Convergence of Hypervolume-Based Archiving Algorithms II: Competitiveness

Karl Bringmann
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Tobias Friedrich
Max-Planck-Institut für Informatik
Saarbrücken, Germany

ABSTRACT

We study the convergence behavior of $(\mu + \lambda)$-archiving algorithms. A $(\mu + \lambda)$-archiving algorithm defines how to choose in each generation μ children from μ parents and λ offspring together. Archiving algorithms have to choose individuals online without knowing future offspring. Previous studies assumed the offspring generation to be best-case. We assume the initial population and the offspring generation to be worst-case and use the competitive ratio to measure how much smaller hypervolumes an archiving algorithm finds due to not knowing the future in advance. We prove that all archiving algorithms which increase the hypervolume in each step (if they can) are only μ-competitive. We also present a new archiving algorithm which is $(4 + 2/\mu)$-competitive. This algorithm not only achieves a constant competitive ratio, but is also efficiently computable. Both properties provably do not hold for the commonly used greedy archiving algorithms, for example those used in SIBEA, SMS-EMOA, or the generational MO-CMA-ES.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]:
Analysis of Algorithms and Problem Complexity

Keywords

Multiobjective Optimization, Theory,
Performance Measures, Selection

1. INTRODUCTION

As many real-world optimization problems have multiple objectives (like time vs. cost), they have in general no unique optimum, but an often very large (or even infinite) set of incomparable solutions which form the Pareto front. Multi-objective optimizers deal with this by trying to find a small set of trade-off solutions which approximate the Pareto front. They typically keep a bounded archive of μ points (population) in order to capture the output of the search process. In each round they generate λ new points (offspring) by mutation and crossover. The key question is then how to select μ individuals from a larger population. We consider the so-called plus selection strategy, where the next population is chosen out of the λ offspring and μ parents together. We call a specific replacement strategy a $(\mu + \lambda)$-*archiving algorithm* which defines how to choose a new population of μ children from the union of μ parents and λ offspring.

The goal for hypervolume-based multi-objective evolutionary algorithms (MOEAs) is to maximize the hypervolume indicator of the output population, which is the volume of the dominated portion of the objective space (see Section 2 for a formal definition). For this type of MOEA, two archiving algorithms are known in the literature:

- A *locally optimal* archiving algorithm returns a subset of μ points from the given $\mu + \lambda$ points such that the hypervolume indicator is maximized.
- A *greedy* archiving algorithm deletes a point such that the hypervolume of the remaining points is maximal. This is repeated until only μ points are left.

We prove in this paper that all locally optimal and all greedy archiving algorithms are NP-hard (cf. Theorem 3.1 and Observation 2.8) and therefore not efficiently computable unless P=NP. Many hypervolume based algorithms like SIBEA [12], SMS-EMOA [1], or the generational MO-CMA-ES [8] use greedy archiving algorithms. As locally optimal algorithms have to choose the best out of a large number, $\binom{\mu+\lambda}{\mu}$, of subsets of the given points, they are generally considered to be computationally infeasible. Note that a locally optimal archiving algorithm in general does not maximize the hypervolume over multiple generations. However, it still seems to have superior theoretical properties: It has long been known that the resulting point sets of both algorithms differ [3], and that the deleted hypervolume (the contribution of the deleted points) can even be arbitrarily larger for greedy archiving algorithms compared to locally optimal algorithms [5].

We want to study the intrinsic limitations of and the potential provided by hypervolume-based archiving algorithms. Beyond the smaller classes of locally optimal and greedy archiving algorithms we thus also consider the following two natural classes of archiving algorithms:

- A *non-decreasing* archiving algorithm chooses the population of children such that the dominated hypervolume does not decrease compared to the parent generation.
- An *increasing* archiving algorithm chooses the population of children such that the dominated hypervolume increases compared to the parent generation, unless there is no subset of population and offspring with larger dominated hypervolume.

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

Both are intuitively desirable properties for hypervolume-based archiving algorithms. We will see that there are algorithms which are non-decreasing, but not increasing (cf. Algorithms 3 and 4). Moreover, we prove that both classes significantly differ. There are non-decreasing archiving algorithms which are better and faster than all increasing archiving algorithms (see Sections 1.2 and 1.3 for more detailed statements).

To rigorously study the impact of archiving algorithms on convergence, we cannot concentrate only on single iterations, but have to consider multiple generations of populations. We model this long run behavior with the initial population being worst-case input to the archiving algorithm, followed by some kind of offspring generation, and we then ask whether we arrive at a population with large hypervolume. There are two natural assumptions on the offspring generation: best-case and worst-case. Each assumption corresponds to one of the following result sections.

1.1 Known results on effectiveness

All previous work in this setting [7, 10, 13] assumes a best-case perspective on the offspring generation. This means that we ask whether, for each population, there exists a sequence of offspring sets such that the archiving algorithm ends up in a population maximizing the hypervolume. This was formalized by Bringmann and Friedrich [7] with the notion of effectiveness: An archiving algorithm is *effective* if there is a sequence of offspring such that the algorithm reaches an optimum. Zitzler et al. [13] proved that all non-decreasing $(\mu + 1)$-archiving strategies are ineffective and all locally optimal $(\mu + \mu)$-archiving strategies are effective. Bringmann and Friedrich [7] proved that all non-decreasing $(\mu + \lambda)$-archiving strategies are ineffective for $\lambda < \mu$.

In order to measure how close to an optimal set the best reachable sets for $\lambda < \mu$ are, we call an archiving algorithm α-*approximate* if it can always reach a set with a hypervolume at least $1/\alpha$ times the largest possible hypervolume. Bringmann and Friedrich [7] proved that no non-decreasing $(\mu + \lambda)$-archiving algorithm can be better than $\left(1 + 0.1338\left(\frac{1}{\lambda} - \frac{1}{\mu}\right)\right)$-approximate. This bound can be tightened if the hypervolume is defined relative to a reference set instead of a single reference point. For this more general definition of the hypervolume indicator, Ulrich and Thiele [10] showed a lower bound of $1 + \frac{1}{2\lambda}$ for $\lambda < \mu$.

On the other hand, Bringmann and Friedrich [7] showed that every increasing $(\mu + \lambda)$-archiving algorithm reaches a 2-approximation. Ulrich and Thiele [10] showed that every increasing $(\mu + 1)$-archiving algorithm reaches a $\left(2 - \frac{1}{\mu}\right)$-approximation. More precisely, they showed that every non-decreasing $(\mu + \lambda)$-archiving algorithm reaches a $\left(2 - \frac{\lambda - p}{\mu}\right)$-approximation, where $\mu = q\,\lambda - p$ with $p < \lambda$ and $p, q \in \mathbb{N}_{\geqslant 0}$. Note that in these results we omitted all summands and factors of arbitrarily small $\varepsilon > 0$.

