, y^{(1)}_m, \ldots, y^{(n)}_1, \ldots, y^{(n)}_m \right)\top,$$

$$concat^{-1} : \mathbb{R}^{nm} \to (\mathbb{R}^m)^n, \quad Y \mapsto \left\{ (Y_1, \ldots, Y_m)\top, (Y_{m+1}, \ldots, Y_{2m})\top, \ldots, (Y_{(n-1)m+1}, \ldots, Y_{nm})\top \right\}.$$

The concatenation map gives rise to a hypervolume function that takes vectors in $\mathbb{R}^{nm}$ as input:

$$H : \mathbb{R}^{nm} \to \mathbb{R}_{\geq 0}, \quad Y \mapsto [HV \circ concat^{-1}](Y),$$

Similarly, we also consider a finite set of decision points (of size $n$) as a single point in $\mathbb{R}^{nd}$, i.e., $X = [x^{(1)}\top, x^{(2)}\top, \ldots, x^{(n)}\top]\top \in \mathbb{R}^{nd}$. In this sense, the objective function $f$ is also extended to:

$$F : \mathbb{A}^n \to \mathbb{R}^{nm}, \quad X \mapsto [f(X_1, \ldots, X_d), f(X_{d+1}, \ldots, X_{2d}), \ldots, f(X_{(n-1)d+1}, \ldots, X_{nd})]\top.$$

Given $F$, we have the relation $Y = F(X)$. Now, we can express hypervolume indicator as a function supported on $\mathbb{R}^{nd}$:

$$H_F : \mathbb{R}^{nd} \to \mathbb{R}_{\geq 0}, \quad X \mapsto [HV \circ concat^{-1} \circ F](X).$$

Notably, assuming $f$ is twice differentiable, the above hypervolume functions $H$ and $H_F$ are twice differentiable almost everywhere in their domains, respectively.

It is straightforward to express the gradient of $H_F$ w.r.t. $X$ using the chain rule as reported in our previous works [ED12, WDBE17]: $\nabla H_F(X) = (\partial H / \partial Y)(\partial F(X) / \partial X)$, in which we also discusses the time complexity of computing the hypervolume gradient.

In this work, we investigate the Hypervolume Indicator Hessian Matrix for more than two objectives and propose an algorithm to compute it efficiently. The work is structured as follows: In Section 2, we briefly review the general construction of the Hypervolume Gradient and Hypervolume Hessian $H_F$ via the chain-rule as it has been outlined.

\(^1\)The derivative of a set function is not defined for an arbitrary family of sets. For some special cases, it can be defined directly, e.g., on Jordan-measurable sets [DD02].
As with many problems related to Pareto dominance, the 2-D and 3-D cases have a special structure that can be exploited

Previously in [ED12] and, respectively, in [SSW+ 20]. Section 3 provides a discussion of the 3-D Hypervolume indicator Hessian matrix HV, including a $O(n \log n)$ dimension sweep algorithm for its asymptotically optimal computation and an analysis of its sparsity, i.e., the number of its non-zero components. Furthermore, it is argued that in the 3-D case, the Hypervolume Hessian Matrix is sparse and has at most $O(n)$ non-zero components. In Section 4, we discuss the analytical formulations of the hypervolume Hessian for the general case of $m > 1$ objective functions. The result reduces the computation of the Hessian of $m$ objective functions to the repeated computation of the hypervolume indicator gradient for collections of vectors in $\mathbb{R}^{m-1}$. A Python implementation is provided for the general cases. In Section 5, we provide numerical examples. In Section 6, we finish the paper with a discussion of some basic properties of the hypervolume Hessian matrix, such as its continuity, one-sidedness, and rank, and point out interesting open questions for its further analysis.

2 General Construction of Hypervolume Hessian and Gradient via the Chain Rule

Taking the $\mathbb{R}^{nd}$-vector $X$ and $\mathbb{R}^{nm}$-vector $Y = F(X)$, we express the Hessian matrix of the hypervolume indicator as follows:

$$\nabla^2 H_F = \frac{\partial}{\partial X} \left( \frac{\partial H}{\partial Y} \frac{\partial Y}{\partial X} \right) = \left[ \frac{\partial}{\partial X} \left( \frac{\partial H}{\partial Y} \right) \right]^T + \frac{\partial H}{\partial Y} \frac{\partial^2 F(X)}{\partial X^2}.$$

The Hessian of vector-valued objective function $F$, i.e., $\partial^2 F/\partial X \partial X^T : \mathbb{R}^{nd} \to \text{Hom}(\mathbb{R}^{nd}, \text{Hom}(\mathbb{R}^{nd}, \mathbb{R}^{nm}))$, is a tensor of $(1, 2)$ type. Let $T_{k,i,j}^{\alpha} = \partial^2 F_i/\partial X_i \partial X_j, i, j \in [1..nd], k \in [1..nm]$, we specify the entries of $T$ as follows:

$$T_{k,i,j}^{\alpha} = \begin{cases} \partial^2 f_{\beta}(x^{(\alpha)})/\partial x^{(\alpha)}_i \partial x^{(\alpha)}_j, & \text{if } (\alpha - 1)d + 1 \leq i, j \leq \alpha d, \\ 0, & \text{otherwise}. \end{cases}$$

Since the above map from $k$ to $(\alpha, \beta)$ is bijective (its inverse is $k = \alpha \beta$), we will equivalently use $\alpha \beta$ for the contravariant index $k$. It is obvious that tensor $T$ is sparse, where for each $k$, only $d^2$ entries are nonzero, giving up to $nmd^2$ nonzero entries in total. Using the Einstein summation convention, we can expand the second term in Eq. (3) as:

$$\left( \frac{\partial H}{\partial Y} \frac{\partial^2 F(X)}{\partial X^2} \right)_{i,j} = \left( \frac{\partial H}{\partial Y} \right)^T_{k} T_{k,i,j}^{\alpha \beta} \frac{\partial f_{\beta}(x^{(\alpha)})}{\partial x^{(\alpha)}_i} \frac{\partial f_{\beta}(x^{(\alpha)})}{\partial x^{(\alpha)}_j},$$

where $(\cdot)_{i,j}$ denotes the $(i,j)$-entry of a tensor. We have discussed the analytical expression of the term $\partial H/\partial Y$ in our previous works [ED12, WDBE17]. Notably, the above expression leads to a block-diagonal matrix containing $n$ matrices of shape $d \times d$ on its diagonal. Therefore, we observe a high sparsity of the second term in Eq. (3). As for the first term, $\partial^2 H/\partial Y \partial X^T$ denotes the Hessian of the hypervolume indicator w.r.t. objective vectors, whose computation and sparsity will be discussed in the following sections.

