Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods

Dimo Brockhoff and Eckart Zitzler

***Abstract*— Hypervolume based multiobjective evolutionary algorithms (MOEA) nowadays seem to be the first choice when handling multiobjective optimization problems with many, i.e., at least three objectives. Experimental studies have shown that hypervolume-based search algorithms as SMS-EMOA can outperform established algorithms like NSGA-II and SPEA2. One problem remains with most of the hypervolume based algorithms: the best known algorithm for computing the hypervolume needs time exponentially in the number of objectives. To save computation time during hypervolume computation which can be better spent in the generation of more solutions, we propose a general approach how objective reduction techniques can be incorporated into hypervolume based algorithms. Different objective reduction strategies are developed and then compared in an experimental study on two test problems with up to nine objectives. The study indicates that the (temporary) omission of objectives can improve hypervolume based MOEAs drastically in terms of the achieved hypervolume indicator values.**

I. Introduction

In the last decade, multiobjective evolutionary algorithms got established for tackling optimization problems with few objectives, whereas optimization problems with many objectives, i.e., at least three or more, have gained interest in recent years. Real world problems intrinsically have many objectives and from a practical point of view it is often desirable with most applications to include as many objectives as possible without the need to specify preferences among the different criteria. Unfortunately, a large number of objectives leads to further difficulties with respect to decision making, visualization, and computation with the result that state-of-the-art algorithms like NSGA-II or SPEA2 are not suited to solve high-dimensional problems as it was shown by Wagner et al. in an experimental study [1]. Hypervolume-based search strategies such as SMS-EMOA [2], however, seem to be a good alternative, but all known algorithms to (exactly) compute the hypervolume for a solution set have running times exponentially in the number of objectives [3], [4], [5]. Another problem with many objectives will occur, if the computation of the objective values itself is expensive, e.g., if the objective values are defined by time-consuming simulations.

Reducing the number of considered objectives would circumvent the latter problems. It is also undoubted that an objective reduction can assist in the decision making process. Although dimensionality reduction is a common task in statistics, data mining, and other fields, and many methods with applications are known [6], the transfer to evolutionary multiobjective optimization just started recently. Deb and Saxena [7] developed an objective reduction method based on principal component analysis (PCA) [8] which was intended for multiobjective optimization scenarios where the Pareto front is of lower dimension than the objective space itself. Because it is based on principal component analysis, the considered objective (sub)sets do not reflect the original problem in terms of Pareto-dominance. Brockhoff and Zitzler [9] developed a relation-based objective reduction method which aims at finding a subset of objectives such that either the entire or most of the dominance relation is preserved in terms of a defined measure of conflict between objectives. This relation-based objective reduction approach can both find an objective subset of predefined size with minimal error and minimal sets which comply with a given error. This paper, for the first time, incorporates a dimensionality reduction approach into a hypervolume based algorithm and investigates for different types how an objective reduction method can improve evolutionary algorithms in terms of computation time and quality.

As main contributions of this paper we

- propose a general scheme to integrate objective reduction methods into a multiobjective evolutionary algorithm,
- develop different types of objective reduction methods with objective sets of fixed, randomly and adaptively chosen sizes, and
- extensively compare the proposed algorithmic variants with and without objective reduction experimentally.

The experimental results show that objective reduction techniques can significantly improve the performance of multiobjective evolutionary algorithms on many-objective test problems. The paper is organized as follows. In Sec. II, we start with a brief review of objective reduction using ε-dominance introduced in [9]. Section III gives an overview of the different objective reduction algorithms used in the experimental study to follow. The used test problems are defined in Sec. IV and Sec. V describes the entire experimental setup. Section VI presents the results of the performed comparison and Sec. VII concludes the paper.

II. Objective Reduction Using ε-dominance

Without loss of generality, in this paper we consider minimization problems with k objective functions $f_i : X \to \mathbb{R}$, $1 \leq i \leq k$, where the vector function $f := (f_1, \ldots, f_k)$ maps each solution $\vec{x} \in X$ to an objective vector $f(\vec{x}) \in$

Both authors are with the Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland (contact via email: {brockhoff,zitzler}@tik.ee.ethz.ch). Dimo Brockhoff has been supported by the Swiss National Science Foundation (SNF) under grant 112079.