1.2 New results on competitiveness

In this paper we assume a worst-case perspective on the offspring generation. This corresponds to the well-known concept of *competitive analysis*. It has already been observed that archiving algorithms fit nicely in this classical theory developed for online algorithms [2]. López-Ibáñez, Knowles, and Laumanns [9, p. 59] suggested it as an open problem "to use competitive analysis techniques from the field of online algorithms to obtain worst-case bounds, in terms of a measure of 'regret' for archivers."

We consider the initial population and offspring as worst-case input and ask again how large a hypervolume we can get. In this case, however, the adversary, who selects the offspring, can limit the search to a very small part of the search space, and it is therefore impossible in general to reach the optimum hypervolume. This motivates the following definition. We say an archiving algorithm is α-*competitive* if for all initial populations and offspring it reaches a hypervolume which is only a factor $1/\alpha$ smaller than the hypervolume of the *best* μ *points seen* (cf. Definition 2.9).

On the negative side, we prove that all increasing archiving algorithms are exactly μ-competitive (see Theorem 3.2 for the upper bound and Theorem 3.3 for the lower bound). This means that there is a sequence of offspring such that the hypervolume of the μ individuals chosen iteratively by an algorithm which maximizes the hypervolume in each step is μ times larger than the maximum hypervolume achievable by another choice of μ individuals. This result seems to suggest that the notion of competitiveness does not help in comparing archiving algorithms, as all reasonable archiving algorithms meet exactly the same bound.

However, on the positive side, we are able to design an archiving algorithm that is $4 + 2/\mu$-competitive (cf. Theorem 4.1), which implies a constant competitive ratio compared to the unbounded ratio of μ. It is a non-decreasing archiving algorithm which is not increasing, i.e., there are populations and offspring where we stay with the current population, although the offspring allows an increase in hypervolume. This proves that significantly better competitive ratios can be achieved for archiving algorithms which are not increasing compared to the typically used increasing archiving algorithms. The algorithm works as follows (for details see Algorithm 3): It adds offspring one by one to the current population. Considering the population and an offspring, we compute the hypervolume for exchanging the offspring with any other point in the population. We take the best exchange only if it increases the population's hypervolume by at least a certain minimal factor.

1.3 New results on computational efficiency

We prove that all increasing archiving algorithms are NP-hard to compute (cf. Theorem 3.1). This also implies that all common greedy archiving algorithms are not efficiently computable unless P=NP (cf. Observation 2.8). This still allows archiving algorithms which are not increasing to be efficiently computable. Indeed, we also prove that a randomized variant of our aforementioned archiving algorithm can be made to run efficiently (cf. Theorem 4.3). Note that this is in sharp contrast to the large set of increasing archiving algorithms, which all have a worse competitive ratio (cf. Theorem 3.3) and a worse computational complexity (cf. Theorem 3.1) compared to the proposed new archiving algorithm. The underlying reason why the new algorithm can beat all increasing archiving algorithms is that approximating the hypervolume is tractable [4] and for the new algorithm it is sufficient only to approximate the hypervolume, as it checks only for constant factor increases. Although the new archiving algorithm might not be used as-is in any practical MOEAs, it is a proof of concept that there are efficiently computable archiving algorithms which can beat the competitive ratio of the thus far typically used locally optimal and greedy archiving algorithms.

The outline is as follows. In Section 2 we introduce the basic concepts and notation. Section 3 presents the tight bound on the competitiveness of increasing archiving algorithms and proves that all of them are NP-hard. We propose a non-decreasing algorithm with constant competitiveness in Section 4, and show that it can even be computed efficiently.

2. PRELIMINARIES

This section formally introduces all necessary notation. The two most fundamental concepts are the hypervolume indicator (Section 2.1) and archiving algorithms (Section 2.2). The combination

of both, i.e. hypervolume-based archiving algorithms, are introduced in Section 2.3. These first three parts of this section closely follow [7]. Our new quality measure of competitiveness is then defined in Section 2.4.

We consider maximization problems with vector-valued objective functions $f\colon \mathcal{X} \to \mathbb{R}^d$, where $\mathcal{X}$ denotes an arbitrary search space. The feasible points $\mathcal{Y} := f(\mathcal{X})$ are called the objective space. Consider the following abstract framework of a MOEA:

Algorithm 1: General $(\mu+\lambda)$-MOEA

1 $P^0 \leftarrow$ initialize with μ individuals
2 **for** $i \leftarrow 1$ **to** N **do**
3 $\quad Q^i \leftarrow$ generate λ offspring
4 $\quad P^i \leftarrow$ select μ individuals from $P^{i-1} \cup Q^i$

We make no assumptions about the specific search space $\mathcal{X}$. We assume that both the initialization (cf. line 1 of Algorithm 1) and the offspring generation (cf. line 3 of Algorithm 1) are worst-case. Our main concern is how the population of children is chosen (cf. line 4 of Algorithm 1). We will formally define and discuss different *archiving algorithms* in Sections 2.2 and 2.3.

We use the terms archive and population synonymously for the set of current solutions P^i of Algorithm 1. In concrete MOEAs, populations are subsets of the search space. As we do not want to assume any structural properties of the search space, we abstract from the search space and will *only work on the objective space* $\mathcal{Y} \subseteq \mathbb{R}^d$ in the remainder. We therefore also identify individuals with points in the d-dimensional Euclidean space.

DEFINITION 2.1. *A population P is a finite multiset and a subset of $\mathbb{R}^d$. If an objective space $\mathcal{Y} \subseteq \mathbb{R}^d$ is fixed, we require $P \subseteq \mathcal{Y}$. We call P a μ-population if $|P| \leqslant \mu$.*

2.1 Hypervolume indicator

The hypervolume indicator $\mathrm{HYP}(P)$ [11] of a population P is the volume of the union of regions of the objective space which are dominated by P and bounded by a reference point R. Here domination refers to the following dominance relation for points in the objective space $\mathcal{Y} \subseteq \mathbb{R}^d$:

$$(x_1, x_2, \ldots, x_d) \preceq (y_1, y_2, \ldots, y_d) \quad \text{iff } x_1 \leqslant y_1, x_2 \leqslant y_2, \ldots, \text{and } x_d \leqslant y_d.$$

Formally, the hypervolume $\mathrm{HYP}(P)$ of a population P is defined as

$$\mathrm{HYP}(P) := \int_{\mathbb{R}^d} A_P(x)\, dx$$

where the attainment function $A_P\colon \mathbb{R}^d \to \{0,1\}$ is an indicator function on the objective space that describes the space above the reference point R which is dominated by P, that is, $A_P(x) = 1$ if $R \preceq x$ and there is a $p \in P$ such that $x \preceq p$, and $A_P(x) = 0$ otherwise.

