Generic Postprocessing via Subset Selection for Hypervolume and Epsilon-Indicator*

Karl Bringmann[1], Tobias Friedrich[2], and Patrick Klitzke[3]

[1] Max Planck Institute for Informatics, Saarbrücken, Germany
[2] Friedrich-Schiller-Universität Jena, Jena, Germany
[3] Universität des Saarlandes, Saarbrücken, Germany

Abstract. Most biobjective evolutionary algorithms maintain a population of fixed size μ and return the final population at termination. During the optimization process many solutions are considered, but most are discarded. We present two generic postprocessing algorithms which utilize the archive of all non-dominated solutions evaluated during the search. We choose the best μ solutions from the archive such that the hypervolume or ε-indicator is maximized. This postprocessing costs no additional fitness function evaluations and has negligible runtime compared to most EMOAs.
We experimentally examine our postprocessing for four standard algorithms (NSGA-II, SPEA2, SMS-EMOA, IBEA) on ten standard test functions (DTLZ 1–2,7, ZDT 1–3, WFG 3–6) and measure the average quality improvement. The median decrease of the distance to the optimal ε-indicator is 95%, the median decrease of the distance to the optimal hypervolume value is 86%. We observe similar performance on a real-world problem (wind turbine placement).

1 Introduction

Biobjective optimization aims at minimizing (or maximizing) a two-dimensional fitness function $f\colon \mathcal{X} \to \mathbb{R}^2$. As the two objectives f_1 and f_2 are typically contradicting, the outcome of the optimization is a set of incomparable solutions describing a Pareto front. Multiobjective evolutionary algorithms (MOEA) typically maintain a set of solutions called population during the optimization. The simplest MOEAs (like SEMO [12, 14, 15]) keep all non-dominated solutions in the population. As the Pareto front of a biobjective fitness function can be exponential in the input size [10], this results in exponential runtimes of SEMO for such fitness functions [6, 13]. More advanced MOEAs therefore avoid keeping all non-dominated solutions in the population and assume some upper limit on the size of the population.

* The research leading to these results has received funding from the Australian Research Council (ARC) under grant agreement DP140103400 and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 618091 (SAGE). K.B. is a recipient of the *Google Europe Fellowship in Randomized Algorithms*, and this research is supported in part by this Google Fellowship.

With a population of fixed maximum size, MOEAs have to decide in each step which solutions to keep in the population and which solutions to remove. This can be done based on various measures like the hypervolume contribution [20], ε-approximation, crowding distance, or many others. Independent of the specific measure, the archiving algorithm of a MOEA has to solve an online problem: It has do decide which solutions to remove without knowing what new solutions will be generated in the future. While the MOEA has to decide 'online' which solutions to keep, an 'offline' algorithm would have access to all points generated during the optimization process and would just choose the best set from this archive. It is known that a MOEA which in each iteration keeps these μ solutions in the population that maximize the hypervolume, can still only reach a final hypervolume which is a factor μ smaller than achieved by the optimal choice of μ solutions from the whole archive (in the worst case) [2].

This shows that MOEAs potentially lose a lot of information by dropping solutions during the optimization. We suggest to make best use of the accumulated information by collecting all search points seen during the optimization in an archive. After the MOEA has stopped, we suggest to do a postprocessing that selects the best subset (of size μ) from the whole archive. This costs no additional fitness evaluations and should not reduce the quality of the reported final population. Depending on the ultimate aim of the optimization the problem solved by this postprocessing is known as the *Hypervolume Subset Selection Problem* (HYPSSP) or *ε-Indicator Subset Selection Problem* (EPSSSP). Until recently the best known runtimes for these subset selection problems were quadratic in the archive size n. As a single run of a MOEA can easily produce $n = 10^5$ or more non-dominated search points, a runtime of $\mathcal{O}(n^2)$ is prohibitively large. Our new postprocessing therefore only becomes tractable due to two new quasi-linear algorithms by the authors [5] published at this years GECCO, specifically, an algorithms for HYPSSP with runtime $\mathcal{O}(n\,(\mu + \log n))$ and one for EPSSSP with runtime $\mathcal{O}(n \log n)$.

In this paper, we are going to investigate the effect of these two postprocessing algorithms for four standard algorithms on 10 common multi-objective optimization test functions and one real-world problem of optimizing the placement of wind turbines [17, 18].

Quality measures. There are several ways to measure the quality of solution sets. We focus on two measures. Our first metric is the hypervolume indicator. It measures the volume of the objective space dominated by the set of solutions relative to a reference point [20]. Its main disadvantage is its high computational complexity in higher dimensions [1, 3].

Our second metric is the ε-indicator. For measuring how well a set P approximates another set R, the (additive) ε-indicator returns the minimal ε by which we have to increase all points in P in all coordinates so that every point in R is dominated by some point in P. The disadvantage of this notion is that its computation requires knowing the Pareto front, as we would like to plug in the Pareto front for R. In contrast to the hypervolume indicator, it can therefore not directly be used to guide the search [4].