In our previous work [SSW+ 20], we have derived the analytical expression of $\nabla^2 H_F$ for bi-objective cases and analyzed the structure and properties of the hypervolume Hessian matrix. Also, we implemented a standalone Hypervolume Newton (HVN) algorithm for unconstrained MOPs. Moreover, we have shown that the Hessian $\nabla^2 H_F$ is a tri-diagonal block matrix in bi-objective cases and provided the non-singularity condition thereof, which states the Hessian is only singular on a null subset of $\mathbb{R}^{nd}$ [SSW+ 20], thereby ascertaining the safety of utilizing hypervolume Hessian matrix, e.g., in the hypervolume Newton method [WED+ 22] for equality constraints.

3 Hypervolume Indicator Hessian Matrix in 3-D

As with many problems related to Pareto dominance, the 2-D and 3-D cases have a special structure that can be exploited for formulating asymptotically efficient dimension sweep algorithms [KLP75, PSSL22]. Next, the dimension sweep technique will be applied to yield an asymptotically efficient algorithm for the problem of computing the Hessian Matrix of the 3-D Hypervolume Indicator HV. The basic idea is sketched in Fig. 1 and consists of computing first the facets of the polyhedron that is measured by the hypervolume indicator by lowering a sweeping plane along each one of the axes. The gradient components are given by the areas of the facets (e.g., the yellow shaded area in Fig. 1, and the length of the line segments of the ortho-convex polygon that surrounds this facet determines the components of the Hessian matrix of HV. This can be easily verified by studying geometrically the effect of small perturbations of the coordinates of points in $Y$ along the coordinate axis on the value of the hypervolume indicator (gradient components) HV.
Without loss of generality, we first compute the derivatives with respect to \( y_3 \). By permuting the roles of \( y_1 \), \( y_2 \), and \( y_3 \), we can get all derivatives. In the context of the dimension sweep algorithm, we assume that points in Fig. 1 are sorted by the 3rd coordinate \( y_3 \), that is, \( Y \) is represented in such a way that \( y_3^{(3)} < y_3^{(2)} < y_3^{(1)} \). We assume that the points in \( Y \) are in general position (otherwise, one-sided derivatives can occur, which will be discussed later).

Alg. 1 computes all positive entries of the Hessian matrix of HV at a point \( Y \). The algorithm proceeds in three sweeps. The sweep coordinate has index \( h \) (like height), and the other two coordinates are termed \( l \) and \( w \) (like length and width). The first sweep sets \( h = 3 \), the second sweep \( h = 2 \), and the third sweep \( h = 1 \). The values of \( l \) and \( w \) are set to the remaining two coordinates. The roles of \( l \) and \( w \) are interchangeable, but here we set them to \( l = 1 \), \( w = 2 \) in the first sweep, to \( l = 1 \), \( w = 3 \) in the second sweep, and to \( l = 3 \), \( w = 2 \) in the third sweep. Next, we describe a single sweep in detail. Without loss of generality, let us choose the sweep along the 3rd coordinate, e.g., \( l = 1 \), \( w = 2 \), \( h = 3 \):

First, we introduce sentinels \( y^{(0)} \) and \( y^{(n+1)} \) that make sure that every point always has a left and a lower neighbor. We use a balanced binary search tree \( T \) to efficiently maintain a list of all non-dominated points in the \( lw \)-plane among the points that have been visited in a single sweep so far. The sorted list represented by tree \( T \) is initialized by the sentinels; note that the sentinels cannot become dominated in the \( lw \)-plane because one of their coordinates is \(-\infty\). The other coordinates are from the reference point. Next, start the loop that starts from the highest \( y_h \) coordinate and lowers the sweeping plane to the next highest \( y_h \) coordinate in each iteration \( t \). The value of \( t \) is thus the index of the point that is currently processed, and points are sorted in the \( h \)-direction using the index transformation \( a[t], t = 1, \ldots, n \). In each iteration, we determine from the sorted list the sublist starting from \( y^{d[t][0]} \) and terminating with \( y^{d[t][N_t]} \). We assume the list is sorted ascendingly in the \( l \)-coordinate. The point is \( T \) with the highest \( y_h \) coordinate that does not exceed \( y_t^{a[t]} \) is chosen as \( t^{d[t][0]} \) and the point with the highest \( y_w \) coordinate that does not yet exceed \( y_w^{a[t]} \) is chosen as \( t^{d[t][N_t+1]} \). These two points always exist because of the sentinels we set initially. The points between these points in the list, given there exist such points, are referred to by \( y^{d[t][1]}, \ldots, y^{d[t][N_t]} \). If no such points exist, then \( N_t \) is set to 0. Note, that the points \( y^{d[t][1]}, \ldots, y^{d[t][N_t]} \) are points that become dominated by \( y^{a[t]} \) in the \( lw \)-projection. They will be discarded from \( T \) at the end of the iteration, and \( y^{a[t]} \) will be inserted to \( T \) thereafter so that the list represented by \( T \) remains a list of mutually non-dominated points in the \( lw \)-projection. Before discarding the points from \( T \), the new positive components of the Hypervolume Hessian are computed. This is done by computing the line segments of the polygonal area that is marked by the points \( y^{a[t]} \) and \( y^{d[t][0]}, \ldots, y^{d[t][N_t+1]} \) as it is indicated graphically in Fig. 2 for a single iteration of the algorithm. This is the polygonal region that marks the area of the hypervolume gradient \( \partial H / y_h^{a[t]} \). Changing the coordinates of the corners of this polygon infinitesimally adds a differential change.
to the area, which is the aforementioned hypervolume indicator gradient component. The details of the assignment to the Hypervolume Hessian can be determined by computing the side-lengths of the edges of the polygon and carefully tracing which coordinates of points in $Y$ determine the coordinates of the region the area of which determines the gradient component (the yellow area in Fig. 1 and in Fig. 2).