We fix the reference point w.l.o.g. to $R = 0^d$, since translations do not change any of our results. This means that the reference point is globally fixed and known to the archiving algorithm.

The aim of a hypervolume-based MOEA is finding a set P^* of size μ which maximizes the hypervolume, that is,

$$\mathrm{HYP}(P^*) = \mathrm{maxHYP}_\mu(\mathcal{Y})$$

where we define for all $Y \subset \mathbb{R}^d$,

$$\mathrm{maxHYP}_\mu(Y) := \sup_{\substack{P \subseteq Y \\ |P| \leqslant \mu}} \mathrm{HYP}(P).$$

In the remainder of the paper, the set Y will often be finite. In these cases, the supremum in the definition of $\mathrm{maxHYP}_\mu(Y)$ becomes a maximum. However, for infinite sets the supremum is necessary in general.

The *contribution* of a point p to a population P is

$$\mathrm{CON}_P(p) := \mathrm{HYP}(P) - \mathrm{HYP}(P - p),$$

where we use the notation $P - p$ for $P \setminus \{p\}$. We also use $P + p$ to shortcut $P \cup \{p\}$ throughout the paper.

Note that according to the definition of $\mathrm{CON}_P(p)$, the contributing hypervolume of a dominated individual is zero.

2.2 Archiving algorithms

We now specify more formally how to choose the μ offsprings in line 4 of Algorithm 1. For this, we consider the following general framework of an archiving algorithm.

Algorithm 2: General $(\mu+\lambda)$-archiving algorithm

input : μ-population P, λ-population Q
output: μ-population P' with $P' \subseteq P \cup Q$

Note that any $(\mu+\lambda)$-archiving algorithm is also a $(\mu+\lambda')$-archiving algorithm for any $\lambda' < \lambda$, as we then allow only a subset of the inputs, namely with smaller offspring population Q. We do not make any assumptions on the runtime of an archiving algorithm. In fact, as hypervolume computation is #P-hard [4], most hypervolume-based archiving algorithms are not computable in polynomial time in the number of objectives d. We will use the following notation to describe an archiving algorithm.

DEFINITION 2.2. *A $(\mu+\lambda)$-archiving algorithm $\mathcal{A}$ is a partial mapping $\mathcal{A}\colon 2^{\mathbb{R}^d} \times 2^{\mathbb{R}^d} \mapsto 2^{\mathbb{R}^d}$ such that for a μ-population P and a λ-population Q, $\mathcal{A}(P,Q)$ is a μ-population and $\mathcal{A}(P,Q) \subseteq P \cup Q$.*

For convenience, we sometimes drop the prefix $(\mu+\lambda)$ and just refer to an archiving algorithm (or even shorter: *algorithm*) without specifying μ and λ. With this notation, we can now formally describe the generation process of Algorithm 1 as follows.

DEFINITION 2.3. *Let P^0 be a μ-population and $Q^1, \ldots, Q^N$ a sequence of λ-populations. Then*

$$P^i := \mathcal{A}(P^{i-1}, Q^i) \qquad \textit{for all } i = 1, \ldots, N.$$

With slight abuse of notation we also set

$$\mathcal{A}(P^0, Q^1, \ldots, Q^i) := P^i \qquad \textit{for all } i = 1, \ldots, N.$$

2.3 Hypervolume-based archiving algorithms

We now specify three classes of hypervolume-based archiving algorithms. The first one only requires the archiving algorithms to never return a solution with a smaller hypervolume:

DEFINITION 2.4. *A $(\mu+\lambda)$-archiving algorithm $\mathcal{A}$ is* non-decreasing*, if for all inputs P and Q we have*

$$\mathrm{HYP}(\mathcal{A}(P,Q)) \geqslant \mathrm{HYP}(P).$$

All reasonable hypervolume-based archiving algorithms are non-decreasing. However, the class also contains ineffective algorithms like the algorithm which always returns P.

The second, slightly smaller class of hypervolume-based archiving algorithms is defined as follows.

DEFINITION 2.5. *A $(\mu+\lambda)$-archiving algorithm $\mathcal{A}$ is* increasing*, if it is non-decreasing and for all inputs P and Q with*

$$\text{maxHYP}_\mu(P \cup Q) > \text{HYP}(P)$$

we have

$$\text{HYP}(\mathcal{A}(P,Q)) > \text{HYP}(P).$$

Moreover, we define locally optimal and greedy archiving algorithms. Note that for both classes there are more than one archiving algorithm fulfilling the respective definition, as ties may be broken arbitrarily.

DEFINITION 2.6. *A $(\mu+\lambda)$-archiving algorithm $\mathcal{A}$ is* locally optimal*, if for all inputs P and Q we have*

$$\text{HYP}(\mathcal{A}(P,Q)) = \text{maxHYP}_\mu(P \cup Q).$$

DEFINITION 2.7. *A $(\mu+\lambda)$-archiving algorithm $\mathcal{A}$ is* greedy*, if there are functions $\mathcal{A}'(P) = P \setminus \{\text{argmin}_{p\in P} \text{CON}_P(p)\}$ such that for all inputs P and Q we have*

$$\mathcal{A}(P,Q) = \underbrace{\mathcal{A}' \circ \ldots \circ \mathcal{A}'}_{\lambda \text{ times}}(P \cup Q).$$

The rest of the paper focuses on increasing and non-decreasing archiving algorithms. Their relation to locally optimal and greedy archiving algorithms is as follows.

OBSERVATION 2.8. *Greedy $(\mu+1)$-archiving algorithms and locally-optimal $(\mu+\lambda)$-archiving algorithms are increasing archiving algorithms. Greedy $(\mu+\lambda)$-archiving algorithms are not necessarily non-decreasing archiving algorithms for $\lambda > 1$.*

This observation allows us to translate all forthcoming bounds for increasing archiving algorithms to locally-optimal archiving algorithms. Observation 2.8 also implies that all our negative results hold for greedy algorithms, since the computational hardness (cf. Theorem 3.1) and the lower bound for the competitive ratio (cf. Theorem 3.2) apply to the restriction of a greedy algorithm to $\lambda = 1$ and, thus, also to greedy algorithms in general. There is only one result for increasing algorithms which does not directly apply to greedy algorithms, which is the upper bound for the competitive ratio (cf. Theorem 3.3). As we are more interested in the lower bound for the competitive ratio of greedy algorithms, we do not prove an additional upper bound, but conjecture that the bound is still tight for greedy archiving algorithms for arbitrary λ.