2 Preliminaries

We consider biobjective minimization problems, where a vector-valued function $f = (f_1, f_2)\colon \mathcal{X} \to \mathbb{R}^2$ is minimized with respect to the weak Pareto dominance relation $\preceq$. We will mainly work in the *objective space* $f(\mathcal{X})$ and say that a point $p = (p_1, p_2) \in \mathbb{R}^2$ *weakly dominates* another point $q = (q_1, q_2) \in \mathbb{R}^2$ (denoted as $p \preceq q$) iff $p_1 \leq q_1$ and $p_2 \leq q_2$. The aim of most MOEAs is to approximate the *Pareto front* $\mathcal{F} = \{f(x) \mid x \in \mathcal{X}\colon \nexists y \in \mathcal{X}\colon f(y) \preceq f(x) \wedge f(x) \not\preceq f(y)\}$.

Hypervolume indicator. For a set of points $P \subset \mathbb{R}^2$, the *hypervolume indicator* is defined as the volume of the set of points that are weakly dominated by solutions in P and at the same time weakly dominate a given reference point $r \in \mathbb{R}^2$, that is, $\mathcal{I}_{\text{hyp}}(P) := \mathcal{I}_{\text{hyp}}(P, r) := \lambda(\{z \in \mathbb{R}^2 \mid \exists a \in P\colon a \preceq z \preceq r\})$, where λ is the Lebesgue measure.

The *Hypervolume Subset Selection Problem* (HYPSSP) is then defined as follows: Given a set $P \subset \mathbb{R}^2$ of size n, $r \in \mathbb{R}^2$, and $\mu \in \mathbb{N}$, compute a subset $P^* \subseteq P$ of size at most μ that maximizes $\mathcal{I}_{\text{hyp}}(P^*, r)$. We write the result of this problem as $\text{HYPSSP}(P, r, \mu)$.

ε-Indicator. How well a point $p = (p_1, p_2) \in \mathbb{R}^2$ approximates another point $r = (r_1, r_2) \in \mathbb{R}^2$ in the objective space can be measured by the minimal number ε by which we have to decrease p in both coordinates so that it dominates q. More formally, we set $\mathcal{I}_{\text{eps}}(p, r) := \max\{p_1 - r_1, p_2 - r_2\}$. This can be used to define how well a set $P \subset \mathbb{R}^2$ approximates a set $R \subset \mathbb{R}^2$: The ε-indicator is defined as $\mathcal{I}_{\text{eps}}(P, R) := \max_{r \in R} \min_{p \in P} \mathcal{I}_{\text{eps}}(p, r)$. This denotes the minimal number ε by which we have to decrease all points in P in both coordinates so that every point in R is dominated by some point in P.

The *ε-Indicator Subset Selection Problem* (EPSSSP) is defined as follows: Given a set $P \subset \mathbb{R}^d$ of size n, $R \subset \mathbb{R}^d$ of size m, and $\mu \in \mathbb{N}$, compute a subset $P^* \subseteq P$ of size at most μ that minimizes $\mathcal{I}_{\text{eps}}(P^*, R)$. We write the result of this problem as $\text{EPSSSP}(P, R, \mu)$.

3 Postprocessing

Consider any EMOA with population size μ running until it performed n fitness evaluations. Let P be the final population after n fitness evaluations and A be the archive of all n solutions that were evaluated during the run. We describe our postprocessing in the objective space, i.e., we let $P, A \subset \mathbb{R}^2$.

Hypervolume. For $\mathcal{I}_{\text{hyp}}$ we may pick any reference point $r \in \mathbb{R}^2$. The general optimization goal is then to find a population P^* of μ Pareto optimal points that maximize the hypervolume, i.e., $P^* = \text{HYPSSP}(\mathcal{F}, r, \mu)$. Unfortunately, the Pareto front is unknown. To overcome this problem, we introduce the assumption that the archive converges to the Pareto front (as has been done in AGE [4] and is

implicit in the design of most EMOAs). Thus, our $\mathcal{I}_{\text{hyp}}$-postprocessing computes the set of μ points maximizing the hypervolume among all points in the archive,

$$\text{PP}_{\text{hyp}}(A, \mu) := \text{HYPSSP}(A, r, \mu).$$

Typically the hypervolume of $\text{PP}_{\text{hyp}}(A, \mu)$ should be larger than the hypervolume of P, so that our postprocessing improves the quality. In Section 5 we will see an experimental evaluation of this claim. In any case, since $P \subseteq A$ we have $\mathcal{I}_{\text{hyp}}(\text{PP}_{\text{hyp}}(A, \mu)) \geq \mathcal{I}_{\text{hyp}}(P)$, so our postprocessing does not decrease the quality of the result.

