Mathematical and Computational Applications

Article

The Hypervolume Newton Method for Constrained Multi-Objective Optimization Problems

Hao Wang [1,*], Michael Emmerich [1], André Deutz [1], Víctor Adrián Sosa Hernández [2] and Oliver Schütze [3]

[1] Leiden Institute of Advanced Computer Science, Leiden University, 2333 CA Leiden, The Netherlands
[2] School of Engineering and Sciences, Tecnológico de Monterrey, Av. Lago de Guadalupe Km 3.5, Atizapán de Zaragoza, Mexico City 52926, Mexico
[3] Computer Science Department, Cinvestav-IPN, Mexico City 07360, Mexico
[*] Correspondence: h.wang@liacs.leidenuniv.nl

Abstract: Recently, the Hypervolume Newton Method (HVN) has been proposed as a fast and precise indicator-based method for solving unconstrained bi-objective optimization problems with objective functions. The HVN is defined on the space of (vectorized) fixed cardinality sets of decision space vectors for a given multi-objective optimization problem (MOP) and seeks to maximize the hypervolume indicator adopting the Newton–Raphson method for deterministic numerical optimization. To extend its scope to non-convex optimization problems, the HVN method was hybridized with a multi-objective evolutionary algorithm (MOEA), which resulted in a competitive solver for continuous unconstrained bi-objective optimization problems. In this paper, we extend the HVN to constrained MOPs with in principle any number of objectives. Similar to the original variant, the first- and second-order derivatives of the involved functions have to be given either analytically or numerically. We demonstrate the applicability of the extended HVN on a set of challenging benchmark problems and show that the new method can be readily applied to solve equality constraints with high precision and to some extent also inequalities. We finally use HVN as a local search engine within an MOEA and show the benefit of this hybrid method on several benchmark problems.

Keywords: multi-objective optimization; hypervolume indicator; newton method; evolutionary algorithms; constraint handling; hypervolume scalarization

check for updates

Citation: Wang, H.; Emmerich, M.; Deutz, A.; Hernández, V.A.S.; Schütze, O. The Hypervolume Newton Method for Constrained Multi-Objective Optimization Problems. *Math. Comput. Appl.* **2023**, *28*, 10. https://doi.org/10.3390/mca28010010

Academic Editor: Efrén Mezura-Montes

Received: 1 November 2022
Revised: 24 December 2022
Accepted: 4 January 2023
Published: 9 January 2023

1. Introduction

Multi-objective optimization problems (MOPs)—i.e., problems where several objectives have to be optimized concurrently –naturally arise in many applications (e.g., [1–4]). As an example, in many portfolio problems, one is interested in maximizing the expected return and social responsibility or sustainability while minimizing the risk to a financial portfolio ([5,6]). In multi-objective optimization, we distinguish between the decision space, which contains the vectors of decision variables, and the objective space, which is the k dimensional real vectors and comprises the images of the vector-valued objective function. A typical approach to the solution of MOPs is to compute or approximate the non-dominated (or efficient) set with respect to the Pareto dominance order (the image of which in the objective space is called the Pareto front). One important characteristic of (continuous) MOPs is that in regular cases, the Pareto front is a manifold of $k-1$ dimensions, where k denotes the number of objective functions. In general, it is possible that parts of the Pareto front are of lower dimension, but the Pareto front is never more than $k-1$ dimensions. Since, in the continuous case, the non-dominated set and the Pareto front can contain infinitely many points, it is usually approximated by a finite set of points. In particular, in the area of evolutionary multi-objective optimization (EMO), many performance indicators have been proposed that propagate optimal approximations of the Pareto front (e.g., [7–10]). While their definitions slightly differ, most have in mind to obtain (more or less) evenly spread solutions along the Pareto front.

Interestingly, with the *hypervolume indicator* [10], there exists an indicator that does not require the knowledge of the location of the true Pareto front. Still, its maximization leads to well-distributed approximation sets consisting of only non-dominated solutions. In this work, by "well-distributed", we mean the objective points have good coverage of the Pareto front and are gap-free when the population size is large. At the maximum of the hypervolume indicator, the density of objective points is inversely proportional to the local curvature of the Pareto front [11]. Here is where the idea of set-scalarization comes into play. In set-scalarization methods, rather than focusing on the improvement of single points of the approximation set, the focus is on the optimization of a fixed cardinality set as an entity concerning a set-based indicator, e.g., the hypervolume indicator. The objective function of the set-scalarization method, in our case, the hypervolume indicator, provides a mapping from the set of fixed cardinality sets in the decision space to a scalar that has to be maximized. Due to the properties mentioned above of the hypervolume indicator, the resulting set will provide a well-distributed set of points on the Pareto front.

Multi-objective evolutionary algorithms (MOEAs) have long since adopted the idea of set-scalarization. The so-called indicator-based MOEAs (e.g., [12–14]) use performance indicators to guide the search, e.g., by indicator-based selection. In numerical methods, the set scalarization approach was first addressed in gradient-based hypervolume maximization [15–19] and in the maximization of the Averaged Hausdorff Metric [14]). More recently, the approach was generalized to second-order methods with the Hypervolume Newton Method (HVN), a set scalarization-based Newton–Raphson method for the maximization of the hypervolume indicator value of a given MOP (e.g., [20,21]). However, this method has only been discussed for unconstrained and bi-objective optimization problems, which limits its application.

In this paper, we extend the HVN for constrained MOPs with a general number of objectives. To this end, we present the HVN for equality-constrained problems and further discuss a straightforward active set method to handle inequalities. Since the HVN is highly local, we also discuss the hybridization of this method with an MOEA. Finally, we present numerical results indicating the strength of the novel approaches.

The remainder of this paper is organized as follows: in Section 2, we present the necessary background required for understanding the sequel, and we review the related work. In Section 3, we present the HVN for constrained multi-objective optimization problems. Section 4 presents the numerical results of the constrained HVN as a standalone algorithm and a local search strategy within a hybrid evolutionary algorithm. Finally, we conclude and give possible paths for future research in Section 5.

2. Background and Related Work

2.1. Notations

We will always denote a finite Pareto approximate set by $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(\mu)}\} \subseteq \mathbb{R}^n$. When differentiating a set function, e.g., the hypervolume, over the input set, we often concatenate the points in $\mathbf{X}$ into a much longer vector, i.e., $\mathbf{X} = [\mathbf{x}^{(1)^\top}, \mathbf{x}^{(2)^\top}, \ldots, \mathbf{x}^{(\mu)^\top}]^\top \in \mathbb{R}^{\mu n}$. To make our discussion less cumbersome, we abuse the notation $\mathbf{X}$ slightly such that it can be interpreted as a finite set in $\mathbb{R}^n$ or an $\mathbb{R}^{\mu n}$-vector, depending on the context. (See [16] for a detailed formal discussion of the mapping between fixed cardinality sets and vectors.) We will explain the meaning of $\mathbf{X}$ on the spot whenever it is unclear from the text. We will always denote by ∇ and ∇^2 the gradient/Jacobian and Hessian operators on real-valued functions, respectively, when the domain of such a function is clear from the text. Otherwise, we take the derivative operator $\partial/\partial\mathbf{X}$. When expressing the Hessian matrix, we will use the numerator layout for matrix calculus notations [22].

2.2. Multi-Objective Optimization

A real-valued multi-objective optimization problem (MOP) involves minimizing multiple objective functions simultaneously, i.e., $F = (f_1, \ldots, f_k), f_i : \mathcal{X} \to \mathbb{R}, \mathcal{X} \subseteq \mathbb{R}^n, i \in \{1, \ldots, k\}$. For every $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)} \in \mathbb{R}^k$, we say $\mathbf{y}^{(1)}$ weakly dominates $\mathbf{y}^{(2)}$ (written as

$\mathbf{y}^{(1)} \preceq \mathbf{y}^{(2)}$) iff $y_i^{(1)} \leq y_i^{(2)}$, $i \in [1 \ldots k]$. The Pareto order $\prec$ on $\mathbb{R}^k$ is defined: $\mathbf{y}^{(1)} \prec \mathbf{y}^{(2)}$ iff. $\mathbf{y}^{(1)} \preceq \mathbf{y}^{(2)}$ and $\mathbf{y}^{(1)} \neq \mathbf{y}^{(2)}$. A point $\mathbf{x} \in \mathcal{X}$ is called efficient or (Pareto) optimal iff. $\nexists \mathbf{x}' \in \mathcal{X}(F(\mathbf{x}') \prec F(\mathbf{x}))$. The set P_Q of all Pareto optimal solutions of a MOP is called the Pareto set, and its image $F(P_Q)$ is called the Pareto front. Typically, i.e., under certain (mild) assumptions on the model, one can assume that the Pareto set and front of a given continuous MOP form at least locally an object of dimension $k-1$ ([23]).

The Pareto order can also be extended to the family of sets [10], i.e., we say $A \prec B$ iff. $\forall \mathbf{y} \in B \exists \mathbf{y}' \in A(\mathbf{y}' \prec \mathbf{y})$. The set of all efficient points of $\mathcal{X}$ is called the *efficient set*. The image of the efficient set under F is called the *Pareto front*. Multi-objective optimization algorithms (MOAs) often employ a finite multiset $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\mu)}\}$ to approximate the efficient set, whose image under F is denoted by $\mathbf{Y}$. Multi-objective optimization is an active research field that has produced many algorithms for the approximation of the entire Pareto set/front of a given MOP. There exist, for instance, scalarization methods, and mathematical programming techniques that transform the given MOP into an auxiliary scalar optimization problem (SOP) (e.g., [24]). Via solving a clever sequence of such SOPs, one can obtain in many cases suitable Pareto front approximations (e.g., [25–28]). In [29], a Newton method is proposed for multi-objective optimization. Next to these point-wise iterative local search strategies there exist global set-based algorithms such as cell-to-cell mapping techniques and subdivision techniques ([30–32]) as well as specialized evolutionary algorithms ([33–36]). There exist in particular indicator-based evolutionary algorithms (IBEAs) that aim for Pareto front approximations of a given performance indicator (e.g., [12–14]). Widely used performance indicators are the Generational Distance (GD [7]), the Inverted Generational Distance and variants ([8,37,38]), the averaged Hausdorff distance Δ_p ([9,39,40]), and the Hypervolume indicator, which we will use in this work and briefly review in the next section.

Finally, there exist multi-objective continuation methods that make use of the fact that the solution set forms at least locally a manifold (e.g., [23,41–46]).

2.3. Hypervolume Indicator and Its First-Order Derivatives

The hypervolume indicator (HV) [10,47] is defined as the Lebesgue measure of the compact set dominated by a Pareto approximation set $\mathbf{Y} \subset \mathbb{R}^k$ and cut from above by a reference point $\mathbf{r}$:

$$\mathrm{HV}(\mathbf{Y}; \mathbf{r}) = \lambda_k(\{\mathbf{p} \colon \exists \mathbf{y} \in \mathbf{Y}(\mathbf{y} \prec \mathbf{p}) \wedge \mathbf{p} \prec \mathbf{r}\}),$$

where λ_k denotes the Lebesgue measure in $\mathbb{R}^k$. HV is Pareto compliant, i.e., for all $\mathbf{Y} \prec \mathbf{Y}'$, $\mathrm{HV}(\mathbf{Y}; \mathbf{r}) > \mathrm{HV}(\mathbf{Y}'; \mathbf{r})$, and is extensively used to assess the quality of approximation sets to the Pareto front, e.g., in SMS-MOEA [12] and multi-objective Bayesian optimization [48]. Being a set function, it is cumbersome to define the derivative of HV. (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 [49].) Therefore, we follow the generic set-based approach for MOPs [16], which considers a finite approximation sets of size μ vectors as a point in $\mathbb{R}^{\mu n}$, i.e., $\mathbf{X} = [\mathbf{x}^{(1)\top}, \mathbf{x}^{(2)\top}, \ldots, \mathbf{x}^{(\mu)\top}]^\top \in \mathbb{R}^{\mu n}$. Similarly, the image of $\mathbf{X}$ under F can also be represented by a $\mathbb{R}^{\mu k}$-vector: $\mathbf{Y} = [F(\mathbf{x}^{(1)})^\top, F(\mathbf{x}^{(2)})^\top, \ldots, F(\mathbf{x}^{(\mu)})^\top]^\top$. In this sense, the objective function F is also extended as follows:

$$\mathbf{F} \colon \mathcal{X}^\mu \to \mathbb{R}^{\mu k}, \mathbf{X} \mapsto [F(X_1, \ldots, X_n), F(X_{n+1}, \ldots, X_{2n}), \ldots, F(X_{(\mu-1)n+1}, \ldots, X_{\mu n})]^\top.$$

Taking $\mathbf{F}$, we can express the hypervolume indicator as a function on $\mathbb{R}^{\mu n}$:

$$\mathcal{H}_{\mathbf{F}} \colon \mathbb{R}^{\mu n} \to \mathbb{R}_{\geq 0}, \quad \mathbf{X} \mapsto \mathrm{HV}(\mathbf{F}(\mathbf{X}); \mathbf{r}).$$

We will henceforth omit the reference point $\mathbf{r}$ in $\mathcal{H}_\mathbf{F}$ for simplicity. It is straightforward to express the gradient of $\mathcal{H}_\mathbf{F}$ with respect to $\mathbf{X}$ using the chain rule as reported in our previous works [16,19]: $\nabla\mathcal{H}_\mathbf{F}(\mathbf{X}) = (\partial\mathcal{H}_\mathbf{F}/\partial\mathbf{F})(\partial\mathbf{F}/\partial\mathbf{X})$, in which we also discussed the time complexity of computing the hypervolume gradient. It is noted here that an alternative to the computation of the gradient of the entire set, it was also suggested to compute only the gradient of a single point with respect to the hypervolume indicator; this approach is referred to as hypervolume scalarization [50].