2.4 Competitiveness

We take a worst-case view on the initial population as well as offspring generation, so we want to bound $\mathcal{A}(P^0, Q^1, \ldots, Q^N)$ for any initial population P^0 and any offspring generations $Q^1, \ldots, Q^N$. Observe that in this setting all results have to be independent of the specific objective space $\mathcal{Y}$ and we cannot hope to reach $\text{maxHYP}_\mu(\mathcal{Y})$ in general. The only aim can be achieving a hypervolume as good as the maximum hypervolume among all μ-populations which are subsets of the points we have seen so far, that is,

$$\text{maxHYP}_\mu\left(P^0 \cup \bigcup_{i=1}^{N} Q^i\right),$$

which can be arbitrarily smaller than $\text{maxHYP}_\mu(\mathcal{Y})$.

This allows us to view archiving algorithms as an online problem where the algorithm is fed with new offspring in a serial fashion and has to decide which individual it should keep in the population without knowing the entire input. To measure the 'regret' of an archiving algorithm we define its competitive ratio α as follows.

DEFINITION 2.9. *Let P^0 be a μ-population and Q^i be λ-populations for $1 \leqslant i \leqslant N$. Then $I := (P^0, Q^1, \ldots, Q^N)$ is an* instance*. We also set*

$$\text{Obs}(I) := P^0 \cup \bigcup_{i=1}^{N} Q^i.$$

An archiving algorithm $\mathcal{A}$ is α-competitive (for some $\alpha \geqslant 1$) if for all instances $I = (P^0, Q^1, \ldots, Q^N)$ we have

$$\mathcal{A}(P^0, Q^1, \ldots, Q^N) \geqslant \frac{1}{\alpha} \text{maxHYP}_\mu(\text{Obs}(I)).$$

3. INCREASING ALGORITHMS

We first study the large class of increasing archiving algorithms, which also includes locally optimal and greedy archiving algorithms (cf. Observation 2.8). We prove that all increasing archiving algorithms are NP-hard to compute (Section 3.1) and only achieve a competitive ratio of μ (Section 3.2).

3.1 Increasing algorithms are inefficient

By reduction from the known hardness of computing a least contributor of a set of points, we show the following theorem.

THEOREM 3.1. *All increasing archiving algorithm are NP-hard to compute (in $\mu + d$).*

Proof. We reduce from the problem of computing a least contributor of a set of points: Given $P \subseteq \mathbb{R}^d$ of size n, compute a point $p \in P$ with $\text{CON}_P(p)$ minimal (see Section 2.1 for the definition). This problem is NP-hard according to [6].

Let P be an instance to the least contributor problem, and let $\mathcal{A}$ be an efficiently computable increasing archiving algorithm. We compute $A(p) := P \setminus \mathcal{A}(P-p, \{p\})$ for each $p \in P$. This is the point with which the archiving algorithm $\mathcal{A}$ exchanges p given population $P - p$ and offspring $\{p\}$.

Consider the graph with vertex set P and directed edges $(p, A(p))$ for each $p \in P$. This graph may have self-loops. It includes a directed cycle as a subgraph: Starting at any point and always following the unique out-edge we will at some point see an already visited point again; this means we traversed a cycle (after some initial path).

Let $(p_0, \ldots, p_{k-1})$ be such a cycle. It can have length $k = 1$, if the cycle is a self-loop. Since $A(p_i) = p_{i+1}$ (with indices modulo k) and the archiving algorithm is increasing — thus also non-decreasing — we have $\text{HYP}(P - p_{i+1}) \geqslant \text{HYP}(P - p_i)$ for all $i \in \{0, \ldots, k-1\}$. Hence, all $\text{HYP}(P - p_i)$ are equal; in particular $\text{HYP}(P - p_0) = \text{HYP}(P - A(p_0))$. Since the archiving algorithm is increasing, this means that no increase was possible given population $P - p_0$ and offspring $\{p_0\}$, and, hence, that $\text{HYP}(P - p_0) = \text{HYP}(P) - \text{CON}_P(p_0)$ is maximal among all $\text{HYP}(P - p)$. In other words, $\text{CON}_P(p_0)$ is minimal and p_0 is a least contributor. The same holds for all other points $p \in P$ that lie on a directed cycle in the constructed graph. Thus, we can compute a least contributor efficiently. □

3.2 Increasing algorithms are not competitive

It is easy to show an upper bound on the competitive ratio of μ for a very large class of archiving algorithms. It applies to all non-decreasing archiving algorithms with the following property: If a single offspring point $q \in Q$ alone dominates a larger hypervolume than all points in the current population together, then the algorithm should take this point q (or do something even better). Note that all increasing — and thus also all locally optimal — archiving algorithms satisfy this condition.

THEOREM 3.2. *Let $\mathcal{A}$ be a non-decreasing archiving algorithm such that for all inputs P and Q and points $q \in Q$ we have*

$$\mathrm{HYP}(\mathcal{A}(P,Q)) \geqslant \mathrm{HYP}(\{q\}).$$

Then $\mathcal{A}$ is μ-competitive.

In particular: All increasing $(\mu+\lambda)$-archiving algorithms are μ-competitive.

Proof. Let $I := (P^0, Q^1, \ldots, Q^N)$ be an instance and $P^* \subseteq \mathrm{Obs}(I)$ be a μ-population with $\mathrm{HYP}(P^*) = \max\mathrm{HYP}_\mu(\mathrm{Obs}(I))$. We have

$$\mathrm{HYP}(P^*) \leqslant \sum_{p\in P^*} \mathrm{HYP}(\{p\}),$$

so there is a point $p^* \in P^*$ with

$$\mathrm{HYP}(\{p^*\}) \geqslant \frac{1}{|P|}\mathrm{HYP}(P^*) \geqslant \frac{1}{\mu}\mathrm{HYP}(P^*).$$

Either $p^* \in P^0$ (in which case we set $r := 0$) or $p^* \in Q^r$ for some r. Consider $P^i = \mathcal{A}(P^0, Q^1, \ldots, Q^i)$. We have $\mathrm{HYP}(P^r) \geqslant \mathrm{HYP}(\{p^*\})$ by the extra assumption and $\mathrm{HYP}(P^r) \leqslant \mathrm{HYP}(P^{r+1}) \leqslant \ldots \leqslant \mathrm{HYP}(P^N)$ by $\mathcal{A}$ being non-decreasing. Taken together these prove the claim. □

Observe that even a very simple algorithm fulfills the premises of the above theorem: It considers the offspring one-by-one and replaces its current population P^i with $\{q\}$, if the offspring point q has greater hypervolume than P^i. This requires only one costly computation of $\mathrm{HYP}(P^0)$; all other populations consist of only a single point in the objective space.

Perhaps surprisingly, no increasing archiving algorithm is better than this simple algorithm in the worst case: For them, the bound of Theorem 3.2 is tight.