ε-Indicator. In case of $\mathcal{I}_{\text{eps}}$ the general optimization goal is to find a population P^* of μ Pareto optimal points that optimally approximate the Pareto front, i.e., $P^* = \text{EPSSSP}(\mathcal{F}, \mathcal{F}, \mu)$. Again, $\mathcal{F}$ is unknown and we assume that the archive converges to the Pareto front. Thus, our $\mathcal{I}_{\text{eps}}$-postprocessing computes the set of μ points among all points in the archive that best approximate the archive,

$$\text{PP}_{\text{eps}}(A, \mu) := \text{EPSSSP}'(A, A, \mu).$$

Here, the prime in EPSSSP' hides a minor modification that improves the experimental results, namely that we choose a population P among all μ-subsets of A that include the leftmost point of A and the bottommost point of A, i.e., the single-objective optima. Intuitively, this is necessary since these extrema are needed to cover the boundaries of the Pareto front. We remark that one can compute $\text{EPSSSP}'(A, A, \mu)$ with a minor modification of [5].

Again, typically $\mathcal{I}_{\text{eps}}(\text{PP}_{\text{eps}}(A, \mu), \mathcal{F})$ should be smaller than $\mathcal{I}_{\text{eps}}(P, \mathcal{F})$, so that our postprocessing improves the quality, and we will examine this claim experimentally. A noteworthy difference to the hypervolume case is that our postprocessing for $\mathcal{I}_{\text{eps}}$ does not come with the guarantee that quality cannot deteriorate, in fact, we will see in Section 5 that worsenings can happen but are rare.

Complexity. For a population size of μ and n fitness evaluations, NSGA-II, SPEA2, and IBEA have a running time of $\mathcal{O}(n\mu \log \mu)$, while SMS-EMOA has a running time of $\mathcal{O}(n\mu^2)$. Our postprocessing takes time $\mathcal{O}(n\,(\mu + \log n))$ for $\mathcal{I}_{\text{hyp}}$ and $\mathcal{O}(n \log n)$ for $\mathcal{I}_{\text{eps}}$ [5], which is comparable to the runtime of typical EMOAs. In our experiments, the postprocessing tends to be even faster, since we only have to store the non-dominated points of the archive, which are much less than n.

4 Experimental Setup

Implementation and hardware. All presented algorithms have been implemented in Java using the jMetal framework [9] and run on a compute cluster with 128 nodes, each having two Intel Xeon E5620 @ 2.40GHz. The code will is available on `http://docs.theinf.uni-jena.de/code/ssp.zip`.

Benchmark problems and EMOAs. We compared the improvement gained by the postprocessing for the well established EMOAs NSGA-II [7], SPEA2 [22], SMS-EMOA [11], and IBEA [19]. As test functions we used DTLZ1, DTLZ2, DTLZ7 from DTLZ [8] with 7 variables, ZDT1–3 from ZDT [21] with 30 variables, and WFG3–6 from the [16] with 4 variables. We chose these benchmarks, because explicit expressions for their Pareto fronts are readily available. For measuring the hypervolume, we choose the reference point $r = (11, 11)$.

Additionally to the standard test functions, we used a simulation of a wind turbine placement function [17], which optimizes for the maximum power and the minimum perimeter of the convex hull formed by the turbine positions with 30 turbines on a discrete area of size 3000×3000.

All experimental results (medians, quartiles, ...) that we will report in the next section are based on 700 independent runs for the benchmark problems and 100 independent runs for the turbine. As population size we used 100 for the benchmark problems and 10 for the turbine.

Quality Evaluation. We want to compare the quality $\mathcal{I}_{\text{hyp}}(P, r)$ of a population P with the optimal hypervolume $\text{OPT}_{\text{hyp}} := \text{HYPSSP}(\mathcal{F}, \mu)$ of any μ points on the Pareto front. For measuring the proximity to the front, we measure $\mathcal{I}^{\Delta}_{\text{hyp}}(P, r) := \text{OPT}_{\text{hyp}} - \mathcal{I}_{\text{hyp}}(P, r)$ in our experiments. However, as $\mathcal{F}$ is infinite it seems impossible to compute OPT_{hyp}. Instead of $\mathcal{F}$ we therefore consider a set $\mathcal{F}' \subseteq \mathcal{F}$ of m points placed equidistantly along the front, which we can compute because we know an explicit expression for $\mathcal{F}$ for the chosen benchmark problems. Now we simply replace $\mathcal{F}$ by $\mathcal{F}'$ in the definition of OPT_{hyp} to obtain an approximation.

In case of $\mathcal{I}_{\text{eps}}$, for the same reasons we cannot directly compute $\text{OPT}_{\text{eps}} := \text{EPSSSP}(\mathcal{F}, \mathcal{F}, \mu)$. Moreover, we have the additional difficulty that we cannot evaluate the quality $\mathcal{I}_{\text{eps}}(P, \mathcal{F})$ of a population P. Again, we replace $\mathcal{F}$ by its finite approximation $\mathcal{F}'$ and obtain approximations for the optimum and the quality of a population. We use $\mathcal{I}^{\Delta}_{\text{eps}}(P, \mathcal{F}) := \text{OPT}_{\text{eps}} - \mathcal{I}_{\text{eps}}(P, \mathcal{F})$ as quality measure.

In our experiments we choose $m = 10^6$, which makes the error smaller than any of our reported values, and we ignore this error from now on. Note that it is infeasible to make m much larger, since the runtime for computing the approximation of, e.g., OPT_{hyp} is $\mathcal{O}(m\,(\mu + \log m))$.