2.4. Hypervolume Hessian and Hypervolume Newton Method

Here, we assume F is at least twice continuously differentiable. In general, the Hessian matrix of the hypervolume indicator can be expressed as follows:

$$\begin{aligned}\nabla^2\mathcal{H}_\mathbf{F} &= \frac{\partial}{\partial\mathbf{X}}\left(\frac{\partial\mathcal{H}_\mathbf{F}}{\partial\mathbf{F}}\frac{\partial\mathbf{F}}{\partial\mathbf{X}}\right) = \left[\frac{\partial}{\partial\mathbf{X}}\left(\frac{\partial\mathcal{H}_\mathbf{F}}{\partial\mathbf{F}}\right)\right]^\top\frac{\partial\mathbf{F}}{\partial\mathbf{X}} + \frac{\partial\mathcal{H}_\mathbf{F}}{\partial\mathbf{F}}\frac{\partial^2\mathbf{F}}{\partial\mathbf{X}\partial\mathbf{X}^\top} \\ &= \nabla\mathbf{F}^\top\frac{\partial^2\mathcal{H}_\mathbf{F}}{\partial\mathbf{F}\partial\mathbf{F}^\top}\nabla\mathbf{F} + \frac{\partial\mathcal{H}_\mathbf{F}}{\partial\mathbf{F}}\frac{\partial^2\mathbf{F}}{\partial\mathbf{X}\partial\mathbf{X}^\top}. \end{aligned} \tag{1}$$

Note that in the above expression, $\partial^2\mathcal{H}_\mathbf{F}/\partial\mathbf{F}\partial\mathbf{F}^\top$ and $\partial^2\mathbf{F}/\partial\mathbf{X}\partial\mathbf{X}^\top$ denote the Hessian matrix of the hypervolume indicator with respect to objective points and of the objective function $\mathbf{F}$, respectively. In our previous work [21], we derived the analytical expression of $\nabla^2\mathcal{H}_\mathbf{F}$ for bi-objective cases and analyzed the structure and properties of the hypervolume Hessian matrix. In addition, we implemented a standalone hypervolume Newton (HVN) algorithm for unconstrained MOPs. Moreover, we have shown that the Hessian $\nabla^2\mathcal{H}_\mathbf{F}$ is a tridiagonal 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 $\mathbb{R}^{\mu n}$ [21], thereby ascertaining the safety of applying the HVN method.

The analytical expression of the Hessian matrix for higher dimensions contains the derivatives $\partial\mathcal{H}_\mathbf{F}/\partial x_i^{(\ell)}\partial x_j^{(m)}$, $m = 1,\ldots,\mu, \ell = 1,\ldots,\mu, i = 1,\ldots,n, j = 1,\ldots,n$. To compute these derivatives analytically, the chain rule can be applied (see [21]). In [21]. However, the Hessian matrix of the second mapping—from the points in the objective space $(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(k)})$ to the hypervolume indicator—was only given analytically for two dimensions. The Hessian matrix of this second mapping can be generalized to k dimensional objective spaces, and it is continuous in regular cases. Here, we will only sketch the construction of this matrix and leave the detailed analysis for future research. It is known that in the N-dimensional case, the first derivatives $\partial\,\mathrm{HV}/\partial y_i$ are given by the $(k-1)$-dimensional Lebesgue measure of the $k-1$ dimensional faces of the attainment surface that separates the dominated space from the non-dominated space (see Figure 1, $\partial\,\mathrm{HV}/\partial y_3^{(1)}$). These faces themselves have a derivative that is given by the $(k-2)$-dimensional Lebesgue measure of the $k-2$-dimensional segments (or patches) at the boundary of these faces, which are also changing continuously with y_i (see Figure 1, examples $\partial\,\mathrm{HV}/\partial y_1^{(1)}\partial y_3^{(1)}$ and $\partial\,\mathrm{HV}/\partial y_2^{(2)}\partial y_3^{(1)}$). Note that points in the objective space need to be in a general position to guarantee differentiability; otherwise, one-sided differentiability applies and one of the two derivatives, i.e., when the derivative with perturbed coordinate falls to the dominated subspace, it is always zero [16].

In this work, however, rather than investigating in detail the analytical and computational properties of the Hessian for more than two objective functions, we compute the second-order derivative $\partial^2\mathcal{H}_\mathbf{F}/\partial\mathbf{F}\partial\mathbf{F}^\top$ with the automatic differentiation (AD) method [51] and focus on solving equality-constrained MOPs using the Hessian matrix of the hypervolume indicator.

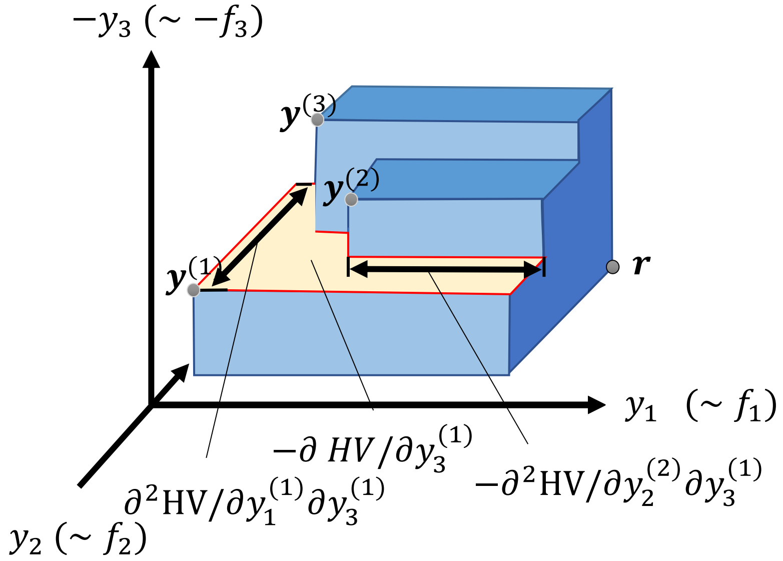

$-y_3\ (\sim -f_3)$
$y^{(3)}$
$y^{(2)}$
$y^{(1)}$
r
$y_1\ \ (\sim f_1)$
$-\partial\, HV/\partial y_3^{(1)}$
$-\partial^2\mathrm{HV}/\partial y_2^{(2)}\partial y_3^{(1)}$
$\partial^2\mathrm{HV}/\partial y_1^{(1)}\partial y_3^{(1)}$
$y_2\ (\sim f_2)$

Figure 1. Example of a hypervolume indicator Hessian computation in three-dimensional objective space with a collection of points $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(3)}\}$ and reference point $\mathbf{r}$.

3. Hypervolume Newton Method for Constrained MOPs

In this section, we first describe the base method of HVN for the treatment of equality constrained MOPs and will then discuss how to deal with inequalities and with dominated points that may be computed during the run of the Newton method.

3.1. Handling Equalities

Consider a continuous equality-constrained MOP of the form

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & F(\mathbf{x}), \\ \text{s.t.} \quad & h(\mathbf{x}) = 0, \end{aligned} \tag{2}$$

where $h(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_p(\mathbf{x}))^\top$, and $h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, p$, being the i-th equality constraint. The objective map is defined by $F : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^k$, where $f_i : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ is the i-the individual objective to be considered in the MOP. The feasible set is given by:

$$Q = \{\mathbf{x} \in \mathcal{X}\ :\ h(\mathbf{x}) = 0\}. \tag{3}$$

The set (population) based hypervolume optimization problem we are considering in this work is the following one:

$$\max_{\substack{X\subset Q \\ |X|=\mu}} \mathrm{HV}(F(X)), \tag{4}$$

where $\mathrm{HV}(F(X))$ denotes the value of the hypervolume for a given set $X = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\mu)}\}$ of magnitude $\mu \in \mathbb{N}$, where each $\mathbf{x}^{(i)} \in \mathbb{R}^n$. Note that the set $X \subset Q$ can be interpreted as a point in $R^{\mu n}$ (via considering $\mathbf{X} = (x_1^{(1)}, \dots, x_n^{(1)}, x_1^{(2)}, \dots, x_n^{(2)}, \dots, x_1^{(\mu)}, \dots, x_n^{(\mu)})$), and hence, problem (4) can be identified by a scalar objective optimization problem of dimension μn.

The feasibility of X (i.e., $X \subset Q$) is identical to

$$h_i(\mathbf{x}^{(j)}) = 0, \qquad i = 1, \dots, p,\ j = 1, \dots, \mu. \tag{5}$$

For the related set-based equality constraints, we define for $i \in \{1, \dots, p\}$ and $j \in \{1, \dots, \mu\}$

$$h_{i,j} : \mathbb{R}^{\mu n} \to \mathbb{R}, \qquad h_{i,j}(\mathbf{X}) = h_i(\mathbf{x}^{(j)}). \tag{6}$$

For checking the feasibility of all decision points, we define $\bar{h} : \mathbb{R}^{\mu n} \to \mathbb{R}^{pn}$ via

$$\bar{h}(\mathbf{X}) = \begin{pmatrix} h_{1,1}(\mathbf{X}) \\ h_{2,1}(\mathbf{X}) \\ \vdots \\ h_{p,1}(\mathbf{X}) \\ h_{1,2}(\mathbf{X}) \\ h_{2,2}(\mathbf{X}) \\ \vdots \\ h_{p,2}(\mathbf{X}) \\ \vdots \\ h_{p,n}(\mathbf{X}) \end{pmatrix} =: \begin{pmatrix} \bar{h}_1(\mathbf{X}) \\ \bar{h}_2(\mathbf{X}) \\ \vdots \\ \bar{h}_p(\mathbf{X}) \\ \bar{h}_{p+1}(\mathbf{X}) \\ \bar{h}_{p+2}(\mathbf{X}) \\ \vdots \\ \bar{h}_{2p}(\mathbf{X}) \\ \vdots \\ \bar{h}_{pn}(\mathbf{X}) \end{pmatrix}, \tag{7}$$

then its Jacobian is given by

$$\bar{H} := \nabla \bar{h}(\mathbf{X}) = \operatorname{diag}\Big(H(\mathbf{x}^{(1)}), \dots, H(\mathbf{x}^{(\mu)})\Big) \in \mathbb{R}^{\mu p \times \mu n}, \tag{8}$$

where

$$H(\mathbf{x}^{(i)}) = \begin{pmatrix} \nabla h_1(\mathbf{x}^{(i)})^\top \\ \vdots \\ \nabla h_p(\mathbf{x}^{(i)})^\top \end{pmatrix} \in \mathbb{R}^{p \times n}. \tag{9}$$

The Karush-Kuhn-Tucker (KKT) equations of the problem (4) hence read as

$$\begin{aligned} \nabla \mathcal{H}_{\mathbf{F}}(\mathbf{X}) + \bar{H}^\top \boldsymbol{\lambda} = 0 \\ \bar{h}(\mathbf{X}) = 0, \end{aligned} \tag{10}$$

for a Lagrange multiplier (or the dual variable) $\boldsymbol{\lambda} \in \mathbb{R}^{\mu p}$ which directly leads to the root finding problem

$$\begin{aligned} G \colon \mathbb{R}^{n(\mu+p)} &\to \mathbb{R}^{n(\mu+p)} \\ G(\mathbf{X}, \boldsymbol{\lambda}) &= \begin{pmatrix} \nabla \mathcal{H}_{\mathbf{F}}(\mathbf{X}) + \bar{H}^\top \boldsymbol{\lambda} \\ \bar{h}(\mathbf{X}) \end{pmatrix} = 0, \end{aligned} \tag{11}$$

where $\lambda \in \mathbb{R}^{\mu n}$. The Jacobian of G at $(X, \lambda)^T$ is given by

$$DG(\mathbf{X}, \boldsymbol{\lambda}) = \begin{pmatrix} \nabla^2 \mathcal{H}_{\mathbf{F}}(\mathbf{X}) + \mathbf{M} & \bar{H}^\top \\ \bar{H} & 0 \end{pmatrix} \in \mathbb{R}^{\mu(n+p) \times \mu(n+p)}, \tag{12}$$

where

$$\mathbf{M} = \sum_{j=1}^{\mu p} \lambda_i \nabla^2 \bar{h}_j(\mathbf{X}) \in \mathbb{R}^{\mu n \times \mu n}. \tag{13}$$

Denoting by $\mathbf{X}_t \in \mathbb{R}^{\mu n}$ and $\boldsymbol{\lambda}_t \in \mathbb{R}^{\mu p}$, the variables in iteration t, a Newton step for problem (11) is given by

$$\begin{pmatrix} \mathbf{X}_{t+1} \\ \boldsymbol{\lambda}_{t+1} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_t \\ \boldsymbol{\lambda}_t \end{pmatrix} - DG(\mathbf{X}_t, \boldsymbol{\lambda}_t)^{-1} G(\mathbf{X}_t, \boldsymbol{\lambda}_t). \tag{14}$$

In our computations, we have omitted M in DG. A Newton step is hence obtained by solving

$$\begin{pmatrix} \nabla^2 \mathcal{H}_{\mathbf{F}}(\mathbf{X}_t) & \bar{H}^\top \\ \bar{H} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{X}_{t+1} - \mathbf{X}_t \\ \boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t \end{pmatrix} = - \begin{pmatrix} \nabla \mathcal{H}_{\mathbf{F}}(\mathbf{X}_t) + \bar{H}^\top \boldsymbol{\lambda}_t \\ \bar{h}(\mathbf{X}_t) \end{pmatrix}. \tag{15}$$

3.2. Handling Inequalities

In order to handle inequalities, we have chosen an active set approach which we will discuss in the following. This approach is straightforward; however, it has led to satisfying results in our computations, in particular when the initial candidate set was computed by the evolutionary algorithm.