THEOREM 3.3. *No increasing $(\mu+\lambda)$-archiving algorithm is $(\mu-\varepsilon)$-competitive for any $\varepsilon > 0$.*

Proof. We construct an instance $I = (P^0, Q^1, \ldots, Q^N)$ as follows. For reals $a, A > 0$ to be chosen later and $j \in \{1, \ldots, \mu-1\}$, we set

$$p_j = (A + j\,a, (\mu-j)a),$$

and $B := (\mu-1)a$. Moreover, for $\delta, \rho > 0$, $\rho < 1$ to be chosen later and $i \in \{0, \ldots, N\}$, we set $q_i = (x_i, y_i)$ with

$$x_i := A\rho^i,$$
$$y_i := B + \frac{1+\delta\,i}{x_i}.$$

These points are depicted in figure Figure 1. Setting $P^0 := \{q_0, p_1, \ldots, p_{\mu-1}\}$ and $Q^i := \{q_i\}$ (or λ copies of q_i), we get an instance I.

We show that $\mathcal{A}(P^{i-1}, Q^i) = P^i$ with $P^i = \{q_i, p_1, \ldots, p_{\mu-1}\}$ for $\mathcal{A}$ being an increasing archiving algorithm. To do this, we have to show that the exchange of q_{i-1} with q_i is increases the hypervolume and is the only increasing exchange. This means we have to show $\mathrm{HYP}(P^i) > \mathrm{HYP}(P^{i-1})$ and $\mathrm{HYP}(P^{i-1}) \geqslant \mathrm{HYP}(P^{i-1} + q_i - p_j)$ for any $1 \leqslant j \leqslant \mu-1$, as those are the only possible exchanges. We have

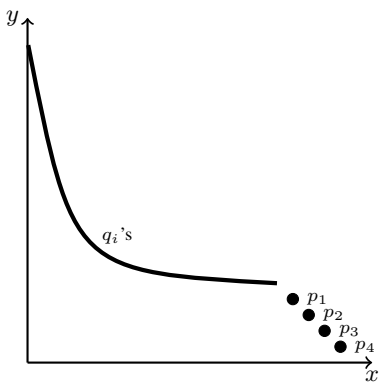

Figure 1: Illustration of the example used in the proof of Theorem 3.3.

$$\mathrm{HYP}(P^i) = \mathrm{HYP}(\{p_1, \ldots, p_{\mu-1}\}) + \mathrm{CON}_{P^i}(q_i),$$

where $\mathrm{CON}_{P^i}(q_i) = x_i(y_i - B) = 1 + \delta i$, and $p_1, \ldots, p_{\mu-1}$ are collinear points, so that their hypervolume can be easily calculated, yielding

$$\mathrm{HYP}(P^i) = A\,B + \tbinom{\mu}{2}a^2 + 1 + \delta\,i. \tag{1}$$

This gives $\mathrm{HYP}(P^i) > \mathrm{HYP}(P^{i-1})$ right away. Moreover, we have

$$\begin{aligned}&\mathrm{HYP}(P^{i-1} + q_i - p_j)\\ &= \mathrm{HYP}(P^{i-1}) - \mathrm{CON}_{P^{i-1}}(p_j) + \mathrm{CON}_{P^{i-1}+q_i}(q_i),\end{aligned}$$

with $\mathrm{CON}_{P^{i-1}}(p_j) \geqslant a^2$ and

$$\begin{aligned}\mathrm{CON}_{P^{i-1}+q_i}(q_i) &= x_i(y_i - y_{i-1})\\ &= 1 + \delta\,i - \rho(1+\delta(i-1))\\ &= (1-\rho)(1+\delta(i-1)) + \delta.\end{aligned}$$

Hence, for

$$a^2 \geqslant (1-\rho)(1+\delta\,N) + \delta, \tag{2}$$

we have $\mathrm{HYP}(P^{i-1}) \geqslant \mathrm{HYP}(P^{i-1} + q_i - p_j)$ for any $1 \leqslant j \leqslant \mu-1$, and $P^i = \mathcal{A}(P^{i-1}, Q^i)$ indeed holds.

Lastly, we need a lower bound on $\max\mathrm{HYP}_\mu(\mathrm{Obs}(I))$. For this we require that μ divides N and consider $P = \{q_{iN/\mu} \mid 0 \leqslant i \leqslant \mu-1\}$. If we consider instead $P' = \{q'_{iN/\mu} \mid 0 \leqslant i \leqslant \mu-1\}$, with $q'_i = (x_i, y'_i)$, $y'_i = B + 1/x_i$, the hypervolume decreases only, as we decrease the y-coordinates. Hence, we have $\max\mathrm{HYP}_\mu(\mathrm{Obs}(I)) \geqslant \mathrm{HYP}(P) \geqslant \mathrm{HYP}(P')$. All that is left to show is

$$\mathrm{HYP}(P') > (\mu-\varepsilon)\mathrm{HYP}(P^N). \tag{3}$$

We compute $\mathrm{HYP}(P')$ to be

$$\begin{aligned}\mathrm{HYP}(P') &= \mathrm{HYP}(\{q'_0\}) + \sum_{i=1}^{\mu-1} x_{iN/\mu}(y'_{iN/\mu} - y'_{(i-1)N/\mu}) \\ &= A(B + 1/A) + \sum_{i=1}^{\mu-1}(1-\rho^{N/\mu}) \\ &= A\,B + \mu - (\mu-1)\rho^{N/\mu}. \qquad (4)\end{aligned}$$

Then equation (3) is fulfilled (using equations (1) and (4)) if

$$\begin{aligned}&A\,B + \mu - (\mu-1)\rho^{N/\mu} > \\ &(\mu-\varepsilon)\left(A\,B + \tbinom{\mu}{2}a^2 + 1 + \delta\,N\right).\end{aligned}$$

Rearranging this, we get

$$\begin{aligned}\varepsilon >& (\mu - 1 - \varepsilon)A\,B + (\mu-\varepsilon)\tbinom{\mu}{2}a^2 + \\ &(\mu-1)\rho^{N/\mu} + (\mu-\varepsilon)\delta\,N.\end{aligned}$$

This inequality is fulfilled by setting $A := \varepsilon/(4\mu B)$, $a^2 := \varepsilon/(4\mu^3)$, $\delta := \min\{\varepsilon/(4\mu N), a^2/2\}$, and $N := \mu\lceil\log_\rho(\varepsilon/(4\mu))\rceil$. As we can assume $\varepsilon \leqslant 1$, we have $\delta \leqslant \min\{1/N, a^2/2\}$, so that requirement (2) can be simplified to $a^2 \geqslant 2(1-\rho) + a^2/2$. We set $\rho := 1 - a^2/4$ to satisfy it. Noting that there is no cyclic dependence in these definitions, we conclude the proof. □

In the proof of Theorem 3.3, we explicitly construct a bad 2-dimensional instance; see Figure 1 for an example with $\mu = 5$. The initial population consists of the points $p_1, \ldots, p_4$ and the rightmost point of the q_i's. Then every offspring consists of (λ copies of) a q_i slightly to the left and above the old one, so that any increasing algorithm has to exchange the two points. This way, the population will always consist of the points $p_1, \ldots, p_4$ and one of the q_i's, with the latter point being dragged to the left. The optimal population, however, consists of 5 nicely spaced q_i's, which has (by choosing the free parameters correctly) a hypervolume that is nearly a factor μ larger than the hypervolume of the population of the increasing algorithm.