**Theorem 1.** Computation of Hessian Matrix components by HV3D-TRISWEEP algorithm Assume that $n$ mutually non-dominated points in $\mathbb{R}^3$ are given by a collection $Y$, and assume they are in general position (no duplicate coordinates). Furthermore, assume that all points in $Y$ dominate the reference point $r$. Then algorithm 1 that we will term HV3D-TRISWEEP computes all non-zero components of the Hessian matrix $HV$.

**Proof.** The idea of the Alg. 1 is that it computes the polygons that mark the region, the area of which is the hypervolume derivative. It uses the same dimension sweep approach then detailed in [ED12] to compute these polygons, one polygon at each lowering of the sweep-plane. Now, it can be easily derived by geometrical considerations on the changes in the coordinates of corner points differentially change the area of the polygon. The corner points of the polygon are coordinates of points in $Y$, and thus the second partial derivatives will be related to these coordinates. Due to the axis aligned, orthogonal geometry of the polygon, these second derivatives are easy to establish as the length of the edges of the polygon.

Next, we analyze the time complexity of Alg. 1 and study the number of non-zero components it computes, which corresponds exactly to the non-zero components of the Hessian matrix of $HV$. To do so, let us first proof a lemma:

**Lemma 1.** In algorithm 1 it holds that $\sum_n N_i = n - 1$.

**Proof.** $N_i$ is the number of dominated points in the 2-D plane spanned by the $y_l$ and $y_w$ coordinate in one iteration of the algorithm marked by iteration index $t$. Each point in $Y$ gets dominated in the 2-D projection at most once, and in this case, it contributes to $N_i$ and is removed from the tree, with the exception of the last point.
Algorithm 1: HVH3D Triple Dimension Sweep algorithm to Compute all non-zero components of $\partial^2 H / \partial Y \partial Y^\top$

1 Procedure: HVH3D-TriSweep($Y$);
2 Input: a collection of vectors $Y = \{y^{(1)}, \ldots, y^{(n)}\} \subset \mathbb{R}^m$, reference point $r \in \mathbb{R}^m$;
3 Output: Non-zero components of the Hessian matrix of HV: $\partial^2 H / \partial y_m^{(i)} \partial y_k^{(j)}$;
   // Three sweeps along the different coordinate axis.
4 for $(l, w, h) \in ((1, 2, 3), (1, 3, 2), (3, 2, 1))$ do
   5 \hspace{1cm} $y_l^{(0)} = -\infty, y_w^{(0)} = r_w, y_h^{(0)} = r_h$; \hspace{1cm} >> Define the sentinels
   6 \hspace{1cm} $y_l^{(n+1)} = r_l, y_l^{(n+1)} = -\infty, y_l^{(n+1)} = r_l$;
   7 \hspace{1cm} $(y^{(a[1])}, \ldots, y^{(a[n])}) \leftarrow$ Sort $(y^{(1)}, \ldots, y^{(n)})$ descendingly by $y_h$ coordinate;
   8 \hspace{1cm} $T = $ BSTree($y^{(0)}, y^{(n+1)}$); \hspace{1cm} >> Initialize balanced search tree
   9 for $t \in (1, \ldots, n)$ do
   10 \hspace{1cm} Determine $N[t]$ and $(y^{d[t][0]}, \ldots, y^{d[t][N[t]+1]})$ based on $y^{a[t]}$ as sublist of $T$ starting from nearest
   11 \hspace{1cm} \hspace{1cm} lower neighbor to $y^{a[t]}$ in $l$ direction and terminating at nearest lower neighbor of $y^{a[t]}$ in $w$-direction;
   12 \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_l^{(a[t])} = -(y^{d[t][0]} - y^{a[t]});$
   13 \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_w^{(a[t])} = -(y^{d[t][N[t]]} - y^{a[t]});$
   14 \hspace{1cm} if $N[t] \geq 0$ then
   15 \hspace{1cm} \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_{d[t][0]}^{a[t]} = y_{d[t][1]}^{a[t]} - y_{d[t][0]}^{a[t]}$;
   16 \hspace{1cm} \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_{d[t][N[t]]}^{a[t]} = y_{d[t][N[t]-1]}^{a[t]} - y_{d[t][N[t]]}^{a[t]}$;
   17 \hspace{1cm} \hspace{1cm} for $j = 1, \ldots, N_t$ do
   18 \hspace{1cm} \hspace{1cm} \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_{d[t][j]}^{a[t]} = y_{d[t][j-1]}^{a[t]} - y_{d[t][j]}^{a[t]}$;
   19 \hspace{1cm} \hspace{1cm} \hspace{1cm} $\partial^2 H / \partial y_h^{(a[t])} \partial y_{d[t][j]}^{a[t]} = y_{d[t][j+1]}^{a[t]} - y_{d[t][j]}^{a[t]}$;
   20 \hspace{1cm} \hspace{1cm} Discard $(y^{d[t][1]}, \ldots, y^{d[t][N[t]]})$ from $T$;
   21 \hspace{1cm} Add $y^{a[t]}$ to tree $T$;
22 return $\partial^2 H / \partial y_m^{(i)} \partial y_k^{(j)}$; \hspace{1cm} >> return only the $O(n)$ computed elements and their indices

Theorem 2. Time Complexity of 3-D Hessian Matrix of HV The computation of all non-zero components of the Hessian matrix of the mapping from a set $Y \in \mathbb{R}^3$ in the objective space to the hypervolume indicator takes computational time $\Theta(n \log n)$.