Assume problem (2) contains inequalities of the form

$$g(\mathbf{x}) \leq 0, \tag{16}$$

where $g(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x}))^\top$ and $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$, is the i-th inequality constraint. Analogous to the equality-constrained case, we define the feasibility of $\mathbf{X} = (x_1^{(1)}, \dots, x_n^{(1)}, x_1^{(2)}, \dots, x_n^{(2)}, \dots, x_1^{(\mu)}, \dots, x_n^{(\mu)})$ by

$$g_i(\mathbf{x}^{(j)}) \leq 0, \qquad i = 1, \dots, m,\ j = 1, \dots, \mu. \tag{17}$$

Define for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, \mu\}$

$$g_{i,j} : \mathbb{R}^{\mu n} \to \mathbb{R}, \qquad g_{i,j}(\mathbf{X}) = g_i(\mathbf{x}^{(j)}) \tag{18}$$

and $\bar{g} : \mathbb{R}^{\mu n} \to \mathbb{R}^{mn}$ by

$$\bar{g}(\mathbf{X}) = \begin{pmatrix} g_{1,1}(\mathbf{X}) \\ h_{2,1}(\mathbf{X}) \\ \vdots \\ h_{m,1}(\mathbf{X}) \\ h_{1,2}(\mathbf{X}) \\ h_{2,2}(\mathbf{X}) \\ \vdots \\ h_{m,2}(\mathbf{X}) \\ \vdots \\ h_{m,n}(\mathbf{X}) \end{pmatrix} =: \begin{pmatrix} \bar{g}_1(\mathbf{X}) \\ \bar{g}_2(\mathbf{X}) \\ \vdots \\ \bar{g}_m(\mathbf{X}) \\ \bar{g}_{m+1}(\mathbf{X}) \\ \bar{g}_{m+2}(\mathbf{X}) \\ \vdots \\ \bar{g}_{2m}(\mathbf{X}) \\ \vdots \\ \bar{g}_{mn}(\mathbf{X}) \end{pmatrix}. \tag{19}$$

The active set we have used is as follows: if for an inequality constraint it holds

$$\bar{g}_l(\mathbf{X}) > -\text{tol} \tag{20}$$

for a given tolerance tol > 0 at $\mathbf{X}$, then we impose the equality

$$\bar{g}_l(\mathbf{X}) = 0, \tag{21}$$

(i.e., it will be added to the set of equalities) while all other inequalities are disregarded at $\mathbf{X}$.

3.3. Handling Dominated Points

Since Newton's method tends to realize relatively longer steps, it often occurs that some decision points are dominated after a Newton step/iteration. Therefore, it is necessary to discuss how the equality-constrained HVN method behaves in this case. For the reason that will become clear during our discussion, we will investigate two scenarios: (1) *infeasible and dominated points* and (2) *feasible but dominated points*.

For the first scenario, we consider the simplest case, where $p = 1$ and there is only one dominated point. Without loss of generality, we can assume that for an approximation set $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\mu)}\} \subseteq \mathcal{X}$, $\mathbf{x}^{(1)}$ is dominated by at least one of the remaining $\mu - 1$

points (as the indices are assigned to $\mathbf{X}$ arbitrarily). Denoting by $\mathbf{X}^{(-1)}$ the approximation set after removing $\mathbf{x}^{(1)}$, we can express the constraint function on $\mathbf{X}^{(-1)}$ as:

$$\begin{aligned}
&\bar{h}^*(\mathbf{X}^{(-1)}) \colon \mathbb{R}^{(\mu-1)n} \to \mathbb{R}^{\mu-1}, \quad \mathbf{X}^{(-1)} \mapsto \left(\bar{h}_2^*(\mathbf{X}^{(-1)}), \bar{h}_3^*(\mathbf{X}^{(-1)}), \ldots, \bar{h}_\mu^*(\mathbf{X}^{(-1)})\right)^\top, \\
&\bar{h}_j^*(\mathbf{X}^{(-1)}) \colon \mathbb{R}^{(\mu-1)n} \to \mathbb{R}, \qquad \mathbf{X}^{(-1)} \mapsto h(\mathbf{x}^{(j)}), \quad j \in [2 \ldots \mu].
\end{aligned}$$

Note that we are only considering the special case of one constraint, i.e., $p = 1$. The root finding problem G can re-expressed in the following form, equivalent to Equation (11):

$$G(\mathbf{X}, \boldsymbol{\lambda}) = \begin{pmatrix} \lambda_1 \nabla h(\mathbf{x}^{(1)}) \\ \nabla \mathcal{H}_{\mathbf{F}}\left(\mathbf{X}^{(-1)}\right) + \sum_{i=2}^{\mu} \lambda_j \nabla \bar{h}_j^*(\mathbf{X}^{(-1)}) \\ h(\mathbf{x}^{(1)}) \\ \bar{h}^*(\mathbf{X}^{(-1)}) \end{pmatrix}.$$

Let $\mu' = \mu - 1$ and $\mathbf{H}\left(\mathbf{X}^{(-1)}\right) = [\nabla \bar{h}_2^*(\mathbf{X}^{(-1)}), \ldots, \nabla \bar{h}_\mu^*(\mathbf{X}^{(-1)})] \in \mathbb{R}^{\mu' n \times \mu'}$, we express the derivative of G as a block matrix:

$$DG(\mathbf{X}, \boldsymbol{\lambda}) = \left[\begin{array}{c|ccc} \lambda_1 \nabla^2 h(\mathbf{x}^{(1)}) & \mathbf{0}_{n \times \mu' n} & \nabla h(\mathbf{x}^{(1)}) & \mathbf{0}_{n \times \mu'} \\ \hline \mathbf{0}_{\mu' n \times n} & \nabla^2 \mathcal{H}_{\mathbf{F}}\left(\mathbf{X}^{(-1)}\right) + \sum_{j=2}^{\mu} \nabla^2 \bar{h}_j^*\left(\mathbf{X}^{(-1)}\right) & \mathbf{0}_{\mu' n \times 1} & \mathbf{H}\left(\mathbf{X}^{(-1)}\right) \\ \nabla h(\mathbf{x}^{(1)})^\top & \mathbf{0}_{1 \times \mu' n} & 0 & \mathbf{0}_{1 \times \mu'} \\ \mathbf{0}_{\mu' \times n} & \mathbf{H}\left(\mathbf{X}^{(-1)}\right)^\top & \mathbf{0}_{\mu' \times 1} & \mathbf{0}_{\mu' \times \mu'} \end{array}\right].$$

Note that the upper left 2×2 block equals $\nabla^2 \mathcal{H}_{\mathbf{F}}(\mathbf{X}) + \sum_{i=1}^{\mu} \nabla^2 h_i^-(\mathbf{X})$. The inverse of DG can be obtained by applying the Schur complement recursively (first consider the block partition indicated above and then apply it again to each partition), provided that both $\nabla^2 h(\mathbf{x}^{(1)})$ and $\nabla^2 \mathcal{H}_{\mathbf{F}}\left(\mathbf{X}^{(-1)}\right)$ are non-singular.

After simplification, the inverse of DG admits the following form:

$$[DG(\mathbf{X}, \boldsymbol{\lambda})]^{-1} = \left[\begin{array}{cccc} \left(\mathbf{I}_{n \times n} - \left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} \mathbf{A} \mathbf{g} \mathbf{g}^\top\right) \mathbf{A} & \mathbf{0}_{n \times \mu' n} & \left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} \mathbf{A} \mathbf{g} & \mathbf{0}_{n \times \mu'} \\ \hline \mathbf{0}_{\mu' n \times n} & \mathbf{B}\left(\mathbf{I} - \mathbf{H}(\mathbf{H}^\top \mathbf{B} \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{B}\right) & \mathbf{0}_{\mu' n \times 1} & \mathbf{0}_{\mu' n \times \mu'} \\ \left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} (\mathbf{A} \mathbf{g})^\top & \mathbf{0}_{1 \times \mu' n} & -\left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} & \mathbf{0}_{1 \times \mu'} \\ \mathbf{0}_{\mu' n \times n} & \mathbf{0}_{\mu' \times \mu' n} & \mathbf{0}_{\mu' \times 1} & -(\mathbf{H}^\top \mathbf{B} \mathbf{H})^{-1} \end{array}\right],$$

where $\mathbf{g} = \nabla h(\mathbf{x}^{(1)}), \mathbf{A} = [\lambda_1 \nabla^2 h(\mathbf{x}^{(1)})]^{-1}, \mathbf{H} = \mathbf{H}(\mathbf{X}^{(-1)})$, and

$$\mathbf{B} = \left[\nabla^2 \mathcal{H}_{\mathbf{F}}\left(\mathbf{X}^{(-1)}\right) + \sum_{j=2}^{\mu} \nabla^2 \bar{h}_j^*\left(\mathbf{X}^{(-1)}\right)\right]^{-1}.$$

The first row of blocks is of particular interest to us since it determines the search step of $\mathbf{x}^{(1)}$. It is obvious that

$$\begin{aligned}
\Delta \mathbf{x}^{(1)} &= -\left([DG(\mathbf{X}, \boldsymbol{\lambda})]^{-1}\right)_{[1:n, 1:\mu(n+1)]} G(\mathbf{X}, \boldsymbol{\lambda}) \\
&= -\left(\lambda_1 \left(\mathbf{I}_{n \times n} - \left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} \mathbf{A} \mathbf{g} \mathbf{g}^\top\right) \mathbf{A} \mathbf{g} + h(\mathbf{x}^{(1)}) \left(\mathbf{g}^\top \mathbf{A} \mathbf{g}\right)^{-1} \mathbf{A} \mathbf{g}\right) \qquad (22) \\
&= -\frac{h(\mathbf{x}^{(1)})}{\nabla h(\mathbf{x}^{(1)})^\top [\nabla^2 h(\mathbf{x}^{(1)})]^{-1} \nabla h(\mathbf{x}^{(1)})} [\nabla^2 h(\mathbf{x}^{(1)})]^{-1} \nabla h(\mathbf{x}^{(1)}),
\end{aligned}$$

where notation $(\mathbf{M})_{[1:n,1:\mu(n+1)]}$ takes rows from 1 to n and columns from 1 to $\mu(n+1)$ in matrix $\mathbf{M}$. Similarly, the search step of the dual variable is:

$$\begin{aligned}\Delta\lambda_1 &= -\left([DG(\mathbf{X},\boldsymbol{\lambda})]^{-1}\right)_{[\mu n+1,1:\mu(n+1)]} G(\mathbf{X},\boldsymbol{\lambda}) \\ &= -\lambda_1\left(\mathbf{g}^\top\mathbf{A}\mathbf{g}\right)^{-1}(\mathbf{A}\mathbf{g})^\top\mathbf{g} + \left(\mathbf{g}^\top\mathbf{A}\mathbf{g}\right)^{-1} h(\mathbf{x}^{(1)}) \\ &= \lambda_1\left(\frac{h(\mathbf{x}^{(1)})}{\nabla h(\mathbf{x}^{(1)})^\top[\nabla^2 h(\mathbf{x}^{(1)})]^{-1}\nabla h(\mathbf{x}^{(1)})} - 1\right).\end{aligned} \tag{23}$$