1-4244-1340-0/07/$25.00

$\mathbb{R}^k$. Furthermore, we assume that the underlying dominance structure is given by the weak Pareto dominance relation which is defined as follows: $\preceq_{\mathcal{F}'} := \{(\vec{x},\vec{y}) \mid \vec{x},\vec{y} \in X \wedge \forall f_i \in \mathcal{F}' : f_i(\vec{x}) \leq f_i(\vec{y})\}$, where $\mathcal{F}'$ is a set of objectives with $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, \ldots, f_k\}$. For better readability, we will sometimes only consider the indices of objective functions, e.g., $\mathcal{F}' = \{1,2,3\}$ instead of $\mathcal{F}' = \{f_1, f_2, f_3\}$. We say *$\vec{x}$ weakly dominates $\vec{y}$ with respect to the objective set $\mathcal{F}'$* ($\vec{x} \preceq_{\mathcal{F}'} \vec{y}$) if $(\vec{x},\vec{y}) \in \preceq_{\mathcal{F}'}$. The solutions $\vec{x}$ and $\vec{y}$ are called *comparable* if either $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ or $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$ and *incomparable* if neither $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ nor $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$. If both $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ and $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$ the two solutions are called *indifferent*. A solution $\vec{x}^* \in X$ is called *Pareto optimal* if there is no other $\vec{x} \in X$ that weakly dominates $\vec{x}^*$ with respect to the set of all objectives. The set of all Pareto optimal solutions is called *Pareto (optimal) set*, for which an approximation is sought.

In the presence of many objectives, the question arises whether all objectives of a problem are necessary. An objective would be called necessary, if the Pareto dominance relation changes while omitting the objective. An omission of objectives can affect the underlying weak Pareto dominance relation only in two ways; (i) comparable solutions can become indifferent and (ii) incomparable solutions can become comparable (or even indifferent). While an objective reduction to the smallest set preserving the weak Pareto dominance relation (a *minimum objective set*) is desirable, in practice a further reduction is often necessary. If a change in the dominance relation is necessary, we want to choose an objective set minimizing a certain error. The (additive) ε-dominance relation $\preceq^{\varepsilon}_{\mathcal{F}} := \{(\vec{x},\vec{y}) \mid \vec{x},\vec{y} \in X \wedge \forall f_i \in \mathcal{F} : f_i(\vec{x}) - \varepsilon \leq f_i(\vec{y})\}$ from [10] is a possible measure for the error occurring when objectives are omitted [9] and used in the definition of δ-minimum objective sets:

Definition 1: Two objective sets $\mathcal{F}_1$ and $\mathcal{F}_2$ are δ-nonconflicting iff $\preceq_{\mathcal{F}_1} \subseteq \preceq^{\delta}_{\mathcal{F}_2}$ and $\preceq_{\mathcal{F}_2} \subseteq \preceq^{\delta}_{\mathcal{F}_1}$.

Definition 2: An objective set $\mathcal{F}' \subseteq \mathcal{F}$ is called δ-minimum with respect to the set $\mathcal{F}$ iff (i) $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$, (ii) $\mathcal{F}'$ is δ'-nonconflicting with $\mathcal{F}$ for all $\delta' < \delta$, and (iii) there exists no $\mathcal{F}'' \subset \mathcal{F}$ with $|\mathcal{F}''| < |\mathcal{F}'|$ that also meets (i) and (ii).

In other words, a δ-minimum objective set is the smallest possible set of original objectives that preserves the original dominance structure except for an error δ. If we can compute a δ-minimum objective set with respect to all objectives and consider only the objectives in the δ-minimum set $\mathcal{F}' \subset \mathcal{F}$, we make an error of at most δ only in the following sense: whenever we wrongly assume that $\vec{x}$ weakly dominates $\vec{y}$ with respect to $\mathcal{F}'$ for two solutions $\vec{x},\vec{y} \in X$, we can assure, that $\vec{x}$ is at least δ-dominating $\vec{y}$ with respect to all objectives, i.e., $f_i(\vec{x}) - \delta \leq f_i(\vec{y})$ for all $f_i \in \mathcal{F} \setminus \mathcal{F}'$. The δ-minimum set approach has another important property: If a solution $\vec{x} \in X$ weakly dominates a solution $\vec{y} \in X$ according to the weak Pareto dominance relation $\preceq_{\mathcal{F}}$, then $\vec{x}$ will still weakly dominate $\vec{y}$ after the objective reduction—provided that the considered objectives represent a δ-minimum set with respect to all objectives. This means, that also the hypervolume indicator I_H [11], [12], [10] of the two solutions is computed qualitatively correct if one solution dominates another.