Theorem 3. Sparsity of 3-D Hessian Matrix of HV The number of all non-zero components of the Hessian matrix of the mapping from a set $Y \in \mathbb{R}^3$ in the objective space to the hypervolume indicator does never exceed $12n - 6$.

Proof. The algorithm produces in each step $t$ exactly $2N_t + 4$ entries of the Hypervolume Hessian Matrix. Per dominated point, there are 2 non-zero components computed. There are exactly $4$ additional non-zero entries computed in each iteration $t$. Therefore the total computation time amounts to $2N_t$ steps to process the newly dominated points in the 2-D projection, plus the 4 additional components of the polytope boundary in each iteration. By amortization described in Lemma 1 of $N_t$, we obtain $2(n-1) + 4n$ components per sweep, and we carry out 3 sweeps, resulting in at most $12n - 6$ non-zero components. \hfill $\square$

4 General Expression of the N-Dimensional Hypervolume Hessian Matrix

For the general cases ($m > 3$), it suffices to compute the term $\partial^2 H / \partial Y \partial Y^\top$ and utilize Eq. (3) for computing the hypervolume Hessian. We summarize the computation in Alg. 2 and explain the details as follows. Let $A = \partial^2 H / \partial Y \partial Y^\top \in \mathbb{R}^{nm \times nm}$. Without loss of generality, we calculate the entries of $A$ in a column-wise manner given indices $i \in [1..n]$ and $k \in [1..n]$, column $ik$ of $A$ takes the following form:

\[
\begin{pmatrix}
\frac{\partial}{\partial y^{(1)}} \left( \frac{\partial H}{\partial y_k^{(i)}} \right)^\top \\
\frac{\partial}{\partial y^{(2)}} \left( \frac{\partial H}{\partial y_k^{(i)}} \right)^\top \\
\vdots \\
\frac{\partial}{\partial y^{(n)}} \left( \frac{\partial H}{\partial y_k^{(i)}} \right)^\top \\
\end{pmatrix}
\]
Algorithm 2: General algorithm for the hypervolume Hessian matrix

1. **Input:** $X = \{x^{(1)}, \ldots, x^{(n)}\}$: decision points, $Y = \{y^{(1)}, \ldots, y^{(n)}\}$: objective points, $\nabla F, \nabla^2 F$: Jacobian and Hessian of the objective function;
2. **Output:** $H$: hypervolume Hessian matrix;
3. $X$ ← concat($X$);
4. $A$ ← $0_{nm \times nm}$;
5. for $i = 1, \ldots, n$ do
   6. if $x^{(i)}$ is dominated then continue;
   7. for $k = 1, \ldots, m$ do
      8. $y_{-k}^{(i)}$ ← proj_k($y^{(i)}$);
      9. $Y'$ ← \{proj_k($y$): $y \in Y \land y_k < y_k^{(i)}$\};
      10. $I(Y, i, k)$ ← \{$\alpha \in [1..n]$: $y_k^{(\alpha)} < y_k^{(i)}$\};
    // compute $\partial (\partial H / \partial y_k^{(i)}) / \partial y^{(i)}$
    11. for $l = 1, \ldots, m$ do
       12. if $l = k$ then $v_l$ ← 0; continue;
       13. $p$ ← $l$ if $l < k$; otherwise $p$ ← $l - 1$;
       14. $v_l$ ← HVC \{proj_p($y_{-k}^{(j)}$), \{proj_p($y_{-k}^{(\alpha)}$): $\alpha \in I(Y', i, l)$\}\};
       \> $v_l := \partial^2 H / \partial y_l^{(i)} \partial y_k^{(i)}$; Eq. (7)
    15. for $s = im + 1, \ldots, (i + 1)m$ do
       16. $l$ ← $s - im$;
       17. $A_{s,im+k}$ ← $v_l$;
    // compute $\partial (\partial H / \partial y_k^{(i)}) / \partial y^{(j)}$
    18. $\tilde{Y}$ ← \{CLIP($y$; $y_{-k}^{(j)}$): $y \in Y'$\};
    19. for $j \in I(Y, i, k)$ do
       20. $\tilde{y}_{-k}^{(j)}$ ← $\tilde{Y}[j]$;
       \> take element $j$ from set $Y'$
       21. for $l = 1, \ldots, m$ do
          22. if $l = k$ then $w_l$ ← 0; continue;
          23. $p$ ← $l$ if $l < k$; otherwise $p$ ← $l - 1$;
          24. $w_l$ ← HVC \{proj_p($\tilde{y}_{-k}^{(j)}$), \{proj_p($\tilde{y}_{-k}^{(\alpha)}$): $\alpha \in I(\tilde{Y}, j, p)$\}\};
          \> $w_l := \partial^2 H / \partial y_l^{(i)} \partial y_k^{(i)}$
       25. for $s = jm + 1, \ldots, (j + 1)m$ do
          26. $l$ ← $s - jm$;
          27. $A_{s,jm+k}$ ← $w_l$;

28. $T$ ← $\nabla^2 F(X)$;
29. $H$ ← $\nabla F(X)^\top A \nabla F(X) + \sum_{\alpha=1}^{n} \sum_{\beta=1}^{m} (\partial H / \partial f_\beta(x^{(\alpha)})) T^{\alpha\beta}$;
\> Eq. (3)

which will be discussed in two scenarios: (1) $\partial(\partial H / \partial y_k^{(i)}) / \partial y^{(i)}$ and (2) $\partial(\partial H / \partial y_k^{(i)}) / \partial y^{(j)}$, $i \neq j$. The dominated space of $S \subseteq \mathbb{R}^m$, that is the subset of $\mathbb{R}^m$ dominated by the points in $S$, is denoted by $\text{Dom}_m(S) = \{p \in \mathbb{R}^m: \exists y \in S(y \prec p) \land p \prec r\}$. Also, we take the canonical basis $\{e_i\}$ of Euclidean spaces in our elaboration.