Now, consider the function $\hat{h}(\mathbf{x}) = h^2(\mathbf{x})/2$, whose first- and second-order derivatives are:

$$\nabla\hat{h}(\mathbf{x}) = h(\mathbf{x})\nabla h(\mathbf{x}), \quad \nabla^2\hat{h}(\mathbf{x}) = h(\mathbf{x})\nabla^2 h(\mathbf{x}) + \nabla h(\mathbf{x})\nabla h(\mathbf{x})^\top.$$

The global minimum/maximum of $\hat{h}$ corresponds to the feasible set, i.e., $h(\mathbf{x}) = 0$. Hence, Newton iterations that optimize $\hat{h}$ will equivalently find the feasible set. Computing the Newton direction of $\hat{h}$, we have:

$$\begin{aligned}&-\left[\nabla^2\hat{h}(\mathbf{x})\right]^{-1}\hat{h}(\mathbf{x}) \\ &= -[h(\mathbf{x})\nabla^2 h(\mathbf{x})]^{-1}\left(\mathbf{I}_{n\times n} - \frac{\nabla h(\mathbf{x})\nabla h(\mathbf{x})^\top[h(\mathbf{x})\nabla^2 h(\mathbf{x})]^{-1}}{1+\nabla h(\mathbf{x})^\top[h(\mathbf{x})\nabla^2 h(\mathbf{x})]^{-1}\nabla h(\mathbf{x})}\right)h(\mathbf{x})\nabla h(\mathbf{x}) \\ &= -\frac{h(\mathbf{x})}{h(\mathbf{x})+\nabla h(\mathbf{x})^\top[\nabla^2 h(\mathbf{x})]^{-1}\nabla h(\mathbf{x})}[\nabla^2 h(\mathbf{x})]^{-1}\nabla h(\mathbf{x}).\end{aligned} \tag{24}$$

Setting $\mathbf{x} = \mathbf{x}^{(1)}$ in the above equation and comparing it to Equation (22), we notice that the Newton direction of $\hat{h}$ and the hypervolume Newton step $\Delta\mathbf{x}^{(1)}$ only differ by a scalar, which can be neglected in practice since we implement a step-size control to re-scale the search step (see the following sub-section). Therefore, we conclude that for infeasible and dominated points, our HVN method (Equation (15)) only considers the constraint function and moves such decision points to the feasible set rapidly (ideally at quadratic speed when the point is close to the feasible set). This satisfactory property allows for handling infeasible and dominated points without modifying our HVN method.

In addition, due to this nice property, an infeasible point will eventually lie on the feasible set, where it can still be dominated if other feasible points exist. This is precisely the second scenario of our discussion, in which the hypervolume of feasible but dominated points will be zero. To move such points, we propose to employ the famous *non-dominated sorting* [36] procedure, where we partition all feasible points into "layers" of mutually non-dominated ones (formally, anti-chains of Pareto order) and compute the Newton direction for each layer (using Equation (15)) regardless of other dominating layers. In this manner, the HVN method can move all feasible points along the feasible set for achieving a good distribution.

3.4. The HVN Method for Constrained MOPs

Taking the above considerations regarding the HVN method, in this section, we aim to devise and implement a standalone HVN algorithm, which is outlined in Algorithm 1. First, we check if any decision point is feasible (i.e., $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ for some $\mathbf{x}$), where the feasibility can be tested numerically with a pre-defined small threshold (e.g., 10^{-4} used in this work) for the equality constraints. Then, we employ the non-dominated sorting point procedure [36] to partition the feasible points $\mathbf{X}_\mathrm{f}$ into "layers" of mutually non-dominated ones, where the Newton direction (Equation (15)) is calculated separately on each layer. Taking L for the indices of points in a layer and $\mathbf{X}_\mathrm{f}[L]$ for the subset indexed by L, we express this partitioning as $\mathbf{X}_\mathrm{f}[L_1] \prec \mathbf{X}_\mathrm{f}[L_2] \prec \cdots \prec \mathbf{X}_\mathrm{f}[L_q], \forall i \neq j (L_i \cap L_j = \emptyset), \cup_i L_i \subseteq [1\ldots\mu]$. Note that the dominance relation for the remaining infeasible and dominated points is not well-defined, considering the equality constraints since they are incomparable to the feasible ones (also among themselves). In this case, we simply merge them into the first layer L_1 and compute the Newton direction thereof, which can be justified by the observation in Equations (22) and (24). The resulting search direction of the infeasible and dominated points is a Newton direction of the function $h^2/2$. In this treatment, a special case arises when there are no feasible points, usually in the first several iterations of the algorithm.

Algorithm 1: Standalone hypervolume Newton algorithm for equality-constrained MOPs

```
Input: $F$: multi-objective function, $h$: equality constraints, $\mathbf{X}_0$: initial
  approximation set, $B$: maximal iterations;
if $\mathbf{X}_0$ is not given then
    sample $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(\mu)}\right\} \subseteq \mathcal{X}$ uniformly at random
else
    $\mathbf{X} \leftarrow \mathbf{X}_0$
$\mathbf{Y} \leftarrow \left\{F(\mathbf{x}^{(1)}), F(\mathbf{x}^{(2)}), \ldots, F(\mathbf{x}^{(\mu)})\right\}$;
$\mathbf{H} \leftarrow \left\{h(\mathbf{x}^{(1)}), h(\mathbf{x}^{(2)}), \ldots, h(\mathbf{x}^{(\mu)})\right\}$;
$\boldsymbol{\lambda} \leftarrow \{\mathbf{1}_{1\times p}/\mu, \ldots, \mathbf{1}_{1\times p}/\mu\}$ and $|\boldsymbol{\lambda}| = \mu$ ;        ▷ initialize dual variables
$\varepsilon \leftarrow 10^{-4}$;
for $c = 1, 2, \ldots, B$ do
    $S \leftarrow \emptyset$ ;                                 ▷ indices for the feasible subset
    for $i \in [1 \ldots \mu]$ do                              ▷ check the feasibility
        if $\max\left\{|h_1(\mathbf{x}^{(i)})|, \ldots, |h_p(\mathbf{x}^{(i)})|\right\} \leq \varepsilon$ then $S \leftarrow S \cup \{i\}$ ;
    $L_1, L_2, \ldots, L_q \leftarrow$ NON-DOMINATED-SORTING($\mathbf{X}[S]$);          ▷ $\mathbf{X}_\mathbf{f} = \mathbf{X}[S]$
    $L_1 \leftarrow L_1 \cup ([1..\mu] \setminus S)$;
    for $L \in \{L_1, L_2, \ldots, L_q\}$ do
        $\nabla^2\mathcal{H}_\mathbf{F}(\mathbf{X}[L]) \leftarrow$ HYPERVOLUME-HESSIAN($\mathbf{X}[L], \mathbf{Y}[L]$) ;     ▷ Equation (1)
        // Equation (15)
        $\Delta_L \leftarrow$ NEWTON-STEP$\left(\nabla^2\mathcal{H}_\mathbf{F}(\mathbf{X}[L]), \nabla^2 h(\mathbf{X}[L]), \nabla h(\mathbf{X}[L]), \mathbf{H}[L], \boldsymbol{\lambda}[L]\right)$;
        // See Section 3.4
        Determine the step-size $\sigma_L$ with Armijo's backtracking line search;
        // Apply Newton step to the primal-dual pair
        $(\mathbf{X}[L], \boldsymbol{\lambda}[L]) \leftarrow (\mathbf{X}[L], \boldsymbol{\lambda}[L]) + \sigma_L \Delta_L$
    $\mathbf{Y} \leftarrow \left\{F(\mathbf{x}^{(1)}), F(\mathbf{x}^{(2)}), \ldots, F(\mathbf{x}^{(\mu)})\right\}$;
    $\mathbf{H} \leftarrow \left\{h(\mathbf{x}^{(1)}), h(\mathbf{x}^{(2)}), \ldots, h(\mathbf{x}^{(\mu)})\right\}$;
return $\mathbf{X}$;
```

Finally, another important aspect is the step-size control for each Newton step. We propose maintaining individual step-sizes for each partition, which is determined using the well-known Armijo's backtracking line search [52]. In detail, this method starts with an initial step-size σ_0 and tests whether the Euclidean norm of $G(\mathbf{X}, \boldsymbol{\lambda})$ has sufficiently decreased after applying the Newton step to the primal-dual pair $(\mathbf{X}, \boldsymbol{\lambda})$. Since Newton's direction for equality-constrained problems (Equation (15)) is not necessarily an ascent direction for the hypervolume, we take the Euclidean norm $||G(\mathbf{X}, \boldsymbol{\lambda})||$ as the convergence measure since (1) the optimality condition is $G(\mathbf{X}, \boldsymbol{\lambda}) = 0$ (Equation (11)) and (2) the Newton step is always a descent direction of $||G(\mathbf{X}, \boldsymbol{\lambda})||$. Let $\mathbf{Z} = (\mathbf{X}^\top, \boldsymbol{\lambda}^\top)^\top$ be the primal-dual variable and $\Delta\mathbf{Z} = -[DG(\mathbf{Z})]^{-1}G(\mathbf{Z})$, then we have $(\frac{\mathrm{d}}{\mathrm{d}\sigma}||G(\mathbf{Z} + \sigma\Delta\mathbf{Z})||)|_{\sigma=0} = -||G(\mathbf{Z})|| \leq 0$. If the test fails, then we halve the step-size and repeat the test. Notably, for infeasible and dominated points, the test checks if the value of the squared constraint value is sufficiently decreased as the HVN method computes the Newton direction of $h^2/2$ for those points. In our implementation, we use maximally six iterations of such tests, resulting in a minimal step-size of $\sigma_0/64$. As for the initial step-size σ_0, the commonly used value $\sigma_0 = 1$ often leads to Newton steps that jump out of the decision space when the Newton direction is large or the point is in the vicinity of the decision boundary. Therefore, we set it to the minimum of one and the maximal step-size that the primal vector $\mathbf{X}$ can take without leaving the decision space, i.e., $\sigma_0 = \min\{1, \sigma_{\max}\}$. The value of $\sigma_{\max}$ can be calculated in a straightforward way when the decision space is a convex and compact subset of $\mathbb{R}^n$, e.g., a hyperbox.

3.5. Computational Cost

The above method requires the knowledge of the Jacobian and the Hessian of both objective and constraint functions. In this work, we have used automatic differentiation (AD) techniques [53]. Note that finite differences can also be utilized when the AD-computation is not applicable. The AD-computation takes maximally four times the used multiply–add operations taken in evaluating the function value [54]. Hence, to make a fair comparison between HVN and MOEA methods, we will take 4 function evaluations (FEs) and $4 + 6n$ FEs to quantify the computational cost of each AD-computed Jacobian and Hessian, respectively. In total, the number of FEs consumed in each iteration comprises:

$$\#\text{FEs}: \underbrace{\mu}_{F} + \underbrace{\mu}_{h} + \underbrace{4\mu}_{\nabla F} + \underbrace{4\mu}_{\nabla h} + \underbrace{(4+6n)\mu}_{\nabla^2 F} + \underbrace{(4+6n)\mu}_{\nabla^2 h} + \underbrace{6(4\mu+4\mu)}_{\text{step-size control}} = (69+12n)\mu,$$

which amounts to computations of function evaluation, constraint evaluation, Jacobian of the objective function and the constraint, Hessian of the objective function and the constraint, and the backtracking line search of the step-size. Computing the hypervolume Hessian takes $\Theta((\mu n)^3)$ time in addition to the AD-computation of derivatives in Equation (1). For solving Equation (15), we use Cholesky decomposition, which has a computational complexity of $O((\mu(n+p))^3)$. It is certainly desired either to have an analytic expression of the HV Hessian or to exploit the block diagonal structure this matrix will certainly have for AD, which we, however, have to leave for future research.

We have implemented the standalone algorithm in Python, which is accessible at https://github.com/wangronin/HypervolumeDerivatives (accessed on 1 November 2022).

4. Numerical Results

In this section, we present some numerical results of the HVN both as standalone algorithms as well as a local search engine within the NSGA-III algorithm.