Statistics. Additionally to calculating the (median) quality with and without postprocessing, we also perform a non-parametric test on the significance of the observed behavior. For this, we use the Wilcoxon-Mann-Whitney two-sample rank-sum test at the 95% confidence level.

5 Experimental Results

Test functions. The results of our experimental study on standard test functions are presented in Figure 1. The tables show the median of the indicators

Function	NSGA-II	IBEA	SPEA2	SMS-EMOA
DTLZ1	$4.7 \cdot 10^{-4} \xrightarrow{-85.0\%} 7.1 \cdot 10^{-5}$ (1.00)	$8.1 \cdot 10^{-2} \xrightarrow{-99.6\%} 3.1 \cdot 10^{-4}$ (1.00)	$2.0 \cdot 10^{-1} \xrightarrow{-99.9\%} 5.4 \cdot 10^{-5}$ (1.00)	$3.1 \cdot 10^{-5} \xrightarrow{-10.1\%} 2.8 \cdot 10^{-5}$ (1.00)
DTLZ2	$1.9 \cdot 10^{-3} \xrightarrow{-99.3\%} 1.3 \cdot 10^{-5}$ (1.00)	$1.3 \cdot 10^{-2} \xrightarrow{-99.9\%} 1.1 \cdot 10^{-5}$ (1.00)	$1.7 \cdot 10^{-1} \xrightarrow{-99.9\%} 1.2 \cdot 10^{-5}$ (1.00)	$1.9 \cdot 10^{-5} \xrightarrow{-41.3\%} 1.1 \cdot 10^{-5}$ (1.00)
DTLZ7	$2.1 \cdot 10 \xrightarrow{=} 2.1 \cdot 10$ (1.00)	$3.4 \cdot 10 \xrightarrow{-0.04\%} 3.4 \cdot 10$ (1.00)	$2.2 \cdot 10 \xrightarrow{-0.6\%} 2.1 \cdot 10$ (1.00)	$2.1 \cdot 10 \xrightarrow{=} 2.1 \cdot 10$ (1.00)
ZDT1	$1.9 \cdot 10^{-3} \xrightarrow{-99.1\%} 1.8 \cdot 10^{-5}$ (1.00)	$5.4 \cdot 10^{-2} \xrightarrow{-99.8\%} 8.2 \cdot 10^{-5}$ (1.00)	$6.0 \cdot 10^{-2} \xrightarrow{-99.9\%} 2.3 \cdot 10^{-5}$ (1.00)	$2.7 \cdot 10^{-5} \xrightarrow{-46.9\%} 1.4 \cdot 10^{-5}$ (1.00)
ZDT2	$1.7 \cdot 10^{-3} \xrightarrow{-99.2\%} 1.4 \cdot 10^{-5}$ (1.00)	$1.6 \cdot 10^{-2} \xrightarrow{-99.9\%} 1.1 \cdot 10^{-5}$ (1.00)	$1.8 \cdot 10^{-1} \xrightarrow{-99.9\%} 2.9 \cdot 10^{-5}$ (1.00)	$3.7 \cdot 10^{-5} \xrightarrow{-47.3\%} 1.9 \cdot 10^{-5}$ (1.00)
ZDT3	$1.1 \cdot 10^{-3} \xrightarrow{-99.4\%} 6.1 \cdot 10^{-6}$ (1.00)	$3.5 \cdot 10^{-2} \xrightarrow{-99.9\%} 3.3 \cdot 10^{-5}$ (1.00)	$5.7 \cdot 10^{-2} \xrightarrow{-99.9\%} 9.2 \cdot 10^{-6}$ (1.00)	$1.3 \cdot 10^{-5} \xrightarrow{-59.3\%} 5.1 \cdot 10^{-6}$ (1.00)
WFG3	$2.9 \cdot 10^{-2} \xrightarrow{-60.1\%} 1.2 \cdot 10^{-2}$ (1.00)	$2.9 \cdot 10^{-1} \xrightarrow{-97.4\%} 7.7 \cdot 10^{-3}$ (1.00)	$8.1 \cdot 10^{-1} \xrightarrow{-98.5\%} 1.2 \cdot 10^{-2}$ (1.00)	$7.3 \cdot 10^{-3} \xrightarrow{-2.1\%} 7.2 \cdot 10^{-3}$ (0.63)
WFG4	$1.8 \cdot 10^{-2} \xrightarrow{-90.5\%} 1.7 \cdot 10^{-3}$ (1.00)	$9.5 \cdot 10^{-2} \xrightarrow{-99.9\%} 3.8 \cdot 10^{-5}$ (1.00)	$1.2 \xrightarrow{-99.0\%} 1.2 \cdot 10^{-2}$ (1.00)	$1.2 \cdot 10^{-2} \xrightarrow{-0.4\%} 1.2 \cdot 10^{-2}$ (0.61)
WFG5	$1.7 \xrightarrow{-0.8\%} 1.6$ (1.00)	$1.7 \xrightarrow{-2.7\%} 1.6$ (1.00)	$2.8 \xrightarrow{-40.3\%} 1.7$ (1.00)	$1.8 \xrightarrow{=} 1.8$ (0.53)
WFG6	$1.9 \cdot 10^{-1} \xrightarrow{-7.2\%} 1.8 \cdot 10^{-1}$ (0.96)	$4.0 \cdot 10^{-1} \xrightarrow{-49.0\%} 2.0 \cdot 10^{-1}$ (1.00)	$1.3 \xrightarrow{-86.2\%} 1.9 \cdot 10^{-1}$ (1.00)	$2.3 \cdot 10^{-1} \xrightarrow{-0.03\%} 2.3 \cdot 10^{-1}$ (0.52)