4. NON-DECREASING ALGORITHMS

Section 3 showed that increasing archiving algorithms are NP-hard to compute and only achieve an unbounded competitive ratio of μ. We now present a non-decreasing (and not increasing) archiving algorithm which achieves constant competitiveness (cf. Section 4.1). Afterwards, we present a randomized variant of the algorithm which is also efficiently computable (cf. Section 4.2).

4.1 A competitive non-decreasing algorithm

The results of Section 3.2 show that all increasing archiving algorithms have a very bad competitive ratio, and, even worse, that the notion of competitiveness is not suited for comparing different increasing archiving algorithms: All of them have the same competitiveness. This leaves open whether there are non-decreasing, but not increasing archiving algorithms with, say, a constant competitive ratio. This would imply that some archiving algorithm, that is not locally optimal, achieves a better competitive ratio than all locally optimal archiving algorithms. We show that this is indeed the case.

THEOREM 4.1. *There is a $(4 + 2/\mu)$-competitive non-decreasing $(\mu+1)$-archiving algorithm.*

Note that we can easily build a $(\mu+\lambda)$-archiving algorithm with the same competitiveness from the $(\mu+1)$-archiving algorithm guaranteed by Theorem 4.1 by feeding the λ offspring one by one to the $(\mu+1)$-archiving algorithm.

We do not prove Theorem 4.1 directly, as it follows from the proof of Theorem 4.3 below. However, one such archiving algorithm $\mathcal{A}_{\text{comp}}$ is given in Algorithm 3. This non-locally-optimal algorithm improves on the locally optimal algorithms with respect to the competitive ratio and hence is is better suited for hypervolume-based selection in the worst case. Note that this does not imply that this algorithm should be used in practice, as worst-case optimality is usually not needed.

Algorithm 3: Competitive $(\mu+1)$-archiving algorithm $\mathcal{A}_{\text{comp}}$

input : μ-population P, offspring $\{q\}$
output: μ-population P'

1 **foreach** $p \in P + q$ **do**
2 $H_p \leftarrow \mathrm{HYP}(P + q - p)$
3 $p' \leftarrow \operatorname{argmax}\{H_p \mid p \in P\}$
4 **if** $H_{p'} > (1 + 1/\mu)\,H_q$ **then**
5 **return** $P + q - p'$
6 **else**
7 **return** P

Unfortunately, the runtime of $\mathcal{A}_{\text{comp}}$ cannot be polynomial in $\mu + d$ (unless P = NP) as the exact hypervolume calculation in line 2 of Algorithm 3 is #P-hard [4]. However, this also holds for all increasing archiving algorithms as shown in Theorem 3.1.

4.2 An efficient randomized competitive non-decreasing archiving algorithm

We now propose a randomized variant of $\mathcal{A}_{\text{comp}}$ which improves on all increasing algorithms not only with respect to the competitive ratio, but also the runtime. It is a randomized algorithm which meets the competitive ratio bound only with a certain high probability. Hence, we need to redefine competitiveness to include randomized algorithms.

DEFINITION 4.2. *Let $\alpha \geqslant 1$ and $p\colon \mathbb{N} \to [0,1]$. A randomized archiving algorithm $\mathcal{A}$ is α-competitive with probability p if for all instances $I = (P^0, Q^1, \ldots, Q^N)$ we have*

$$\mathcal{A}(P^0, Q^1, \ldots, Q^N) \geqslant \frac{1}{\alpha}\,\mathrm{maxHYP}_\mu(\mathrm{Obs}(I))$$

with probability $\geqslant p(N)$.

Our proposed algorithm $\mathcal{A}_{\text{eff}}$ is given in Algorithm 4. It takes additional parameters $\varepsilon, \delta > 0$ and is $(4 + 2/\mu + \varepsilon)$-competitive with probability $p(N) = 1 - N\delta$.

The critical feature of $\mathcal{A}_{\text{eff}}$ is line 4. It makes use of the hypervolume approximation scheme of Bringmann and Friedrich [4] which computes with probability at least $1-\delta$ a $(1+\varepsilon)$-approximation of the hypervolume of a given set of μ points in $\mathbb{R}^d$ in time $\mathcal{O}(\log(1/\delta)\mu d/\varepsilon^2)$. It is clear from the hypervolume approximation algorithm used here that $\mathcal{A}_{\text{eff}}$ is efficiently computable, namely with a run-time of at most $\mathcal{O}(\log(\mu/\delta)\mu^4 d/\varepsilon^2)$. Note that we aimed only for a polynomial runtime and did not try to optimize the algorithm for a better runtime bound.

We can prove that $\mathcal{A}_{\text{eff}}$ is competitive. The following theorem states our result.

Algorithm 4: Randomized Efficient Competitive $(\mu+1)$-archiving algorithm $\mathcal{A}_{\text{eff}}$

input : μ-population P, offspring $\{q\}$, error bound ε, error probability δ
output: μ-population P'

$\varepsilon' \leftarrow \varepsilon/104$
$c \leftarrow 1 + 2\varepsilon'$
foreach $p \in P + q$ **do**
  $H_p \leftarrow$ compute $(1+\varepsilon'/\mu)$-approximation of $\mathrm{HYP}(P+q-p)$ with error probability $\delta/(\mu+1)$
$p' \leftarrow \operatorname{argmax}\{H_p \mid p \in P\}$
if $H_{p'} > (1+\frac{c}{\mu})\, H_q$ **then**
  return $P + q - p'$
else
  return P

THEOREM 4.3. *Let $0 < \varepsilon \leqslant 1$. Algorithm $\mathcal{A}_{\text{eff}}$ is a randomized $(\mu+1)$-archiving algorithm which is non-decreasing and $(4+2/\mu+\varepsilon)$-competitive with probability $p(N) = 1 - N\delta$ and has a deterministic runtime polynomial in μ, λ, d, $\log(1/\delta)$, and $1/\varepsilon$.*

Proof. Let $I = (P^0, Q^1, \ldots, Q^N)$ be an instance. Consider the probability that every hypervolume approximation of $\mathcal{A}_{\text{comp}}$ on I indeed lies in the specified bounds, i.e., we have

$$(1-\varepsilon'/\mu)\mathrm{HYP}(P+q-p) \leqslant H_p \leqslant (1+\varepsilon'/\mu)\mathrm{HYP}(P+q-p)$$

for every computation of a H_p in $\mathcal{A}_{\text{comp}}$. For a single call, this happens with probability at least $1-\delta/(\mu+1)$. Furthermore, we have at most $N(\mu+1)$ hypervolume approximations running $\mathcal{A}_{\text{comp}}$ on I, as in every invocation of $\mathcal{A}_{\text{comp}}$, at most $\mu+1$ hypervolume approximations are computed. With the union bound, we arrive at a probability of at least $1-N\delta$ that all the hypervolume approximations lie within the specified bounds.