4.1. HVN as Standalone Algorithm

We showcase the behavior of the proposed Newton method as a standalone method on three example problems:

$$\begin{aligned}
(\text{P1}): \quad & F(\mathbf{x}) = \left[(\mathbf{x}-\mathbf{1})^2, (\mathbf{x}+\mathbf{1})^2\right]^\top, \\
& h(\mathbf{x}) = \mathbf{x}^2 - 1, \quad \mathcal{X} = [-2,2]^2, \quad \mathbf{r} = [20,20]^\top. \\
(\text{P2}): \quad & F(\mathbf{x}) = \left[(\mathbf{x}-(1,1,0)^\top)^2, (\mathbf{x}-(1,-1,0)^\top)^2, (\mathbf{x}-(-1,1,0)^\top)^2\right]^\top, \\
& h(\mathbf{x}) = \left(\mathbf{x} - \left(\frac{2\sqrt{3}}{3} - 1, 0, -1.5\right)^\top\right)^2 - 1, \quad \mathcal{X} = [-2,2]^3, \quad \mathbf{r} = [38,38,38]^\top. \\
(\text{P3}): \quad & F(\mathbf{x}) = \left[(\mathbf{x}+(1,1,1)^\top)^2, (\mathbf{x}+(1,0,0)^\top)^2, (\mathbf{x}+(2,2,-4)^\top)^2\right]^\top, \\
& g(\mathbf{x}) = -x_0, \quad \mathcal{X} = [-4,4]^3, \quad \mathbf{r} = [90,90,90]^\top.
\end{aligned}$$

Importantly, we will use different initializations of the decision points that are specific to each problem in order to investigate the behavior of the standalone HVN with respect to the characteristic of each problem; We do not aim to provide a unified and systematic initialization method for the standalone HVN in this section. Note that problem P3 defines an inequality constraint on the first component x_0 of the decision point, where the feasible set is $\{\mathbf{x} \in \mathcal{X} \colon x_0 \geq 0\}$, and the optimum lies on the active set of g, i.e., $x_0 = 0$. This problem is meant to test if the proposed HVN algorithm can manage to solve inequality-constrained problems where the optimum is on the active set of the constraint. To measure the empirical performance of the HVN algorithm, we take the Euclidean norm $||G(\mathbf{X}, \boldsymbol{\lambda})||$ since the Newton direction is not necessarily an ascent direction for the hypervolume.

Moreover, since it is well-known that the Newton-like method can be affected by choice of initial solutions, we investigate the performance of the HVN algorithm on problem P1 with three different initializations. Specifically, in the two-dimensional decision space, we create $\mu = 50$ initial decision points on the line segment $x_2 = x_1 - 2, x_1 \in [0, 2]$, where we determine the value of x_1 by (i) taking evenly spaced points (linear), (ii) logistic-transformed evenly spaced points (which makes the points denser around the tails of the line segment), or (iii) logit-transformed evenly spaced points (higher density of points in the middle). The results are illustrated in Figure 2 and Table 1, which shows a set of well-distributed points on the feasible set in the objective space (the red dashed sphere) for all three initializations. In addition, the empirical convergence rate is quadratic regardless of the choice of initialization methods, as reported in Table 1.

Table 1. The evolution of $\|G(\mathbf{X}, \boldsymbol{\lambda})\|$ on problems P1 with three different initialization strategies.

	Linear	**Logistic**	**Logit**
1	4.23×10^{1}	4.55×10^{1}	4.20×10^{1}
2	2.33×10^{1}	2.54×10^{1}	2.27×10^{1}
3	8.81×10^{0}	1.01×10^{1}	8.52×10^{0}
4	8.19×10^{0}	7.82×10^{0}	8.30×10^{0}
5	2.29×10^{0}	2.17×10^{0}	2.29×10^{0}
6	1.06×10^{-1}	8.77×10^{-2}	1.11×10^{-1}
7	1.91×10^{-4}	3.48×10^{-4}	1.93×10^{-3}
8	7.38×10^{-10}	7.05×10^{-7}	1.03×10^{-5}
9	1.76×10^{-14}	1.06×10^{-12}	1.55×10^{-10}
10	1.62×10^{-14}	1.79×10^{-14}	2.33×10^{-14}

The results on problem P2 are depicted in Figure 3 and Table 2 for three different sizes $\mu \in \{20, 40, 60\}$ of the approximation set. The initial decision points are sampled uniformly at random in the convex hull of three points $(1, 1, 0)^\top, (1, -1, 0)^\top$, and $(-1, 0, 0)^\top$. Whereas the final approximation set is well-distributed in the objective space, we observe that empirical convergence of $||G(\mathbf{X}, \boldsymbol{\lambda})||$ is considerably rugged in the first 20–25 iterations, after which quadratic convergence appears. This is indeed attributed to the fact that decision points often become dominated in the first couple of iterations on this problem, resulting in zero hypervolume gradient thereof and hence quite a large norm of $||G(\mathbf{X}, \boldsymbol{\lambda})||$. Nevertheless, the proposed treatment of those dominated points (Algorithm 1), which is based on the non-dominated sorting procedure, is capable of bringing the dominated points to the active set with a quadratic speed. Similarly, the same ruggedness is seen in the convergence chart of problem P3 (shown in Figure 4). On this problem, we again take the setting $\mu \in \{20, 40, 60\}$, and the initial decision points are sampled uniformly at random in the feasible space of $[0, 4] \times [-4, 4]^2$. We extend the HVN algorithm slightly for this inequality-constrained problem in the following way: whenever the decision points are feasible, i.e., $g(\mathbf{x}) \leq 0$, and quite distant from the active set ($g(\mathbf{x}) = 0$, shown as the red plane in Figure 4), we ignore the constraint function when computing the Newton step. When the feasible decision points are sufficiently close to the active set (the distance is less than 10^{-4} in our implementation), we consider $g(\mathbf{x})$ an equality constraint and utilize Equation (15) to compute the Newton step.

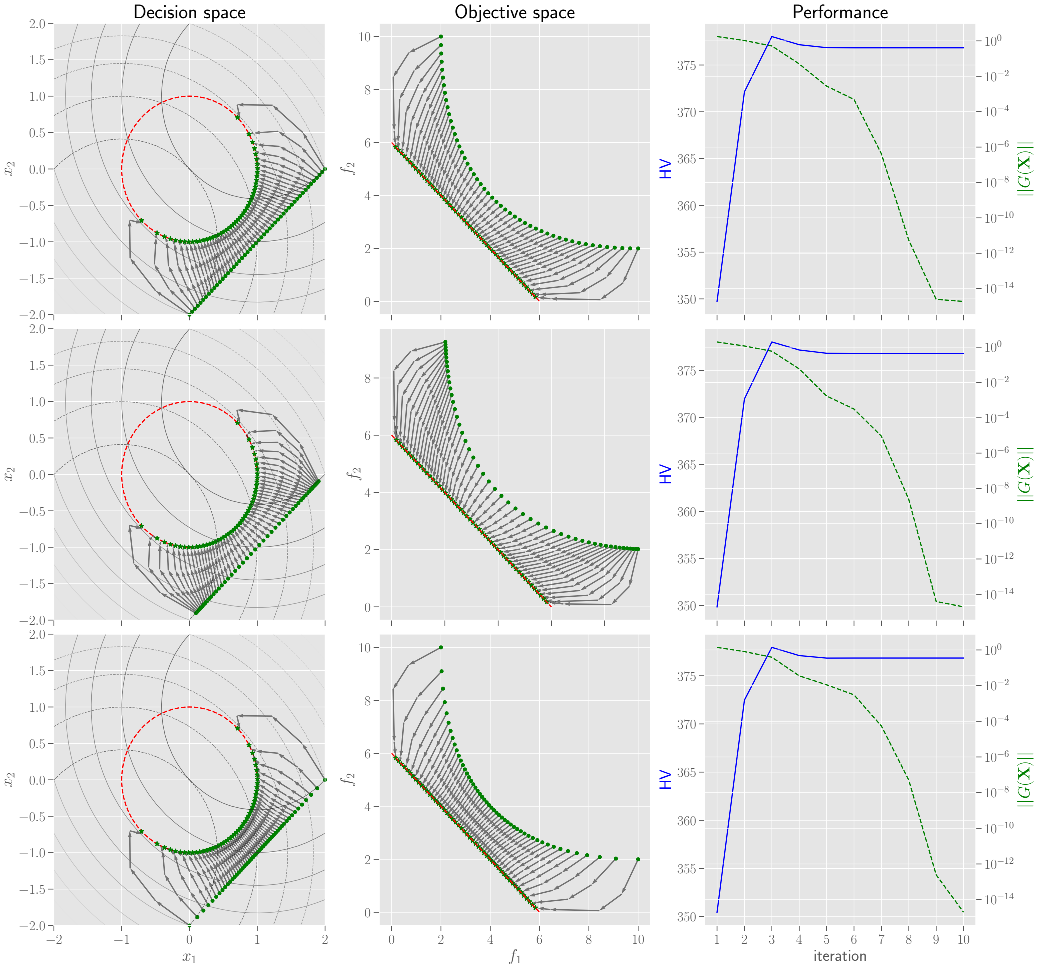


Figure 2. On problem P1, the convergence of the HVN method is shown for three different initializations of the starting approximation set ($\mu = 50$)—linear (**top row**), logistic (**middle**), and logit spacing (**bottom**). We depict the final approximation set (**left column**; green stars), the corresponding objective points (**middle column**; green stars), and the evolution of the HV value and $\|G(\mathbf{X}, \boldsymbol{\lambda})\|$ (**right column**).

Moreover, we test the standalone HVN method on large-scale, complicated MOPs. We choose the well-known DTLZ problems with one spherical constraint [55,56] with $\mu = 200$ decision points, resulting in a relatively large Hessian matrix (for an 11-dimensional decision space and one constraint, the $DG(\mathbf{X}, \boldsymbol{\lambda})$ object is of size 2400×2400). In this case, we use sparse matrix operations for computation efficiency, exploiting the sparsity of the Hessian. Since the DTLZ problems are highly multi-modal, the initial approximation set is generated in a local vicinity of the Pareto set, i.e., $\mathbf{X}^* + 0.02\mathcal{U}(0,1)$, where $\mathbf{X}^*$ is sampled uniformly at random on the Pareto set. We execute the standalone HVN method for 15 iterations and illustrate the result in Figure 5. In the plot, we observe well-distributed final points (green dots) in contrast to non-uniform initial ones (black crosses), showing the standalone HVN works properly as a local method for large-scale problems.

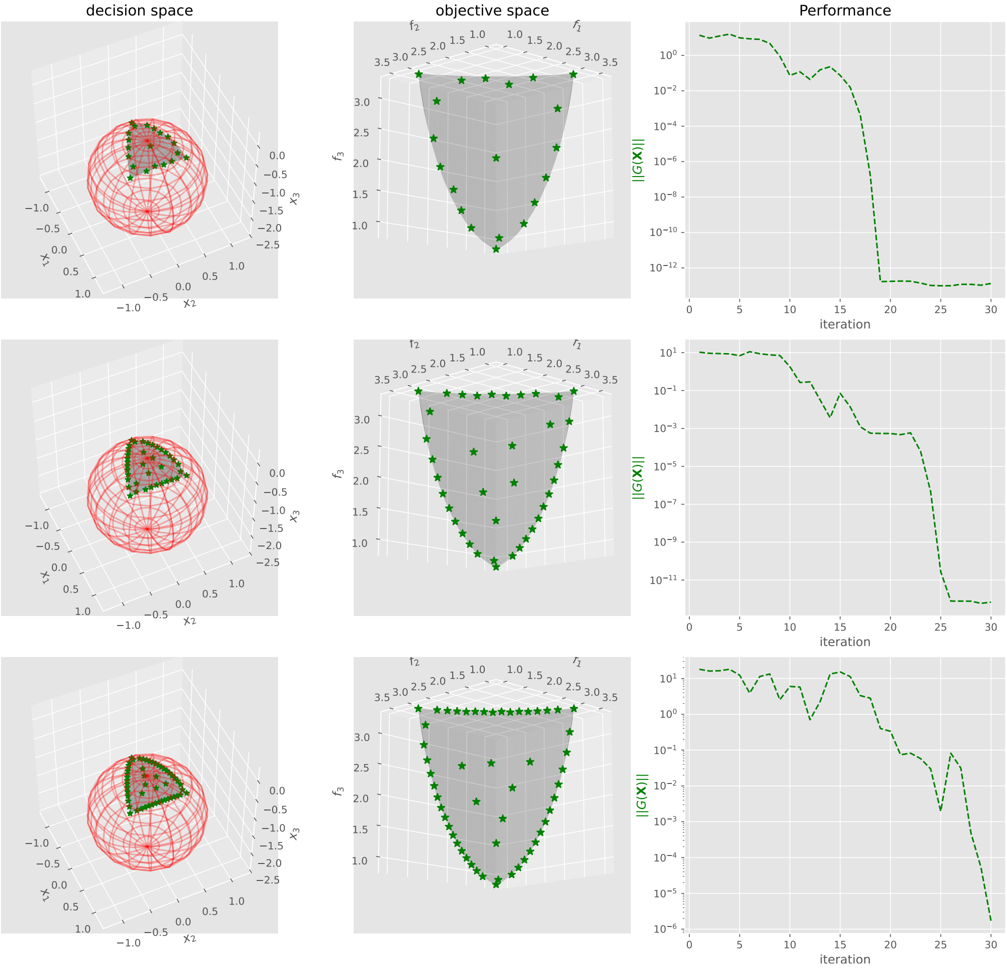


Figure 3. On problem P2 with a spherical constraint, we depict for three sizes of the approximation set ($\mu \in \{20, 40, 60\}$; from **top** to **bottom**), the final approximation set (**left column**; green stars), the corresponding objective points (**middle column**; green stars), and the evolution of the HV value and $\|G(\mathbf{X}, \lambda)\|$ (**right column**). The initial points are sampled uniformly at random in the convex hull of three points $(1, 1, 0)^\top, (1, -1, 0)^\top$, and $(-1, 0, 0)^\top$.