(a) Distance to optimal hypervolume: $\mathcal{I}^{\Delta}_{\text{hyp}}(P) = \text{OPT}_{\text{hyp}} - \mathcal{I}_{\text{hyp}}(P)$.

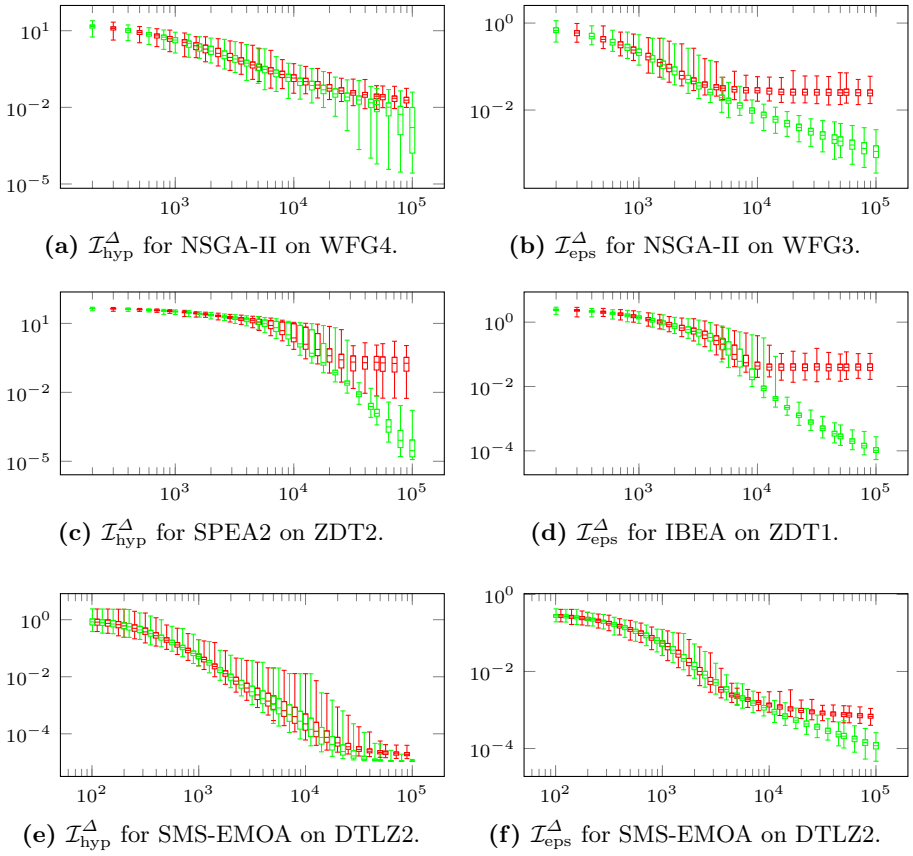
Function	NSGA-II	IBEA	SPEA2	SMS-EMOA
DTLZ1	$4.1 \cdot 10^{-3} \xrightarrow{-94.3\%} 2.3 \cdot 10^{-4}$ (1.00)	$2.0 \cdot 10^{-1} \xrightarrow{-96.2\%} 7.5 \cdot 10^{-3}$ (1.00)	$2.5 \cdot 10^{-2} \xrightarrow{-99.1\%} 2.2 \cdot 10^{-4}$ (1.00)	$3.5 \cdot 10^{-4} \xrightarrow{-50.4\%} 1.7 \cdot 10^{-4}$ (1.00)
DTLZ2	$8.2 \cdot 10^{-3} \xrightarrow{-97.9\%} 1.7 \cdot 10^{-4}$ (1.00)	$4.0 \cdot 10^{-2} \xrightarrow{-99.7\%} 1.2 \cdot 10^{-4}$ (1.00)	$2.4 \cdot 10^{-2} \xrightarrow{-99.4\%} 1.4 \cdot 10^{-4}$ (1.00)	$6.7 \cdot 10^{-4} \xrightarrow{-82.3\%} 1.2 \cdot 10^{-4}$ (1.00)
DTLZ7	$2.3 \xrightarrow{=} 2.3$ (0.51)	$3.6 \xrightarrow{-0.03\%} 3.6$ (1.00)	$2.3 \xrightarrow{-0.2\%} 2.3$ (1.00)	$2.3 \xrightarrow{=} 2.3$ (0.99)
ZDT1	$8.3 \cdot 10^{-3} \xrightarrow{-98.0\%} 1.7 \cdot 10^{-4}$ (1.00)	$4.0 \cdot 10^{-2} \xrightarrow{-99.7\%} 1.0 \cdot 10^{-4}$ (1.00)	$2.2 \cdot 10^{-2} \xrightarrow{-99.3\%} 1.5 \cdot 10^{-4}$ (1.00)	$5.7 \cdot 10^{-4} \xrightarrow{-79.7\%} 1.2 \cdot 10^{-4}$ (1.00)
ZDT2	$8.1 \cdot 10^{-3} \xrightarrow{-98.1\%} 1.5 \cdot 10^{-4}$ (1.00)	$4.1 \cdot 10^{-2} \xrightarrow{-99.6\%} 1.8 \cdot 10^{-4}$ (1.00)	$2.5 \cdot 10^{-2} \xrightarrow{-99.4\%} 1.5 \cdot 10^{-4}$ (1.00)	$5.5 \cdot 10^{-4} \xrightarrow{-76.1\%} 1.3 \cdot 10^{-4}$ (1.00)
ZDT3	$6.0 \cdot 10^{-3} \xrightarrow{-98.3\%} 1.0 \cdot 10^{-4}$ (1.00)	$2.4 \cdot 10^{-2} \xrightarrow{-98.4\%} 3.8 \cdot 10^{-4}$ (1.00)	$2.5 \cdot 10^{-2} \xrightarrow{-99.5\%} 1.2 \cdot 10^{-4}$ (1.00)	$1.3 \cdot 10^{-3} \xrightarrow{-94.9\%} 6.6 \cdot 10^{-5}$ (1.00)
WFG3	$2.5 \cdot 10^{-2} \xrightarrow{-95.5\%} 1.1 \cdot 10^{-3}$ (1.00)	$1.1 \cdot 10^{-1} \xrightarrow{-99.4\%} 6.5 \cdot 10^{-4}$ (1.00)	$8.3 \cdot 10^{-2} \xrightarrow{-98.7\%} 1.1 \cdot 10^{-3}$ (1.00)	$1.9 \cdot 10^{-3} \xrightarrow{-66.7\%} 6.2 \cdot 10^{-4}$ (1.00)
WFG4	$2.4 \cdot 10^{-2} \xrightarrow{-97.8\%} 5.2 \cdot 10^{-4}$ (1.00)	$8.7 \cdot 10^{-2} \xrightarrow{-99.7\%} 2.3 \cdot 10^{-4}$ (1.00)	$1.0 \cdot 10^{-1} \xrightarrow{-99.5\%} 5.0 \cdot 10^{-4}$ (1.00)	$2.2 \cdot 10^{-3} \xrightarrow{-92.4\%} 1.7 \cdot 10^{-4}$ (1.00)
WFG5	$1.2 \cdot 10^{-1} \xrightarrow{=} 1.2 \cdot 10^{-1}$ (1.00)	$1.3 \cdot 10^{-1} \xrightarrow{-10.3\%} 1.2 \cdot 10^{-1}$ (1.00)	$2.1 \cdot 10^{-1} \xrightarrow{-41.0\%} 1.2 \cdot 10^{-1}$ (1.00)	$1.4 \cdot 10^{-1} \xrightarrow{=} 1.4 \cdot 10^{-1}$ (0.50)
WFG6	$3.3 \cdot 10^{-2} \xrightarrow{-73.8\%} 8.7 \cdot 10^{-3}$ (1.00)	$1.2 \cdot 10^{-1} \xrightarrow{-91.9\%} 9.8 \cdot 10^{-3}$ (1.00)	$1.1 \cdot 10^{-1} \xrightarrow{-92.3\%} 8.7 \cdot 10^{-3}$ (1.00)	$1.2 \cdot 10^{-2} \xrightarrow{-16.2\%} 1.0 \cdot 10^{-2}$ (1.00)