### 4.1 Partial derivative $\partial(\partial H / \partial y_k^{(i)}) / \partial y^{(i)}$

Intuitively, from Fig. 1, we observe that $\partial^2 H / \partial y_k^{(i)} \partial y_k^{(i)}$ is always zero for the 3D case since $\partial H / \partial y_k^{(i)}$ is essentially the hypervolume improvement of the projection of $y^{(i)}$ along axis $e_k$ (bright yellow area in Fig. 1), ignoring the points that dominates $y^{(i)}$ after the projection. In addition, $\partial^2 H / \partial y_k^{(i)} \partial y_k^{(i)}$, $i \neq k$ equals the negation of partial derivative of the hypervolume indicator w.r.t. $y_k^{(i)}$ in the $m - 1$-dimensional space, resulted from the projection. The computation of this quantity has been investigated in great detail previously [ED12]. We prove this argument for $m > 3$ as follows. First, we define the orthogonal projection operator onto $e_k$:

$$\text{proj}_k: y \mapsto (\ldots, y_{k-1}, y_{k+1}, \ldots)^\top,$$
which drops the \( k \)-th components of the input point \( \mathbf{y} \). Here, we do not specify the dimensionality of \( \mathbf{y} \) on purpose, since later in the discussion, \( \text{proj}_k \) will be applied to points in \( \mathbb{R}^m \) or \( \mathbb{R}^{m-1} \).

**Theorem 4** (Partial derivative of \( \mathcal{H} \)). Assume a finite approximation set \( Y = \{ \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)} \} \subset (\mathbb{R}^m)^n \). We define \( \mathbf{y}^{(\alpha)}_k = \text{proj}_k(\mathbf{y}^{(\alpha)}) \), \( \alpha \in [1 \ldots n] \) and the index set \( I(Y, i, k) = \{ \alpha \in [1 \ldots n] : y^{(\alpha)}_k < y^{(i)}_k \} \) that selects the points in \( Y \) whose \( k \)-th component is strictly better/smaller than \( y^{(i)}_k \). The partial derivative of \( \mathcal{H} \) w.r.t. \( y^{(i)}_k \) for all \( i \in [1 \ldots n] \) and \( k \in [1 \ldots m] \) admits the following expression:

\[
\frac{\partial \mathcal{H}(\mathbf{Y})}{\partial y^{(i)}_k} = -\text{HVC}\left(\mathbf{y}^{(i)}_k, \{ \mathbf{y}^{(\alpha)}_k : \alpha \in I(Y, i, k) \}\right),
\]

where \( \mathbf{Y} = \text{concat}(Y) \) and \( \text{HVC}(\mathbf{y}, P) \) is the hypervolume contribution of point \( \mathbf{y} \) to a finite point set \( P \), i.e.,

\[
\text{HVC}(\mathbf{y}, P) = \text{HV}(P \cup \{ \mathbf{y} \}) - \text{HV}(P) = \lambda_{m-1}(\text{Dom}_{m-1}(\mathbf{y}) \setminus \cup_{\mathbf{p} \in P} \text{Dom}_{m-1}(\mathbf{p})).
\]

**Proof.** Without loss of generality, we can assume the points in \( Y \) are indexed in the ascending order w.r.t. the \( k \)-component, i.e., \( y^{(1)}_k < \cdots y^{(i-1)}_k < y^{(i)}_k < y^{(i+1)}_k < \cdots < y^{(n)}_k \). Then, we have \( I(Y, i, k) = [1 \ldots i - 1] \). Assume a small perturbation \( \delta \) that satisfies \( 0 < |\delta| < \min\{ y^{(i)}_k - y^{(i-1)}_k, y^{(i+1)}_k - y^{(i)}_k \} \). After adding \( \delta \) to \( y^{(i)}_k \), we shall denote the resulting range of \( y^{(i)}_k \) by \( R(\delta) = [y^{(i)}_k + \delta, y^{(i)}_k] \) if \( \delta > 0 \), or \( R(\delta) = [y^{(i)}_k - \delta, y^{(i)}_k] \) if \( \delta < 0 \). Then, the dominated space of \( \mathbf{y}^{(i)} \) will expand (\( \delta < 0 \)) or shrink (\( \delta > 0 \)) by the follow set:

\[
S(\delta) = R(\delta) \times \text{Dom}_{m-1}(y^{(i)}_k),
\]

where \( \times \) denotes the Cartesian product. Due to the assumption on \( \delta \), we observe that (1) \( S(\delta) \) is disjoint from \( \text{Dom}_m(\mathbf{y}^{(\alpha)}) \) for \( \alpha \in [i + 1 \ldots n] \) and (2) for \( \alpha \in [1 \ldots i - 1] \), \( \text{Dom}_m(\mathbf{y}^{(\alpha)}) \neq \emptyset \); However, \( \text{Dom}_m(\mathbf{y}^{(\alpha)}) \) will not be affected by this small perturbation on \( y^{(i)}_k \). Hence, the change of HV can only be attributed to \( S(\delta) \), which is:

\[
\Delta \text{HV}(\delta) = -\text{sgn}(\delta) \lambda_m \left( S(\delta) \setminus \left( \bigcup_{\alpha=1}^{\alpha=n} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \right) \right)
= -\text{sgn}(\delta) \lambda_m \left( S(\delta) \setminus \left( \bigcup_{\alpha=1}^{\alpha=n} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \right) \right)
= -\text{sgn}(\delta) \lambda_m \left( S(\delta) \setminus \left( \bigcup_{\alpha \in I} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \right) \right),
\]

(5)

where \( \text{sgn} \) is the sign function. Noticing that \( \forall \mathbf{p} \in \cup_{\alpha \in I} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \), \( (p_k \not\in R(\delta) \iff p \not\in S(\delta)) \), meaning the only subset of \( \mathbb{R}^m \) that intersects \( S(\delta) \) is \( U = \{ p \in \cup_{\alpha \in I} \text{Dom}_m(\mathbf{y}^{(\alpha)}) : p_k \in R(\delta) \} \). Due to the construction of \( U \), it admits the following expression:

\[
U = R(\delta) \times \bigcup_{\alpha \in I} \text{Dom}_{m-1}(\text{proj}_k(\mathbf{y}^{(\alpha)})).
\]