Table 2. The evolution of $\|G(\mathbf{X}, \lambda)\|$ on problems P2 and P3.

	Problem P2			**Problem P3**		
	$\mu = 20$	$\mu = 40$	$\mu = 60$	$\mu = 20$	$\mu = 40$	$\mu = 60$
1	1.365×10^{1}	1.055×10^{1}	18.036836	1.433×10^{3}	569.121097	438.983791
2	9.454×10^{0}	9.259×10^{0}	16.083916	9.368×10^{2}	541.523806	434.703270
3	1.247×10^{1}	8.977×10^{0}	16.403643	1.197×10^{3}	444.774066	365.952443
4	1.589×10^{1}	8.628×10^{0}	18.052126	9.522×10^{2}	261.636562	362.014326
5	9.791×10^{0}	6.888×10^{0}	12.364802	6.194×10^{2}	212.841570	341.897644
6	8.618×10^{0}	1.123×10^{1}	3.899254	5.232×10^{2}	145.076665	254.253017
7	8.024×10^{0}	8.779×10^{0}	11.323440	3.557×10^{2}	103.986300	240.719767
8	4.737×10^{0}	7.632×10^{0}	13.320606	2.419×10^{2}	57.592159	165.603954
9	9.037×10^{-1}	7.090×10^{0}	2.543622	1.511×10^{2}	12.628821	109.411195
10	7.393×10^{-2}	1.816×10^{0}	5.984437	8.527×10^{1}	0.104307	70.516402
11	1.182×10^{-1}	2.660×10^{-1}	5.749496	3.732×10^{1}	0.097777	41.699152
12	4.399×10^{-2}	2.877×10^{-1}	0.702964	3.248×10^{0}	0.097013	19.525977
13	1.535×10^{-1}	3.232×10^{-2}	2.240449	2.008×10^{0}	0.096634	0.447690
14	2.299×10^{-1}	3.694×10^{-3}	13.274468	1.829×10^{0}	0.096256	0.257345

Table 2. *Cont.*

	Problem P2			**Problem P3**		
	$\mu = 20$	$\mu = 40$	$\mu = 60$	$\mu = 20$	$\mu = 40$	$\mu = 60$
15	7.425×10^{-2}	7.159×10^{-2}	15.201915	5.425×10^{-2}	0.005277	2.066379
16	1.572×10^{-2}	1.378×10^{-2}	11.273571	1.735×10^{-1}	0.002934	2.016149
17	4.216×10^{-4}	1.231×10^{-3}	3.318978	6.372×10^{-6}	0.001602	1.019636
18	1.630×10^{-7}	5.630×10^{-4}	2.818340	9.702×10^{-3}	0.001552	0.944297
19	1.674×10^{-13}	5.454×10^{-4}	0.400360	9.373×10^{-5}	0.001528	4.904926
20	1.733×10^{-13}	5.411×10^{-4}	0.335107	2.546×10^{-8}	0.001522	2.937413
21	1.803×10^{-13}	4.697×10^{-4}	0.074058	7.243×10^{-12}	0.001516	3.118031
22	1.761×10^{-13}	5.901×10^{-4}	0.081798	8.897×10^{-12}	0.001139	0.336917
23	1.384×10^{-13}	6.140×10^{-5}	0.057776	6.654×10^{-12}	0.001072	0.004270
24	1.020×10^{-13}	4.508×10^{-7}	0.029809	7.210×10^{-12}	0.001010	0.000866
25	9.765×10^{-14}	2.759×10^{-11}	0.001956	5.851×10^{-12}	0.000994	0.000489
26	9.788×10^{-14}	7.794×10^{-13}	0.081949	5.851×10^{-12}	0.000990	0.000477
27	1.177×10^{-13}	7.767×10^{-13}	0.031275	5.851×10^{-12}	0.000954	0.000459
28	1.176×10^{-13}	7.688×10^{-13}	0.000492	5.851×10^{-12}	0.128140	0.000460
29	1.052×10^{-13}	5.918×10^{-13}	0.000053	5.851×10^{-12}	0.252721	0.000460
30	1.314×10^{-13}	6.821×10^{-13}	0.000002	5.851×10^{-12}	0.022233	0.000460

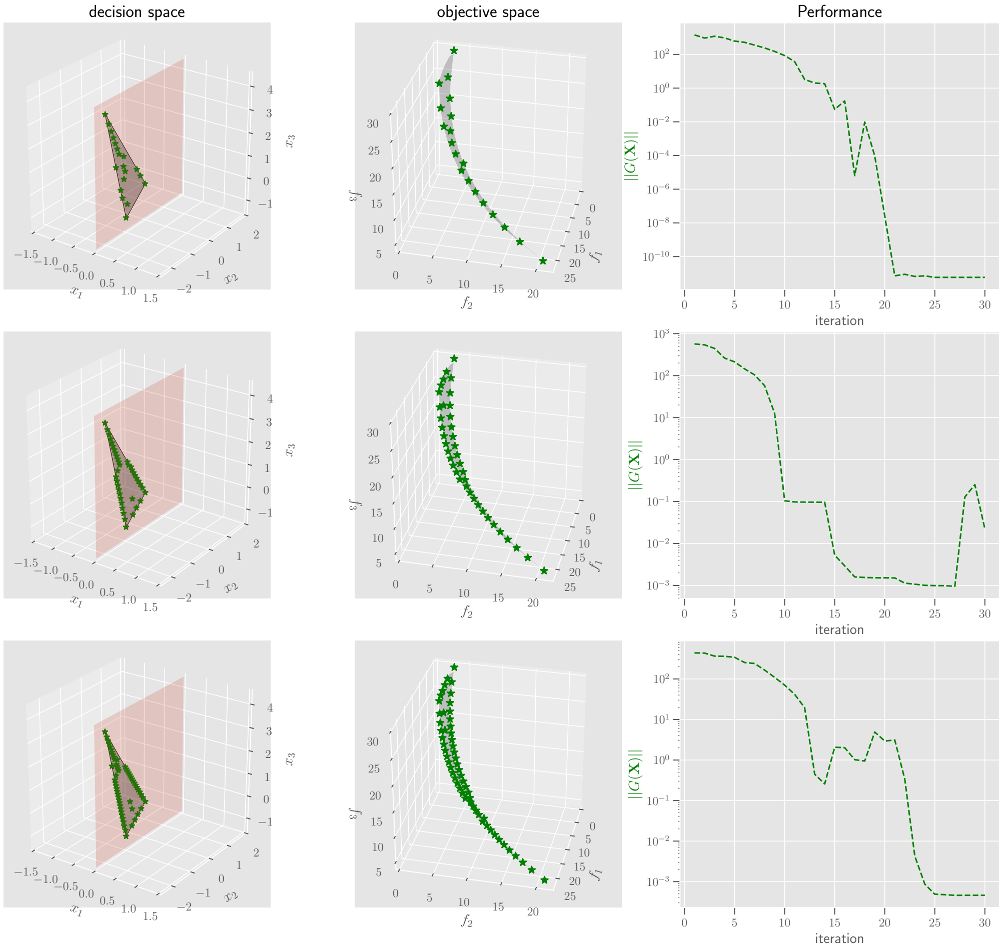


Figure 4. On problem P3 with a spherical constraint, we depict for three sizes of the initial approximation set ($\mu \in \{20, 40, 60\}$; from **top** to **bottom**), the final approximation set (**left column**; green stars), the corresponding objective points (**middle column**; green stars), and the evolution of the HV value and $\|G(\mathbf{X})\|$ (**right column**). The initial decision points are sampled uniformly at random in the feasible space of $[0, 4] \times [-4, 4]^2$.

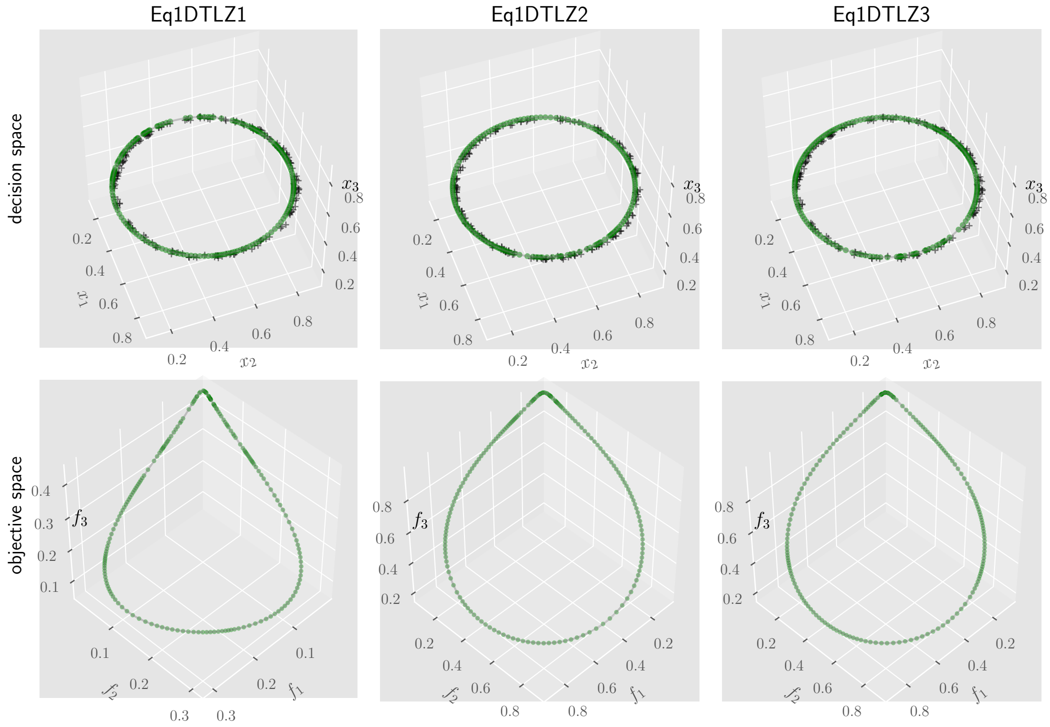


Figure 5. On Eq-DTLZ1-3 problems, the HVN method starts from a small local perturbation (black crosses) of the Pareto set (sphere in the decision space), i.e., $\mathbf{X}^* + 0.02\mathcal{U}(0,1)$, where $\mathbf{X}^*$ (of size 200) is sampled uniformly at random on the Pareto set. The final approximation set of the HVN method is depicted as green points. Only the first three search dimensions are shown for the decision space.

4.2. HVN within NSGA-III

In this section, we investigate the empirical performance of the HVN algorithm on more complicated, equality-constrained DTLZ (Eq-DTLZ) problems [55,56] and their inverted counterparts (Eq-IDTLZ). As Newton-like algorithms are local methods, running the standalone algorithm (Algorithm 1) will stagnate at local Pareto sets. Therefore, we hybridize the HVN algorithm with an MOEA, in which we first execute the MOEA for a pre-defined budget to overcome the local optimum and get close to the global Pareto set, and then initialize the HVN algorithm from the final approximation set of the MOEA to make local refinements. We summarize this hybrid approach in Algorithm 2. Notably, in line 3, we transfer the whole approximation set (rather than only the non-dominated points) to HVN upon the termination of MOEA since the standalone HVN method is able to move dominated points towards the feasible set at quadratic speed, as proven in Section 3.3.