(b) Distance to optimal ε-Indicator: $\mathcal{I}^{\Delta}_{\text{eps}}(P) = \text{OPT}_{\text{eps}} - \mathcal{I}_{\text{eps}}(P, \mathcal{F})$.

Fig. 1: Medians after 100 000 evaluations. We show the value of the respective indicator before and after the postprocessing. We also give the factor by which the indicator got smaller (=better). Improvements of less than 0.01% are treated as equal and marked with "$\xrightarrow{=}$". The value in the brackets gives the p-value for the one-tailed Mann-Whitney U-test. If the p-value is larger or equal 0.95, our postprocessing yields a statistically significant improvement at the 95% confidence level. For each test function we marked the best indicator values before and after postprocessing in blue.

$\mathcal{I}^{\Delta}_{\text{hyp}}$ and $\mathcal{I}^{\Delta}_{\text{eps}}$ after 100 000 fitness evaluations. Our postprocessing improves (or does not worsen) the hypervolume and ε-indicator for all functions and algorithms. In all but six (out of 80 combinations) the improvement is statistically significant at the 95% confidence level. In fact, the median hypervolume is always increased and the distance to the optimal hypervolume $\mathcal{I}^{\Delta}_{\text{hyp}}$ therefore decreased. In 37 out of 40 cases the median $\mathcal{I}^{\Delta}_{\text{hyp}}$ decreases by more than 0.01%. The median reduction of $\mathcal{I}^{\Delta}_{\text{hyp}}$ for all 40 combinations of algorithms and functions is -85.6% (mean -60.5%). On the other hand, the median $\mathcal{I}^{\Delta}_{\text{eps}}$ could be decreased by more than 0.01% in 36 out of 40 cases. The median reduction of $\mathcal{I}^{\Delta}_{\text{eps}}$ is -95.2% (mean -73.4%).