Either we stay with the current population or the hypervolume increases by a constant factor, in which case

$$\begin{aligned}\left(1+\frac{\varepsilon'}{\mu}\right)\mathrm{HYP}(P^i) &\geqslant H_{p'}\\ &\geqslant \left(1+\frac{c}{\mu}\right)H_q \geqslant \left(1-\frac{\varepsilon'}{\mu}\right)\left(1+\frac{c}{\mu}\right)\mathrm{HYP}(P^{i-1}).\end{aligned}$$

For $c = 1+2\varepsilon' \geqslant 1$, $\mu \geqslant 1$ and $\varepsilon' \leqslant 1/3$, which is true by assumption, we have $\left(1-\frac{\varepsilon'}{\mu}\right)\left(1+\frac{c}{\mu}\right) \geqslant \left(1+\frac{\varepsilon'}{\mu}\right)$, implying that the algorithm is non-decreasing.

We prove that $\mathcal{A}_{\text{comp}}$ is $(4+2/\mu+\varepsilon)$-competitive if all the hypervolume approximations lie in the specified bounds. To do this, let $P^i = \mathcal{A}_{\text{comp}}(P^{i-1}, Q^i)$ for $i = 1, \ldots, N$. Consider $\hat{P} := \bigcup_{i=0}^{N} P^i$, the set of all points in $\mathrm{Obs}(I)$ that were taken by $\mathcal{A}_{\text{comp}}$ at some point. Let $P^* \subseteq \mathrm{Obs}(I)$ be a μ-population with $\mathrm{HYP}(P^*) = \mathrm{maxHYP}_\mu(\mathrm{Obs}(I))$. We have

$$\begin{aligned}\mathrm{maxHYP}_\mu(\mathrm{Obs}(I)) = \mathrm{HYP}(P^*) &\leqslant \mathrm{HYP}(\hat{P}\cup P^*)\\ &\leqslant \mathrm{HYP}(\hat{P}) + \sum_{q\in P^*\setminus\hat{P}} \mathrm{CON}_{\hat{P}+q}(q) \quad (5)\end{aligned}$$

To continue, we bound $\mathrm{HYP}(\hat{P})$ as well as the contribution of any point $q \in P^* \setminus \hat{P}$ to $\hat{P}$ in terms of $\mathrm{HYP}(P^N)$. This will yield the desired bound for $\mathrm{HYP}(P^N)$. We start with the latter.

Consider a point not chosen by the algorithm, $q \in \mathrm{Obs}(I) \setminus \hat{P}$. We have $Q^i = \{q\}$ for some $1 \leqslant i \leqslant N$. Let $\tilde{p} \in P^{i-1}+q$ with $\mathrm{CON}_{P^{i-1}+q}(\tilde{p})$ minimal among all $p \in P^{i-1}+q$. Then we have

$$\begin{aligned}\mathrm{CON}_{P^{i-1}+q}(\tilde{p}) &\leqslant \frac{1}{\mu+1}\sum_{p\in P^{i-1}+q}\mathrm{CON}_{P^{i-1}+q}(p)\\ &\leqslant \frac{1}{\mu+1}\mathrm{HYP}(P^{i-1}+q)\\ &= \frac{1}{\mu+1}\left(\mathrm{HYP}(P^{i-1}) + \mathrm{CON}_{P^{i-1}+q}(q)\right). \quad (6)\end{aligned}$$

Let $p' = \operatorname{argmax}\{H_p \mid p \in P^{i-1}\}$, see Algorithm 4. The point q was not taken by the algorithm, so we have

$$\begin{aligned}&(1-\frac{\varepsilon'}{\mu})\mathrm{HYP}(P^{i-1}+q-\tilde{p})\\ &\leqslant H_{\tilde{p}} \leqslant H_{p'} \leqslant (1+\frac{c}{\mu})H_q\\ &\leqslant (1+\frac{c}{\mu})(1+\frac{\varepsilon'}{\mu})\mathrm{HYP}(P^{i-1}),\end{aligned}$$

where the factors $(1\pm\varepsilon'/\mu)$ stem from the H_p being approximations. Using $\mathrm{HYP}(P^{i-1}+q-\tilde{p}) = \mathrm{HYP}(P^{i-1}) + \mathrm{CON}_{P^{i-1}+q}(q) - \mathrm{CON}_{P^{i-1}+q}(\tilde{p})$, equation (6) and $\frac{1}{1-\varepsilon'/\mu} \leqslant 1+\frac{2\varepsilon'}{\mu}$ for $\varepsilon' \leqslant 1/2$, this can be simplified to

$$\mathrm{CON}_{P^{i-1}+q}(q) \leqslant \beta\mathrm{HYP}(P^{i-1}),$$

where

$$\beta = \frac{\mu+1}{\mu}\left(\left(1+\frac{c}{\mu}\right)\left(1+\frac{\varepsilon'}{\mu}\right)\left(1+\frac{2\varepsilon'}{\mu}\right)-1\right)+\frac{1}{\mu}.$$

As $\mathrm{HYP}(P^N) \geqslant \mathrm{HYP}(P^{i-1})$ and $\hat{P} \supseteq P^{i-1}$, this implies

$$\mathrm{CON}_{\hat{P}+q}(q) \leqslant \beta\mathrm{HYP}(P^N). \quad (7)$$

Plugging in the definition of $c = 1+2\varepsilon'$, simplifying and roughly bounding the number of terms involving ε'/μ together with their coefficients, and using $\mu \geqslant 1$, we get

$$\beta \leqslant \frac{2}{\mu} + \frac{1}{\mu^2} + 64\frac{\varepsilon'}{\mu}. \quad (8)$$

Now we bound $\mathrm{HYP}(\hat{P})$. Consider $I = \{i \mid P^i \neq P^{i-1}, 1 \leqslant i \leqslant N\}$, the indices where P^i changed, and let p^i be the unique point in $P^{i-1} \setminus P^i$, i.e., the point we deleted in round $i \in I$. Then we have for $i \in I$ and every $p \in P^i + p^i$

$$\begin{aligned}\left(1+\frac{\varepsilon'}{\mu}\right)\mathrm{HYP}(P^i) &\geqslant H_{p_i} \geqslant H_p\\ &\geqslant \left(1-\frac{\varepsilon'}{\mu}\right)\mathrm{HYP}(P^i+p^i-p).\end{aligned}$$