Thereby, we can continue simplifying Eq. (5) using \( U \):

\[
\Delta \text{HV}(\delta) = -\text{sgn}(\delta) \lambda_m \left( S(\delta) \setminus U \right)
= -\text{sgn}(\delta) \lambda_m \left( R(\delta) \times \left( \bigcup_{\alpha \in I} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \right) \right)
= -\text{sgn}(\delta) \lambda_1 \left( R(\delta) \right) \lambda_{m-1} \left( \left( \bigcup_{\alpha \in I} \text{Dom}_m(\mathbf{y}^{(\alpha)}) \right) \right)
= -\delta \lambda_{m-1} \left( \text{Dom}_{m-1}(\mathbf{y}^{(i)}_k) \setminus \left( \bigcup_{\alpha \in [1 \ldots i - 1]} \text{Dom}_{m-1}(\mathbf{y}^{(\alpha)}_k) \right) \right)
= -\delta \text{HVC}(\mathbf{y}^{(i)}_k, \{ \mathbf{y}^{(\alpha)}_k : \alpha \in [1 \ldots i - 1] \}).
\]

Finally, we have:

\[
\frac{\partial \mathcal{H}}{\partial y^{(i)}_k} = \lim_{\delta \to 0} \frac{\Delta \text{HV}(\delta)}{\delta} = -\text{HVC}(\mathbf{y}^{(i)}_k, \{ \mathbf{y}^{(\alpha)}_k : \alpha \in [1 \ldots i - 1] \}) = -\text{HVC}(\mathbf{y}^{(i)}_k, \{ \mathbf{y}^{(\alpha)}_k : \alpha \in I(Y, i, k) \}) \]

\[\Box\]

**Corollary 4.1.** It follows immediately from Thm. 4 that

\[
\frac{\partial^2 \mathcal{H}}{\partial y^{(i)}_k \partial y^{(i)}_k} = 0, \quad i \in [1 \ldots n], k \in [1 \ldots m], m \in \mathbb{N}_{>0}.
\]

(6)
For computing $\partial^2 \mathcal{H} / \partial y^{(i)}_i \partial y^{(i)}_k$ ($l \neq k$), we recursively apply Thm. 4 to the point set

$$Y' := \{ y^{(i)}_k : \alpha \in I(Y, i, k) \}.$$ 

Also, noticing that $y^{(i)}_l$ is the $l$-th component of $y^{(i)}_k$ if $l < k$, and is the $l - 1$-th component otherwise, we define $p = l$ if $l < k$; otherwise $p = l - 1$. Taking the index set $I(Y', i, l) = \{ \alpha \in [1..|Y'|] : y^{(i)}_l < y^{(i)}_l \}$, we have:

$$\frac{\partial^2 \mathcal{H}}{\partial y^{(i)}_i \partial y^{(i)}_k} = \mathrm{HVC} \left( \{ \text{proj}_p(y^{(i)}_k), \{ \text{proj}_p(y^{(i)}_l) : \alpha \in I(Y', i, l) \} \} \right),$$

which is $m - 2$-dimensional Lebesgue measure. This recursive computation is employed by line 14 of Alg. 2.

**Proof.** Eq. (6) is obvious since $\partial \mathcal{H} / \partial y^{(i)}_k$ is not a function of $y^{(i)}_k$ (due to the $\text{proj}_k$ operation in Thm. 4). For Eq. (7), we start from the result of Thm. 4, which is $\partial \mathcal{H} / \partial y^{(i)}_k = -\mathrm{HVC}(y^{(i)}_k, \{ y^{(i)}_k : \alpha \in I(Y, i, k) \})$. Note that when the perturbation of $y^{(i)}_l$ ($l \neq k$) is sufficiently small, the resulting change of the hypervolume contribution of $y^{(i)}_k$ to $\{ y^{(i)}_k : \alpha \in I(Y, i, k) \}$ is exactly the same as the change of HV of the set $Y' := \{ y^{(i)}_k \} \cup \{ y^{(i)}_k : \alpha \in I(Y, i, k) \}$. Noticing that $y^{(i)}_l$ is the $l$-th component of the projected point $y^{(i)}_l$ if $l < k$ and is the $l - 1$-th component otherwise, we define $p = l$ if $l < k$; otherwise $p = l - 1$. Hence, when applying Thm. 4 again on $Y'$, we have to project all points in $Y'$ onto $e_p^+$. Now, we can simplify $\partial (\partial \mathcal{H} / \partial y^{(i)}_k) / \partial y^{(i)}_l, l \neq i$ as follows:

$$\frac{\partial^2 \mathcal{H}}{\partial y^{(i)}_i \partial y^{(i)}_l} = -\frac{\partial}{\partial y^{(i)}_i} \left( \mathrm{HVC} \left( y^{(i)}_i, \{ y^{(i)}_k : \alpha \in I(Y, i, k) \} \right) \right)$$

$$= -\frac{\partial}{\partial y^{(i)}_i} \mathrm{H} (\text{concat} (Y'))$$

$$= \mathrm{HVC} \left( \{ \text{proj}_p(y^{(i)}_i), \{ \text{proj}_p(y^{(i)}_l) : \alpha \in I(Y', i, l) \} \} \right).$$

The last step above is obtained by applying Thm. 4 again on the point set $Y'$, which involves projecting the objective points in $\mathbb{R}^{m-1}$ onto $e_p^+$. 

In all, we have elaborated the method to compute $\partial (\partial \mathcal{H} / \partial y^{(i)}_k) / \partial y^{(i)}_l, l \neq i$ which constitutes $d$ entries of column $i k$ in matrix $A$. We proceed to the computation of the remaining entries in the next sub-section.