Figure 2 gives a more detailed view on how the indicators $\mathcal{I}^{\Delta}_{\text{hyp}}$ and $\mathcal{I}_{\text{eps}}$ decrease over time for some exemplary combinations of MOEAs and test functions. Shown are box-and-whisker plots that specify the median, quartiles, and whiskers from minimum to maximum. Figures 2a and 2b show the typical behaviour of our postprocessing for NSGA-II. It is interesting to compare Figure 2b with Figure 3a, which shows the size of the population and of the archive over time. This shows that our postprocessing starts to "kick in" at the time we are starting to throw away points, where the archive size diverges from the population size.



(a) $\mathcal{I}_{\text{hyp}}^{\Delta}$ for NSGA-II on WFG4.

(b) $\mathcal{I}_{\text{eps}}^{\Delta}$ for NSGA-II on WFG3.

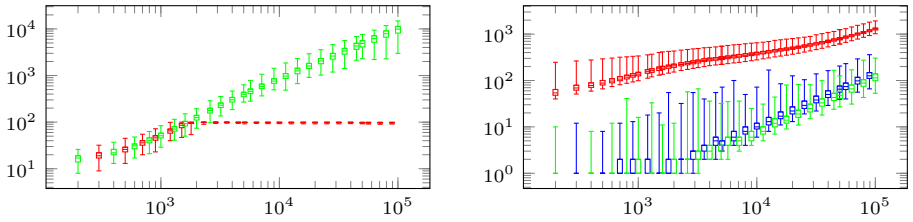
(c) $\mathcal{I}_{\text{hyp}}^{\Delta}$ for SPEA2 on ZDT2.

(d) $\mathcal{I}_{\text{eps}}^{\Delta}$ for IBEA on ZDT1.

(e) $\mathcal{I}_{\text{hyp}}^{\Delta}$ for SMS-EMOA on DTLZ2.

(f) $\mathcal{I}_{\text{eps}}^{\Delta}$ for SMS-EMOA on DTLZ2.

Fig. 2: Quality measures as a function of time (evaluations) before (in red) and after (in green) postprocessing for some exemplary combinations of MOEAs and test functions.



(a) Population size (in red) and archive size (in green) as a function of time (evaluations) for NSGA-II on WFG3.

(b) Runtime (in milliseconds) of NSGA-II (in red) on WFG3 compared to runtime of postprocessing for $\mathcal{I}_{\text{hyp}}$ (in blue) and $\mathcal{I}_{\text{eps}}$ (in green).

Fig. 3: Population size and runtime of NSGA-II on WFG3.

Figure 2c shows the visually largest improvement that our $\mathcal{I}_{\text{hyp}}$-postprocessing was able to achieve for any benchmark problem and MOEA. Note that SPEA2 without postprocessing converges to a suboptimal hypervolume, but our $\mathcal{I}_{\text{hyp}}$-postprocessing is able to correct this. Figure 2d shows a similar profitable situation for $\mathcal{I}_{\text{eps}}$-postprocessing.

Since SMS-EMOA is hypervolume driven, one expects that $\mathcal{I}_{\text{hyp}}$-postprocessing is less effective for this algorithm compared to other EMOAs. Indeed, Figure 1 confirms this expectation, specifically, the median improvement of $\mathcal{I}^{\Delta}_{\text{hyp}}$ for SMS-EMOA is only -6.1% (mean -20.8%). However, Figure 2e shows that even the hypervolume driven SMS-EMOA can benefit a lot from our $\mathcal{I}_{\text{hyp}}$-postprocessing in some situations.

In contrast to the $\mathcal{I}_{\text{hyp}}$-postprocessing, the $\mathcal{I}_{\text{eps}}$-postprocessing can (at least theoretically) make $\mathcal{I}_{\text{eps}}$ worse. However, Figure 2f shows the only case where we observed that the $\mathcal{I}_{\text{eps}}$-postprocessing visually worsens the algorithm at some time, namely in a thin region from 2500 to 5000 fitness evaluations. After that point, the $\mathcal{I}_{\text{eps}}$-postprocessing gives again a huge improvement.

Figure 3b shows the total runtime up to n fitness evaluations of NSGA-II, the runtime of the other algorithms is similar (with SMS-EMOA being somewhat slower). Moreover, the runtime of the postprocessing when started after n fitness evaluations is plotted. This shows that the runtime of both postprocessing algorithms is negligible compared to the runtime of the MOEA.