Algorithm 2: Hybridization of HVN and MOEA

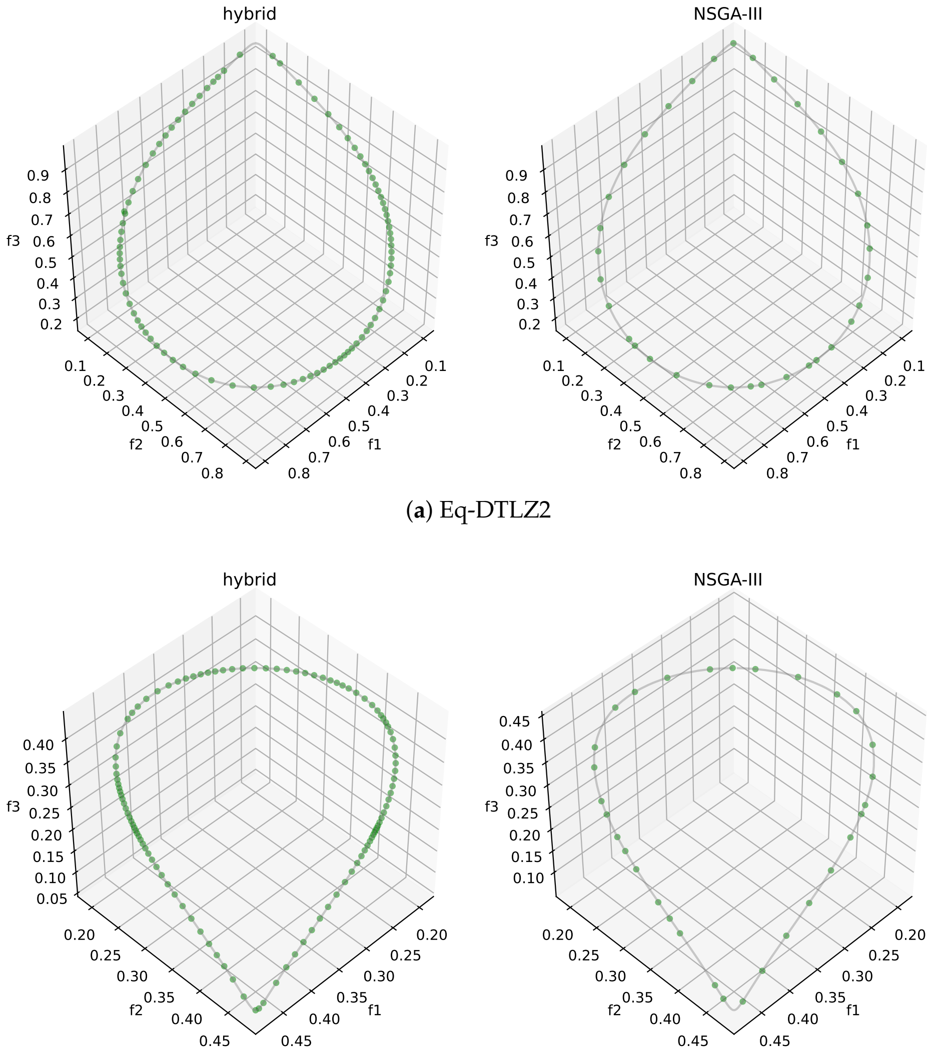
```
1 Input: $F$: multi-objective function, $\mathbf{h}$: equality constraints, $B_1$: #iteration for
    MOEA, $B_2$: #iteration for HVN;
2 $\mathbf{X} \leftarrow \text{MOEA}(F, \mu, \mathbf{h}, B_1)$ ;            ▷ $\mathbf{X}$ is the entire approximation set
3 $\mathbf{X}' \leftarrow \text{HVN}(F, \mathbf{h}, \mathbf{X}, B_2)$;                              ▷ Algorithm 1
4 return $\mathbf{X}'$;
```

The following empirical study aims to check whether the hybridization approach can achieve a better final approximation set/front than an MOEA alone under the same computation budget. As for the test problem, a single spherical constraint $h(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.16$ is imposed on problems DTLZ1 $-$ 4. The decision space is $[0,1]^{11}$, the reference point is $\mathbf{r} = (1,1,1)^\top$ for HVN, and the approximation set is of size 200. Here, we choose the well-known NSGA-III algorithm [34,35], where the equality constraints are handled using the adaptive ε-constraint handling technique. We utilize the implementation in the Pymoo library: https://pymoo.org/constraints/eps.html (accessed on 1 November 2022). The method considers a solution feasible subject to a small ε threshold, which decreases linearly to zero. The initial value of ε is set to the average constraint value of the initial population. In our experiment, we control the ε decrease to zero after 50% of the iterations of NSGA-III. In addition, we use Das and Dennis's approach [28] to generate well-spaced reference directions (18 partitions which lead to 190 directions) for NSGA-III. As for its hyperparameters, we use the default setting: $\eta = 30$ and $p = 1$ for simulated binary crossover and $\eta = 20$ for polynomial mutation. Furthermore, the hybrid algorithm first executes NSGA-III with $\mu = 200$ for 1000 iterations and then runs the HVN method for 10 iterations. In HVN, the total function evaluations and AD operations take ca. 270 s CPU time on an Intel(R) Core(TM) i5-8257U CPU. Considering the CPU time of a single function evaluation, which is on average ca. 5.6×10^{-5} s measured on the same hardware, the total function evaluations plus the AD operations are equivalent to roughly $270/5.6 \times 10^{-5} \approx 4.8 \times 10^5$ FEs. Therefore, the total budget of the hybrid algorithm is roughly $4.8 \times 10^5/200 + 1000 \approx 3400$ iterations. We will execute the standalone NSGA-III algorithm for the same iterations to keep the fairness of comparisons.

We first depict one example of the final approximation set (only the non-dominated subset is shown) in Figure 6 for both methods, where we clearly observe that the hybridization achieves much more non-dominated points than NSGA-III. Second, we show, in Table 3, the hypervolume indicator value and the number of final non-dominated points for both algorithms obtained from 15 independent runs. In addition, we compute the above metrics for the hybrid algorithm right before the HVN phase starts (NSGA-III (1000) in the table), showing the progress that HVN manages to make. From the results, we conclude that the hybrid algorithm significantly improves upon the hypervolume metric and outputs substantially more non-dominated points than NSGA-III alone. We conjecture that the observed advantage of the hybrid algorithm is very likely attributed to HVN's ability to move dominated points to the feasible set with quadratic convergence (see Section 3), which disregards the objective function and thereby its multi-modal landscape.

Table 3. On Eq-DTLZ1-4 and Eq-IDTLZ1-4 problems, the sample mean and standard error of the hypervolume (HV) value and the number of final non-dominated (ND) points over 15 independent runs for each algorithm. The hypervolume values are computed with reference point $(1,1,1)^\top$ for all problems except Eq-DTLZ4, Eq-IDTLZ3, and Eq-IDTLZ4, which we use $(1.2, 5 \times 10^{-3}, 5 \times 10^{-4})^\top$, $(800, 800, 700)^\top$, and $(-0.4, 0.6, 0.6)^\top$, respectively. The initial population is $\mu = 200$ for all algorithms. Hybridization = NSGA-III (iter = 1000) + HVN (iter = 10), which consumes roughly the same CPU time on function evaluations with NSGA-III for 3400 iterations (see caption of Figure 6 for the detail).

	Eq-DTLZ1		**Eq-DTLZ2**		**Eq-DTLZ3**		**Eq-DTLZ4**	
Algorithm	**HV**	**#ND**	**HV**	**#ND**	**HV**	**#ND**	**HV**	**#ND**
NSGA-III (1000)	$0.867 \pm 1.4 \times 10^{-3}$	28.4 ± 0.7	$0.297 \pm 1.9 \times 10^{-3}$	32.7 ± 0.9	$0.292 \pm 1.9 \times 10^{-3}$	26.0 ± 1.0	$8.4 \times 10^{-4} \pm 7.0 \times 10^{-5}$	12.3 ± 0.8
Hybridization	$0.876 \pm 2.4 \times 10^{-4}$	80.9 ± 2.0	$0.324 \pm 3.6 \times 10^{-4}$	95.3 ± 1.9	$0.321 \pm 6.6 \times 10^{-4}$	75.2 ± 2.4	$1.1 \times 10^{-3} \pm 5.1 \times 10^{-5}$	200.0 ± 0.0
NSGA-III (3400)	$0.873 \pm 4.5 \times 10^{-4}$	38.5 ± 1.3	$0.304 \pm 9.2 \times 10^{-4}$	32.6 ± 0.9	$0.301 \pm 1.1 \times 10^{-3}$	30.1 ± 0.7	$9.2 \times 10^{-4} \pm 5.2 \times 10^{-5}$	14.5 ± 0.6
	Eq-IDTLZ1		**Eq-IDTLZ2**		**Eq-IDTLZ3**		**Eq-IDTLZ4**	
Algorithm	**HV**	**#ND**	**HV**	**#ND**	**HV**	**#ND**	**HV**	**#ND**
NSGA-III (1000)	$0.517 \pm 1.8 \times 10^{-2}$	23.2 ± 0.5	$3.224 \pm 2.0 \times 10^{-2}$	74.1 ± 1.2	$1.5\times 10^{9} \pm 8.0 \times 10^{6}$	81.7 ± 1.6	$8.4 \times 10^{-4} \pm 7.0 \times 10^{-5}$	12.3 ± 0.8
Hybridization	$0.534 \pm 1.5 \times 10^{-3}$	112.1 ± 2.1	$3.388 \pm 1.7 \times 10^{-2}$	198.2 ± 0.4	$1.6\times 10^{9} \pm 5.4 \times 10^{6}$	197.1 ± 0.4	$1.1 \times 10^{-3} \pm 5.1 \times 10^{-5}$	200.0 ± 0.0
NSGA-III (3400)	$0.529 \pm 2.9 \times 10^{-4}$	33.4 ± 0.4	$3.359 \pm 4.7 \times 10^{-3}$	88.3 ± 0.4	$1.5\times 10^{9} \pm 2.5 \times 10^{6}$	92.1 ± 0.8	$9.2 \times 10^{-4} \pm 5.2 \times 10^{-5}$	14.5 ± 0.6



(**a**) Eq-DTLZ2

(**b**) Eq-IDTLZ1

Figure 6. On the Eq-DTLZ2 (**a**) and the Eq-IDTLZ1 (**b**) problem, we compare the hybridization of HVN and NSGA-III to NSGA-III with roughly the same budget: for the former, the hybrid algorithm first executes NSGA-III with $\mu = 200$ for 1000 iterations and then runs the HVN method for 10 iterations. In HVN, the total function evaluations and AD takes ca. 270 s CPU time on an Intel(R) Core(TM) i5-8257U CPU, which corresponds to ca. 4.8×10^5 FEs. Hence, for the latter, we set 3400 (=$4.8 \times 10^5 / 200 + 1000$) iterations in total for $\mu = 200$. We use the same hyperparameter setting for the standalone NSGA-III and the one used in the hybridization. The decision space is $[0,1]^{11}$, and the reference point is $(1,1,1)^\top$ for HVN.

5. Conclusions

In this paper, we propose a hypervolume Newton method for equality-constrained multi-objective optimization problems (MOPs) under the assumption that both the objective and the constraint functions are twice continuously differentiable. Based on previous works on set-oriented hypervolume Hessian matrix and hypervolume Newton (HVN) method for unconstrained MOPs, we propose, in this paper, the generalization of the HVN for equality-constrained problems and also elaborate a treatment for inequality-constrained based on an active set approach, which regards an inequality function as equality if the constraint values are within some small tolerance. In addition, we devised and tested two resulting algorithms: the standalone HVN method as an efficient local optimizer and a hybridization of the HVN and an MOEA for solving complicated and multi-modal MOPs. Moreover, in detail, we discuss the search direction for dominated points obtained from the set-oriented

Newton step in which we prove that for dominated and infeasible points, the computed search step is the Newton step of the squared equality constraint function. Therefore, our HVN method can efficiently steer the non-dominated and dominated decision points.

We first illustrate the empirical behavior of the standalone algorithm on three simple MOPs, where we observe quadratic convergence of the two-norm of the root finding problem G. Then, on highly multi-modal DTLZ problems with one spherical constraint (Eq-DTLZ), we tested the local convergence of the standalone HVN algorithm with a relatively large approximation set ($\mu = 200$) by initializing the approximation set in the neighborhood around the Pareto set, which shows a fast convergence to well-distributed points on the feasible set. Finally, we benchmark the hybrid algorithm against NSGA-III on Eq-DTLZ1-4 and Eq-IDTLZ1-4 problems, in which we observe that with roughly the same computational budget, the hybrid algorithm achieves substantially more non-dominated points in the final population, which leads to significantly higher hypervolume values. We conjecture that such an advantage is attributed to (1) the fast local convergence of the HVN method and (2) HVN's ability to move infeasible and dominated points.

For future works, we contemplate (1) testing the hybridization of the HVN method with other EMOAs for more than three objectives, e.g., SMS-EMOA, to investigate the benefit of the HVN method in a broader setup; (2) comparing the hybrid HVN method to other state-of-the-art algorithms, e.g., MOEA/D (decomposition-based), EHVI-EGO (Bayesian optimization), or the average Hausdorff distance-based Newton method (mathematical optimization) on complex, or even real-world MOPs with multiple non-linear constraint functions; (3) investigating the analytical expression (as sketched in Figure 1) and computation of the hypervolume Hessian matrix, which can reduce the computation cost of the HVN method; (4) devising generic methodologies to handle inequality constraints for the HVN method, which will make it more applicable in practice; (5) extending the HVN to methods that provide non-zero sub-gradients for dominated points as in [17,18]; and (6) incorporating a surrogate-assisted method for tackling high-dimensional and complex problems, e.g., as in [57].