Algorithm 1 A greedy algorithm for `k-EMOSS`.

Init:
  $\mathcal{F}' := \emptyset$
while $|\mathcal{F}'| < \mathtt{k}$ **do**
  $\mathcal{F}' := \mathcal{F}' \cup \operatorname*{argmin}_{i \in \mathcal{F} \setminus \mathcal{F}'} \{\delta_{\min}(\mathcal{F}' \cup \{i\}, \mathcal{F}) \text{ w.r.t. } A\}$
end while

Algorithm 2 A greedy algorithm for δ-`MOSS`.

Init:
  compute the relations $\preceq_i$ for all $1 \leq i \leq k$ and $\preceq_{\mathcal{F}}$
  $\mathcal{F}' := \emptyset$
  $R := A \times A \setminus \preceq_{\mathcal{F}}$
while $R \neq \emptyset$ **do**
  $i^* =$
    $\operatorname*{argmin}_{i \in \mathcal{F} \setminus \mathcal{F}'} \{ |(R \cap \preceq_i) \setminus \left(\preceq^0_{\mathcal{F}' \cup \{i\}} \cap \preceq^{\delta}_{\mathcal{F} \setminus (\mathcal{F}' \cup \{i\})} \right)| \}$
  $R := (R \cap \preceq_{i^*}) \setminus \left(\preceq^0_{\mathcal{F}' \cup \{i^*\}} \cap \preceq^{\delta}_{\mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})} \right)$
  $\mathcal{F}' := \mathcal{F}' \cup \{i^*\}$
end while

The task of computing a δ-minimum objective set, given a set $A \subseteq X$ of solutions, was denoted in [9] as the δ-`MINIMUM OBJECTIVE SUBSET PROBLEM`, or δ-`MOSS` for short, and is known to be $\mathcal{NP}$-hard. The problem of finding an objective subset of predefined size `k` with minimum error according to $\mathcal{F}$ and the set $A \subseteq X$ was denoted as `MINIMUM OBJECTIVE SUBSET OF SIZE k WITH MINIMUM ERROR` (`k-EMOSS`) [9]. In [9], also simple greedy heuristics for both problems were proposed which are shown as Algorithm 1 for `k-EMOSS` and Algorithm 2 for δ-`MOSS`. Both algorithms are used in the following section for objective reduction during the search process.

III. Objective Reduction During Search

Two concrete problems might occur when many objectives are considered in multiobjective optimization scenarios. First, the computation effort needed to compute the hypervolume indicator I_H increases rapidly with the number of objectives [13], [4], [14], [5]. Second, the computation of the objective values itself in practice often needs much time, e.g., when simulations have to be run to evaluate a solution. With many objectives, the computation of the objective vectors may even become infeasible. In both cases, a limitation of the number of objectives would yield smaller running times of evolutionary algorithms, especially if they are based on the hypervolume indicator. However, it is not clear whether such an objective reduction affects the quality of the found solutions. Surely, a reduction yields faster evaluations and therefore more solutions can be evaluated. But on the other hand, an omission of objectives will cause a loss of information which might be useful during the search for good solutions. The following sections discuss different approaches of objective reduction and propose various objective reduction methods which are compared experimentally later on.