Additionally, we examined whether a hypervolume-based algorithm (like SMS-EMOA) achieves more hypervolume than a non-hypervolume-based algorithms (like NSGA-II, SPEA2) with $\mathcal{I}_{\text{hyp}}$-postprocessing. To this end, we compared SMS-EMOA without postprocessing to the other three algorithms with postprocessing. The Wilcoxon-Mann-Whitney U-test at 95% confidence level showed that NSGA-II with $\mathcal{I}_{\text{hyp}}$-postprocessing outperforms SMS-EMOA without postprocessing on 7 out of 10 test functions. The same holds for SPEA2, but not for IBEA. This shows that our $\mathcal{I}_{\text{hyp}}$-postprocessing makes algorithms that do not aim at maximizing $\mathcal{I}_{\text{hyp}}$ very competitive, even compared to algorithms that directly optimize $\mathcal{I}_{\text{hyp}}$. As one might expect, we no longer observe this behavior if SMS-EMOA is also allowed postprocessing: SMS-EMOA with $\mathcal{I}_{\text{hyp}}$-postprocessing achieves higher hypervolumes than all other algorithms with $\mathcal{I}_{\text{hyp}}$-postprocessing (on more than half of the test functions).

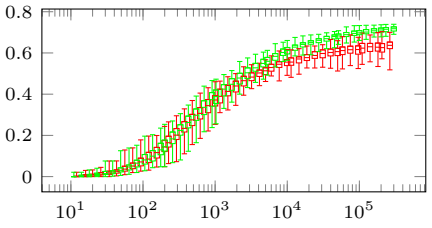


Fig. 4: Experimental results for turbine: $\mathcal{I}_{\text{hyp}}$ (not $\mathcal{I}^{\Delta}_{\text{hyp}}$!) over time (evaluations) for SMS-EMOA before (in red) and after (in green) postprocessing.

Figure 4 shows an exemplary result for the turbine problem for SMS-EMOA. Note that we cannot plot $\mathcal{I}^{\Delta}_{\text{hyp}}$ as OPT_{hyp} is not known. We observe a significant improvement also for this real-world problem. Specifically, after 300 000 evaluations our $\mathcal{I}_{\text{hyp}}$-postprocessing increased $\mathcal{I}_{\text{hyp}}$ by 14%.

6 Conclusion

We studied two generic postprocessing methods which have the potential to improve the final output of any EMOA on any biobjective optimization problem. These methods choose the optimal subset of μ solutions from the archive of all solutions seen during the run of an EMOA such that the hypervolume or ε-indicator is optimized. This requires no additional fitness evaluations and therefore zero additional 'optimization time'. Moreover, the computation time of our postprocessing methods is negligible compared to the computation time of typical EMOAs. We experimentally evaluated the quality of our postprocessing on four standard EMOAs and ten standard test functions and one real-world problem. This showed that our postprocessing typically returns a set of solutions which is about 90% closer to the optimum than the regular outcome of the EMOAs.

Bibliography

[1] 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.

[2] K. Bringmann and T. Friedrich. Convergence of hypervolume-based archiving algorithms II: Competitiveness. In *14th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '12)*, pages 457–464. ACM Press, 2012.

[3] K. Bringmann and T. Friedrich. Parameterized average-case complexity of the hypervolume indicator. In *15th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '13)*, pages 575–582. ACM Press, 2013.

[4] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pages 1198–1203. IJCAI/AAAI, 2011.

[5] K. Bringmann, T. Friedrich, and P. Klitzke. Two-dimensional subset selection for hypervolume and epsilon-indicator. In *16th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '14)*. ACM Press, 2014. `http://docs.theinf.uni-jena.de/paper/2014GECCO.pdf`.

[6] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. On the effects of adding objectives to plateau functions. *IEEE Transactions on Evolutionary Computation*, 13(3):591–603, 2009.

[7] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. 2005.

[9] J. J. Durillo, A. J. Nebro, and E. Alba. The jMetal framework for multi-objective optimization: Design and architecture. In *IEEE Congress on Evolutionary Computation (CEC '10)*, pages 4138–4325, 2010.

[10] M. Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.

[11] M. T. M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO '05)*, pages 62–76. Springer, 2005.

[12] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4):617–633, 2010.

[13] T. Friedrich, N. Hebbinghaus, and F. Neumann. Plateaus can be harder in multi-objective optimization. *Theoretical Computer Science*, 411(6):854–864, 2010.

[14] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *IEEE Congress on Evolutionary Computation (CEC '03)*, pages 1918–1925, 2003.

[15] O. Giel and P. K. Lehre. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18(3):335–356, 2010.

[16] S. Huband, L. Barone, R. L. While, and P. Hingston. A scalable multi-objective test problem toolkit. In *3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO '05)*, volume 3410 of *LNCS*, pages 280–295. Springer, 2005.

[17] R. Tran, J. Wu, C. Denison, T. Ackling, M. Wagner, and F. Neumann. Fast and effective multi-objective optimisation of wind turbine placement. In *15th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '13)*, pages 1381–1388, 2013.

[18] M. Wagner, J. Day, and F. Neumann. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, 51:64 – 70, 2013.

[19] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 832–842. Springer, 2004.

[20] 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.

[21] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[22] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN '01)*, pages 95–100, 2002.