Author Contributions: Conceptualization, O.S., H.W., A.D., M.E. and V.A.S.H.; methodology, O.S. and H.W.; software, M.E. geometrical analysis, and visualization, H.W. and V.A.S.H.; validation, H.W., M.E. and O.S.; formal analysis, H.W., O.S. and M.E.; investigation, H.W. and O.S.; resources, O.S.; data curation, H.W.; writing—original draft preparation, all; writing—review and editing, all; visualization, H.W.; supervision, O.S. and M.E.; project administration, O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: We have hosted all the data sets of this work on Zenodo: https://doi.org/10.5281/zenodo.7509148.

Acknowledgments: We dedicate this work to Kalyanmoy Deb for his pioneering, inspiring, and fundamental contributions to the evolutionary multi-objective optimization (EMO) community, and particularly for his famous non-dominated sorting procedure, which plays a crucial role in this work in order to efficiently handle dominated points that can be generated throughout the Newton iteration.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stewart, T.; Bandte, O.; Braun, H.; Chakraborti, N.; Ehrgott, M.; Göbelt, M.; Jin, Y.; Nakayama, H. Real-World Applications of Multiobjective Optimization. In *Proceedings of the Multiobjective Optimization, Lecture Notes in Computer Science*; Slowinski, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5252; pp. 285–327.
2. Deb, K. Evolutionary multi-objective optimization: Past, present and future. In Proceedings of the GECCO '20: Proceedings of the 22th annual Conference on Genetic and Evolutionary Computation, Cancún, Mexico, 8–12 July 2020; pp. 343–372.
3. Aguilera-Rueda, V.J.; Cruz-Ramírez, N.; Mezura-Montes, E. Data-Driven Bayesian Network Learning: A Bi-Objective Approach to Address the Bias-Variance Decomposition. *Math. Comput. Appl.* **2020**, *25*, 37. [CrossRef]

4. Frausto-Solis, J.; Hernández-Ramírez, L.; Castilla-Valdez, G.; González-Barbosa, J.J.; Sánchez-Hernández, J.P. Chaotic Multi-Objective Simulated Annealing and Threshold Accepting for Job Shop Scheduling Problem. *Math. Comput. Appl.* **2021**, *26*, 8. [CrossRef]
5. Utz, S.; Wimmer, M.; Hirschberger, M.; Steuer, R.E. Tri-criterion inverse portfolio optimization with application to socially responsible mutual funds. *Eur. J. Oper. Res.* **2014**, *234*, 491–498. [CrossRef]
6. Estrada-Padilla, A.; Lopez-Garcia, D.; Gómez-Santillán, C.; Fraire-Huacuja, H.J.; Cruz-Reyes, L.; Rangel-Valdez, N.; Morales-Rodríguez, M.L. Modeling and Optimizing the Multi-Objective Portfolio Optimization Problem with Trapezoidal Fuzzy Parameters. *Math. Comput. Appl.* **2021**, *26*, 36. [CrossRef]
7. Van Veldhuizen, D.A. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*; Technical report; Air Force Institute of Technology: Kaduna, Nigeria, 1999.
8. Coello, C.A.C.; Cortés, N.C. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genet. Program. Evolvable Mach.* **2005**, *6*, 163–190. [CrossRef]
9. Schütze, O.; Esquivel, X.; Lara, A.; Coello, C.A.C. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2012**, *16*, 504–522. [CrossRef]
10. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. [CrossRef]
11. Auger, A.; Bader, J.; Brockhoff, D.; Zitzler, E. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In Proceedings of the Foundations of Genetic Algorithms, 10th ACM SIGEVO International Workshop, FOGA 2009, Orlando, FL, USA, 9–11 January 2009; Garibay, I.I., Jansen, T., Wiegand, R.P., Wu, A.S., Eds.; ACM: New York, NY, USA, 2009; pp. 87–102. [CrossRef]
12. Beume, N.; Naujoks, B.; Emmerich, M.T.M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **2007**, *181*, 1653–1669. [CrossRef]
13. Bader, J.; Zitzler, E. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evol. Comput.* **2011**, *19*, 45–76. [CrossRef]
14. Schütze, O.; Domínguez-Medina, C.; Cruz-Cortés, N.; de la Fraga, L.G.; Sun, J.Q.; Toscano, G.; Landa, R. A scalar optimization approach for averaged Hausdorff approximations of the Pareto front. *Eng. Optim.* **2016**, *48*, 1593–1617. [CrossRef]
15. Emmerich, M.; Deutz, A.; Beume, N. Gradient-Based/Evolutionary Relay Hybrid for Computing Pareto Front Approximations Maximizing the S-Metric. In *Proceedings of the Hybrid Metaheuristics*; Bartz-Beielstein, T., Blesa Aguilera, M.J., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 140–156.
16. Emmerich, M.; Deutz, A.H. Time Complexity and Zeros of the Hypervolume Indicator Gradient Field. In Proceedings of the EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III EVOLVE 2012, Mexico City, Mexico, 7–9 August 2012; Studies in Computational Intelligence; Schuetze, O., Coello, C.A.C., Tantar, A., Tantar, E., Bouvry, P., Moral, P.D., Legrand, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 500, pp. 169–193. [CrossRef]
17. Wang, H.; Ren, Y.; Deutz, A.; Emmerich, M. On steering dominated points in hypervolume indicator gradient ascent for bi-objective optimization. In *NEO 2015*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 175–203.
18. Deist, T.M.; Maree, S.C.; Alderliesten, T.; Bosman, P.A. Multi-objective optimization by uncrowded hypervolume gradient ascent. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 186–200.
19. Wang, H.; Deutz, A.H.; Bäck, T.; Emmerich, M. Hypervolume Indicator Gradient Ascent Multi-objective Optimization. In Proceedings of the Evolutionary Multi-Criterion Optimization—9th International Conference, EMO 2017, Münster, Germany, 19–22 March 2017; Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M.M., Jin, Y., Grimme, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10173, pp. 654–669. [CrossRef]
20. Sosa Hernández, V.A.; Schütze, O.; Emmerich, M. Hypervolume maximization via set based Newton's method. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 15–28.
21. Sosa-Hernández, V.A.; Schütze, O.; Wang, H.; Deutz, A.H.; Emmerich, M. The Set-Based Hypervolume Newton Method for Bi-Objective Optimization. *IEEE Trans. Cybern.* **2020**, *50*, 2186–2196. [CrossRef] [PubMed]
22. Petersen, K.B.; Pedersen, M.S. The matrix cookbook. *Tech. Univ. Den.* **2008**, *7*, 510.
23. Hillermeier, C. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; Volume 135.
24. Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 12.
25. Klamroth, K.; Tind, J.; Wiecek, M. Unbiased Approximation in Multicriteria Optimization. *Math. Methods Oper. Res.* **2002**, *56*, 413–437. [CrossRef]
26. Fliege, J. Gap-free computation of Pareto-points by quadratic scalarizations. *Math. Methods Oper. Res.* **2004**, *59*, 69–89. [CrossRef]
27. Eichfelder, G. *Adaptive Scalarization Methods in Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 2008.
28. Das, I.; Dennis, J.E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optim.* **1998**, *8*, 631–657. [CrossRef]

29. Fliege, J.; Drummond, L.G.; Svaiter, B.F. Newton's method for multiobjective optimization. *SIAM J. Optim.* **2009**, *20*, 602–626. [CrossRef]
30. Dellnitz, M.; Schütze, O.; Hestermeyer, T. Covering Pareto Sets by Multilevel Subdivision Techniques. *J. Optim. Theory Appl.* **2005**, *124*, 113–155. [CrossRef]
31. Hernández, C.; Naranjani, Y.; Sardahi, Y.; Liang, W.; Schütze, O.; Sun, J.Q. Simple Cell Mapping Method for Multi-objective Optimal Feedback Control Design. *Int. J. Dyn. Control.* **2013**, *1*, 231–238. [CrossRef]
32. Sun, J.Q.; Xiong, F.R.; Schütze, O.; Hernández, C. *Cell Mapping Methods—Algorithmic Approaches and Applications*; Springer: Berlin/Heidelberg, Germany, 2019.
33. Zhang, Q.; Li, H. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
34. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [CrossRef]
35. Jain, H.; Deb, K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Trans. Evol. Comput.* **2014**, *18*, 602–622. [CrossRef]
36. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
37. Ishibuchi, H.; Masuda, H.; Nojima, Y. *A Study on Performance Evaluation Ability of a Modified Inverted Generational Distance Indicator*; Association for Computing Machinery: New York, NY, USA, 2015. [CrossRef]
38. Dilettoso, E.; Rizzo, S.A.; Salerno, N. A Weakly Pareto Compliant Quality Indicator. *Math. Comput. Appl.* **2017**, *22*, 25. [CrossRef]
39. Rudolph, G.; Schütze, O.; Grimme, C.; Domínguez-Medina, C.; Trautmann, H. Optimal averaged Hausdorff archives for bi-objective problems: theoretical and numerical results. *Comput. Optim. Appl.* **2016**, *64*, 589–618. [CrossRef]
40. Bogoya, J.M.; Vargas, A.; Schütze, O. The Averaged Hausdorff Distances in Multi-Objective Optimization: A Review. *Mathematics* **2019**, *7*, 894. [CrossRef]
41. Schütze, O.; Dell'Aere, A.; Dellnitz, M. On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems. In *Proceedings of the Practical Approaches to Multi-Objective Optimization*; Number 04461 in Dagstuhl Seminar Proceedings; Branke, J., Deb, K., Miettinen, K., Steuer, R.E., Eds.; Internationales Begegnungs- und Forschungszentrum (IBFI): Schloss Dagstuhl, Germany, 2005. Available online: http://drops.dagstuhl.de/opus/volltexte/2005/349 (accessed on 1 November 2022).
42. Martin, B.; Goldsztejn, A.; Granvilliers, L.; Jermann, C. On continuation methods for non-linear bi-objective optimization: towards a certified interval-based approach. *J. Glob. Optim.* **2014**, *64*, 3–16. [CrossRef]
43. Martín, A.; Schütze, O. Pareto Tracer: A predictor-corrector method for multi-objective optimization problems. *Eng. Optim.* **2018**, *50*, 516–536. [CrossRef]
44. Schütze, O.; Cuate, O.; Martín, A.; Peitz, S.; Dellnitz, M. Pareto Explorer: A global/local exploration tool for many-objective optimization problems. *Eng. Optim.* **2020**, *52*, 832–855. [CrossRef]
45. Beltrán, F.; Cuate, O.; Schütze, O. The Pareto Tracer for General Inequality Constrained Multi-Objective Optimization Problems. *Math. Comput. Appl.* **2020**, *25*, 80. [CrossRef]
46. Bolten, M.; Doganay, O.T.; Gottschalk, H.; Klamroth, K. Tracing Locally Pareto-Optimal Points by Numerical Integration. *SIAM J. Control. Optim.* **2021**, *59*, 3302–3328. [CrossRef]
47. Zitzler, E.; Thiele, L. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Case Study. In Proceedings of the Parallel Problem Solving from Nature—PPSN V, 5th International Conference, Amsterdam, The Netherlands, 27–30 September 1998; Lecture Notes in Computer Science; Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1498, pp. 292–304. [CrossRef]
48. Emmerich, M.; Yang, K.; Deutz, A.H.; Wang, H.; Fonseca, C.M. A Multicriteria Generalization of Bayesian Global Optimization. In *Advances in Stochastic and Deterministic Global Optimization*; Pardalos, P.M., Zhigljavsky, A., Zilinskas, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 107, pp. 229–242. [CrossRef]
49. DiBenedetto, E.; Debenedetto, E. *Real Analysis*; Springer: Berlin/Heidelberg, Germany, 2002.
50. Paquete, L.; Schulze, B.; Stiglmayr, M.; Lourenço, A.C. Computing representations using hypervolume scalarizations. *Comput. Oper. Res.* **2022**, *137*, 105349. [CrossRef]
51. Margossian, C.C. A Review of Automatic Differentiation and its Efficient Implementation. *WIREs Data Mining Knowl. Discov.* **2019**, *9*, e1305. [CrossRef]
52. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: Berlin/Heidelberg, Germany, 1999. [CrossRef]
53. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic Differentiation in Machine Learning: A Survey. *J. Mach. Learn. Res.* **2017**, *18*, 153:1–153:43.
54. Griewank, A.; Walther, A. *Evaluating Derivatives—Principles and Techniques of Algorithmic Differentiation*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2008. [CrossRef]
55. Cuate, O.; Uribe, L.; Lara, A.; Schütze, O. A benchmark for equality constrained multi-objective optimization. *Swarm Evol. Comput.* **2020**, *52*, 100619. [CrossRef]

56. Cuate, O.; Uribe, L.; Lara, A.; Schütze, O. Dataset on a Benchmark for Equality Constrained Multi-objective Optimization. *Data Brief* **2020**, *29*, 105130. [CrossRef]
57. Fu, C.; Wang, P.; Zhao, L.; Wang, X. A distance correlation-based Kriging modeling method for high-dimensional problems. *Knowl. Based Syst.* **2020**, *206*, 106356. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.