### 4.2 Partial derivative $\partial \left( \partial \mathcal{H} / \partial y^{(i)}_k \right) / \partial y^{(i)}_l, i \neq j$

Based on Thm. 4, we conclude that $\partial (\partial \mathcal{H} / \partial y^{(i)}_k) / \partial y^{(i)}_l = 0 \iff j \notin I(Y, i, k)$, since $\partial \mathcal{H} / \partial y^{(i)}_k$ only depends on $y^{(i)}_k$ and $\{ y^{(i)}_k : \alpha \in I(Y, i, k) \}$. Also, for all $j \in I(Y, i, k)$, we have

$$\partial^2 \mathcal{H} / \partial y^{(i)}_k \partial y^{(i)}_j = 0,$$

due to the projection operation. Note that, $\partial (\partial \mathcal{H} / \partial y^{(i)}_k) / \partial y^{(i)}_j$ constitutes the remaining $(n-1)$ entries of column $ik$ of matrix $A$, containing at most $(i-1)(d-1)$ nonzero values, where $i$ depends on the number of points which have a smaller $k$th-component than that of $y^{(i)}_k$. Hence, we can bound the nonzero elements in $A$ by $O(n(n-1)/2(d-1)) = O(n^2d)$. Note that possible a sharper bound can be formulated by considering the technique used to prove Thm. 3 in more than three dimensions.

The remaining partial derivatives can be computed by first clipping points in $\{ y^{(i)}_k : \alpha \in I(Y, i, k) \}$ by $y^{(i)}_k$ from below (line 13 in Alg. 2; sub-procedure $\text{CLIP}$): $\tilde{y}^{(i)}_k = \text{CLIP}(y^{(i)}_k; y^{(i)}_k)$, where for $a, b \in \mathbb{R}^{m-1}$,

$$\text{CLIP}(a; b) = (a_1 + \min\{0, a_1 - b_1\}, \ldots, a_{m-1} + \min\{0, a_{m-1} - b_{m-1}\})^\top.$$ (9)

Note that, (1) the clipping operation restricts points $y^{(i)}_k, \alpha \in I(Y, i, k)$ in $\text{Dom}_{m-1}(y^{(i)}_k)$, which is crucial to the following steps since otherwise, the infinitesimal perturbation of $y^{(i)}_k$ will not change the hypervolume contribution of
After taking the clipping operation, we have the following result:

$$y_{(\alpha)}^{(i)}$$; (2) this clipping operation does not change the Lebesgue measure of $\text{Dom}_{m-1}(y_{(i)}^{(i)}) \setminus \left( \cup_{\alpha \in I} \text{Dom}_{m-1}(y_{(\alpha)}^{(i)}) \right)$. After taking the clipping operation, we have the following result:

$$
\begin{align*}
\text{HVC} \left( y_{(i)}^{(i)}, \{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\} \right) \\
= \lambda_{m-1} \left( \text{Dom}_{m-1}(y_{(i)}^{(i)}) \setminus \left( \cup_{\alpha \in I} \text{Dom}_{m-1}(y_{(\alpha)}^{(i)}) \right) \right) \\
= \lambda_{m-1} \left( \text{Dom}_{m-1}(y_{(i)}^{(i)}) \setminus \text{Dom}_{m-1}(\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}) \right) \\
= \lambda_{m-1} \left( \text{Dom}_{m-1}(y_{(i)}^{(i)}) \right) - \lambda_{m-1} \left( \text{Dom}_{m-1}(\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}) \right).
\end{align*}
$$

The last step in the above equation is due to the fact that after clipping, $\text{Dom}_{m-1}(\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}) \subset \text{Dom}_{m-1}(y_{(i)}^{(i)})$. As with corollary 4.1, we define $p = l$, if $l < k$; $p = l - 1$, if $l > k$. For $j \neq i, l \neq k$, we have:

$$
\begin{align*}
\frac{\partial^2 \mathcal{H}}{\partial y_{(j)}^{(i)} \partial y_{k}^{(i)}} \\
= -\frac{\partial}{\partial y_{(j)}^{(i)}} \left( \text{HVC} \left( y_{(i)}^{(i)}, \{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\} \right) \right) \\
= -\frac{\partial}{\partial y_{(j)}^{(i)}} \left[ \lambda_{m-1} \left( \text{Dom}_{m-1}(y_{(i)}^{(i)}) \right) - \lambda_{m-1} \left( \text{Dom}_{m-1}(\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}) \right) \right] \quad \text{apply Eq. (10)} \\
= \frac{\partial}{\partial y_{(j)}^{(i)}} \lambda_{m-1} \left( \text{Dom}_{m-1}(\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}) \right) \\
= \frac{\partial}{\partial y_{(j)}^{(i)}} \mathcal{H} \left( \text{concat} \left( \{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\} \right) \right).
\end{align*}
$$

Based on the above result, we can apply Thm. 4 on the set $\{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\} \subset \mathbb{R}^{m-1}$ for computing the second-order derivatives, leading to the following corollary.

**Corollary 4.2.** Let $p = l$, if $l < k$; $p = l - 1$, if $l > k$ and $\tilde{Y} = \{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\}$, where $y_{(\alpha)}^{(i)} = \text{CLIP}(y_{(\alpha)}^{(i)}; y_{(i)}^{(i)})$ and $y_{(\alpha)}^{(i)} = \text{proj}_k(y_{(\alpha)}^{(i)})$. Also, we define the index set $I(\tilde{Y}, j, p) = \{\alpha \in [1..|\tilde{Y}|] : \tilde{y}_{(\alpha)}^{(i)} < \tilde{y}_{(j)}^{(i)} \}$ (where $\tilde{y}_{(\alpha)}^{(i)}$ is the $p$-th component of $y_{(\alpha)}^{(i)}$). For $j \neq i, l \neq k$, the partial derivative $\partial(\partial \mathcal{H}/\partial y_{k}^{(i)})/\partial y_{(j)}^{(i)}$ admits the following expression:

$$
\frac{\partial^2 \mathcal{H}}{\partial y_{(j)}^{(i)} \partial y_{k}^{(i)}} = -\text{HVC} \left( \text{proj}_p(y_{(\alpha)}^{(i)}; \{y_{(\alpha)}^{(i)} : \alpha \in I(\tilde{Y}, j, p)\}) \right),
$$

which is $m - 2$-dimensional Lebesgue measure. This recursive computation is employed by line 24 of Alg. 1.