Since $\mathrm{HYP}(P^i+p^i-p) = \mathrm{HYP}(P^i) + \mathrm{CON}_{P^i+p^i}(p^i) - \mathrm{CON}_{P^i+p^i}(p)$, this is equivalent to

$$\frac{2\varepsilon'}{\mu}\left(1-\frac{\varepsilon'}{\mu}\right)^{-1}\mathrm{HYP}(P^i) + \mathrm{CON}_{P^i+p^i}(p) \geqslant \mathrm{CON}_{P^i+p^i}(p^i).$$

Summing over all $p \in P^i + p^i$ and using $\mathrm{HYP}(P) \geqslant \sum_{p\in P}\mathrm{CON}_P(p)$ we get

$$\begin{aligned}&2\varepsilon'\left(1+\frac{1}{\mu}\right)\left(1-\frac{\varepsilon'}{\mu}\right)^{-1}\mathrm{HYP}(P^i) + \mathrm{HYP}(P^i+p^i)\\ &\geqslant (\mu+1)\mathrm{CON}_{P^i+p^i}(p^i),\end{aligned}$$

which yields, after substituting $\mathrm{HYP}(P^i + p^i) = \mathrm{HYP}(P^i) + \mathrm{CON}_{P^i+p^i}(p^i)$ again,

$$\mathrm{CON}_{P^i+p^i}(p^i) \leqslant \gamma \mathrm{HYP}(P^i),$$

with $\gamma = \frac{1}{\mu}\left(1 + 2\varepsilon'\left(1+\frac{1}{\mu}\right)\left(1-\frac{\varepsilon'}{\mu}\right)^{-1}\right)$. Now,

$$\begin{aligned}\mathrm{HYP}(\hat{P}) &= \mathrm{HYP}(P^N) + \sum_{i\in I} \mathrm{CON}_{P^N\cup\ldots\cup P^i+p^i}(p^i)\\ &\leqslant \mathrm{HYP}(P^N) + \sum_{i\in I} \mathrm{CON}_{P^i+p^i}(p^i)\\ &\leqslant \mathrm{HYP}(P^N) + \gamma\sum_{i\in I} \mathrm{HYP}(P^i).\end{aligned}$$

As we go to a new population only if we have an improvement of a factor $(1 + c/\mu)$, but we have approximations, we get $\delta \cdot \mathrm{HYP}(P^i) \geqslant \mathrm{HYP}(P^{i-1})$ for $i \in I$ and $\delta := (1+c/\mu)^{-1}(1-\varepsilon'/\mu)^{-1}(1+\varepsilon'/\mu)$, yielding

$$\begin{aligned}\mathrm{HYP}(\hat{P}) &\leqslant \mathrm{HYP}(P^N)\left(1+\gamma\sum_{i=0}^{N-1}\delta^i\right)\\ &\leqslant \mathrm{HYP}(P^N)\left(1+\frac{\gamma}{1-\delta}\right). \qquad (9)\end{aligned}$$

Plugging in the definition of $c = 1 + 2\varepsilon'$, δ and γ, simple calculations and rough estimations using $\mu \geqslant 1$ and $\varepsilon' \leqslant 1/6$ show that

$$\frac{\gamma}{1-\delta} \leqslant 1 + \frac{1}{\mu} + 40\varepsilon'. \qquad (10)$$

Now we can take equation (5), plug in equations (7) and (9), and simplify using equations (8) and (10) to get

$$\begin{aligned}\mathrm{HYP}(P^*) &\leqslant \mathrm{HYP}(\hat{P}) + \sum_{q\in P^*\setminus\hat{P}} \mathrm{CON}_{\hat{P}+q}(q)\\ &\leqslant \left(1+\frac{\gamma}{1-\delta}+\mu\beta\right)\mathrm{HYP}(P^N)\\ &\leqslant \left(4+\frac{2}{\mu}+104\varepsilon'\right)\mathrm{HYP}(P^N)\\ &= \left(4+\frac{2}{\mu}+\varepsilon\right)\mathrm{HYP}(P^N). \qquad \square\end{aligned}$$

Observe that by setting $\varepsilon = 0$, so that all hypervolume approximations are in fact exact computations, Algorithm 4 becomes the same as Algorithm 3. This implies that the proof of Theorem 4.3 is a proof of Theorem 4.1, too.

The perhaps surprising probability bound is caused by the (necessary) assumption that every call to the hypervolume approximation algorithm indeed returns a $(1+\varepsilon'/\mu)$-approximation. The factor N in the probability can be easily canceled out by a sufficiently small δ, as the runtime depends only logarithmically on $1/\delta$.

5. CONCLUSION

Hypervolume based MOEAs like SIBEA [12], SMS-EMOA [1], and the generational MO-CMA-ES [8] use greedy archiving algorithms. We have proven that such increasing archiving algorithms are computationally inefficient, and achieve only an unbounded competitive ratio of μ. In sharp contrast to this, we presented a non-decreasing archiving algorithm which not only achieves a constant competitive ratio of $4 + 2/\mu$, but is also efficiently computable.

References

[1] N. Beume, B. Naujoks, and M. T. M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181:1653–1669, 2007.

[2] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*, Vol. 2. Cambridge University Press Cambridge, 1998.

[3] L. Bradstreet, L. Barone, and L. While. Maximising hypervolume for selection in multi-objective evolutionary algorithms. In *Proc. IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 6208–6215, 2006.

[4] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry: Theory and Applications*, 43:601–610, 2010.

[5] K. Bringmann and T. Friedrich. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18:383–402, 2010.

[6] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, 2012.

[7] K. Bringmann and T. Friedrich. Convergence of hypervolume-based archiving algorithms I: Effectiveness. In *Proc. 13th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '11)*, pp. 745–752. ACM Press, 2011.

[8] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15:1–28, 2007.

[9] M. López-Ibáñez, J. D. Knowles, and M. Laumanns. On sequential online archiving of objective vectors. In *Proc. 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO '11)*, Vol. 6576 of *LNCS*, pp. 46–60. Springer, 2011.

[10] T. Ulrich and L. Thiele. Bounding the effectiveness of hypervolume-based $(\mu + \lambda)$-archiving algorithms. In *Proc. 6th International Conference on Learning and Intelligent Optimization (LION '12)*, 2012. To appear.

[11] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evolutionary Computation*, 3: 257–271, 1999.

[12] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Proc. Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO '07)*, Vol. 4403 of *LNCS*, pp. 862–876. Springer, 2007.

[13] E. Zitzler, L. Thiele, and J. Bader. On set-based multiobjective optimization. *IEEE Trans. Evolutionary Computation*, 14:58–79, 2010.