**Proof.** Considering Eq. (11), the second-order derivatives of interest can be obtained by apply Thm. 4 on the set $\tilde{Y} = \{y_{(\alpha)}^{(i)} : \alpha \in I(Y, i, k)\} \subset \mathbb{R}^{m-1}$. □

In summary, with Eqs. (6) (7) (8) (12), we have specified the computation of all entries of column $ik$ of matrix $A$. Iterating over all the columns will compute the full matrix for our needs. To calculate the hypervolume Hessian matrix, it suffices to substitute term $\partial^2 \mathcal{H}/\partial y_{(i)}^{(i)} \partial \tilde{Y} \partial \tilde{Y}^{T}$ in Eq. (3) with $A$ and evaluate the equation (lines 20 - 23 in Alg. 1).

**5 Numerical Examples**

In this section, we showcase some numerical examples of the computation of the hypervolume Hessian matrix. For the sake of comprehensibility, we only compute the Hessian w.r.t. the objective points since the Hessian matrix w.r.t. the decision points can be easily obtained using the former (see Eq. (3)). We specify the objective points for the numerical problem below.

- **Example 1:** $m = 3, n = 2$, $Y = [(5, 3, 7)^{T}, (2, 1, 10)^{T}]^{T}$, and $r = (0, 10, 12)^{T}$.
- **Example 2:** $m = 3, n = 3$, $Y = [(8, 7, 10)^{T}, (4, 11, 17)^{T}, (2, 9, 21)^{T}]^{T}$, and $r = (10, 13, 23)^{T}$.
We illustrate the corresponding Hessian matrices as heatmaps in Fig. 3, from which we see clearly a high sparsity in all obtained from Python, Mathematica, and automatic differentiation performed in our previous work [WED+22].
6 Discussion and Outlook

This paper highlights two approaches for computing the components of the Hessian matrix of the hypervolume indicator $HV$ of a multi-set of points in objective space and of a multi-set of points in the decision space $HF$. The approach of set-scalarization, as originated in [EDB07, ED12] for the gradient of the hypervolume indicator. The main results of the paper are as follows:

1. the hypervolume indicator of $HF$ can now be computed analytically not only for the bi-objective case as in [SSW+20], but also for more than two objective functions (Theorem 1).
2. the time complexity of computing all non-zero components of the 3-D hypervolume indicator $HV$ for vectorized sets $Y$ with $n$ points in general position is in $\Theta(n \log n)$. (Theorem 2)
3. The number of non-zero components of the Hessian matrix is at most $12n - 6$. The space complexity of the Hessian matrix computation and the space required to store all components is in $O(n)$. (Theorem 3).
4. it holds that $\partial HV/\partial y_k$ is always the hypervolume contribution of the projection of $y_k$ along axis $k$. (Theorem 4).
5. the analytical computation of the higher derivatives of the $m$-dimensional hypervolume indicator $HV$ for $m > 1$can be formulated by computing the gradient of the gradient, which can be essentially achieved by the recursive application of Theorem (Theorem 4) (computing the $m-2$-dimensional projection’s contributions along the $y_k$ axis, of the $m-1$ dimensional projections’ hypervolume contributions along the $y_k$); and taking special care of the role of the reference points and signs of non-zero components as detailed in Alg. 2.

Some interesting next steps would be to

1. investigate the rank of the Hypervolume Hessian matrix and its numerical stability of second-order methods that use the Hessian matrix of $HF$ (or its inverse) in their iteration, such as the Hypervolume Newton Method [SHSE14, SSW+20].
2. find (asymptotically) efficient algorithms for the computation of the higher-order derivative tensors and for more than three-dimensional cases. The latter might, however, find the asymptotical time complexity of the N-D Hessian matrix computation might turn out to be a difficult endeavor, as it is not even known what the asymptotical time complexity of $HV$ is. What is more promising is to bound the number of the non-zero components in the Hessian matrix, which is related to the number of $n-2$ dimensional facets in the ortho-convex polyhedron that marks the measured region of $HV$. It is conjectured that in the $m$-dimensional case, it also grows linearly in $n$ (the number of points in the approximation set) but exponentially in $m$ (the number of objectives). However, dimension sweep algorithms also yield high efficiency in the 4-D case and can probably be easily adapted [GFE+12].

More generally, it is remarked that besides the hypervolume indicator also, other measures have been proposed for the quality of Pareto front approximations, such as the inverted generational distance [IMN15] or the averaged Hausdorff distance [SDMCC+16, UBV+20], with sometimes advantageous properties regarding the uniformity the point distributions in their maximum. Also, for those, the approach of vectorization of the input set is promising, and the analytical computation and subsequent analysis of the Hessian matrix can be an interesting approach. The code for computing analytically the Hessian matrix of $HV$ has been validated on example data and made available in a GitHub repository. The repository includes an implementation based on Alg. 2 in Python and in Mathematica, as well as the data of the examples.

Remark: The authors have been listed alphabetically in this paper, and all authors have contributed to the completion of the manuscript.

Author contributions: HW + ME + AD: Concept and formulation of the general analytical expression for the matrix, ME: 3-D Dimension sweep algorithm and 3-D Complexity Analysis; AD+HW: Improved mathematical notation and numerical validation experiments; HW: theoretical analysis of the general N-dimensional Hessian matrix; All authors: General set-vectorization concepts, motivation, revision/editing of formulation of the paper.

References

[BNE07] Nicola Beume, Boris Naujoks, and Michael T. M. Emmerich. SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur. J. Oper. Res., 181(3):1653–1669, 2007.
[DD02] Emmanuele DiBenedetto and Emmanuele Debenedetto. Real analysis. Springer, 2002.

2https://github.com/wangronin/HypervolumeDerivatives