A. An Objective-Adaptive EA

The simplest possible objective reduction method is to decide in advance which of the k objectives are considered during the search. The decision can either be driven by preferences of a human decision maker or via a dimensionality reduction technique as PCA, applied to a set of randomly chosen solutions. The former approach has the drawback that it is often too little known about the problem such that deciding which objectives to take for optimization is difficult. Thus, we would prefer an automated technique independent of any human preference as the latter is. Nevertheless, there is still a drawback: it might be the case that in different stages of the optimization, different objectives are required to reduce the distance to the Pareto front. In terms of conflicting objectives, this would mean that an objective pair might be nonconflicting in the beginning, e.g., with respect to randomly drawn solutions, but near the Pareto front, the two objectives are conflicting implying that both objectives should be considered together to cover the entire Pareto front. Therefore, we limit the discussion to objective reduction methods applied during search.

One recently proposed approach is based on PCA and was intended to be used for problems with many objectives the Pareto front of which, however, has a lower dimension [7]. The authors present a procedure to extract from a solution set the objectives which preserve most of the objective correlation. The objective reduction is integrated into the Non-dominated Sorting Genetic Algorithm NSGA-II. After NSGA-II is run, the objective reduction procedure is applied to the algorithm's outcome and the algorithm is started again while optimizing only the objectives in the computed objective set. This loop of running NSGA-II and applying the objective reduction technique afterwards is repeated until the number of objectives cannot be further reduced. Although the approach was originally intended for high-dimensional problems with low-dimensional Pareto fronts, the objective reduction procedure can also be used to reduce the number of objectives in general. A problem of the correlation-based objective reduction is the unpredictable effect on the Pareto dominance relation. The relation-based objective reduction approach, discussed in Sec. II, can predict whether an error in the dominance relation occurs while omitting objectives. In the following, we will see, how this objective reduction approach can be integrated into a hypervolume based evolutionary algorithm to improve the quality of its outcomes.

The basis of all objective reduction methods, discussed below, is a simple indicator based evolutionary algorithm, namely SIBEA [15]. Algorithm 3 shows the organization of the original SIBEA, extended with a general objective reduction functionality. SIBEA starts with randomly choosing a set of μ solutions, the population P. Until a certain time limit T is reached, the μ solutions of the current population P are randomly selected for recombination and mutation, the variated solutions are inserted into the population, and the population of the next generation is determined by the following procedure: after a non-dominated sorting of the population, the non-dominated fronts are, starting with the best front, completely inserted into the new population until the size of the new population is at least μ. For the first front F the inclusion of which yields a population size larger than μ, the solutions $\vec{x}$ in this front with the smallest indicator loss $d(\vec{x}) := I(F) - I(F \setminus \{x\})$ are successively removed from the new population where the indicator loss is recalculated every time a solution is removed. In addition, SIBEA can apply various objective reduction strategies to improve the running time of the hypervolume computation. To this end, every G generations an objective reduction is performed, i.e., it is decided which objectives are chosen for optimization and which ones are neglected during the next G generations.

Algorithm 3 Simple Indicator Based Evolutionary Algorithm (**SIBEA**)

Input: population size μ; running time T in seconds; indicator function I; reduction frequency G in generations;
Output: approximation of Pareto-optimal set A;

Step 1 (Initialization):
Generate an initial set of decision vectors P of size μ; set the current time t_0; set generation counter $m := 0$.
Step 2 (Dimensionality reduction):
If $m \equiv 0 \bmod G$: Use the objective vectors of all solutions in P to decide which objectives to consider in the following G generations.
Step 3 (Environmental Selection):
Iterate the following three steps until the size of the population does no longer exceed μ:

1) Rank the population using Pareto dominance and determine the set of individuals $P' \subseteq P$ with the worst rank. Here, dominance depth [16] is used.
2) For each solution $\vec{x} \in P'$ determine the loss $d(\vec{x})$ w.r.t. the indicator I if it is removed from P', i.e., $d(\vec{x}) := I(P') - I(P' \setminus \{x\})$.
3) Remove the solution with the smallest loss $d(\vec{x})$ from the population P (ties are broken randomly).

Step 4 (Termination):
If T seconds expired since t_0 then set $A := P$ and stop; otherwise set $m := m + 1$.
Step 5 (Mating):
Randomly select elements from P to form a temporary mating pool Q of size μ. Apply recombination and mutation operators to the mating pool Q which yields Q'. Set $P := P + Q'$ (multi-set union) and continue with Step 2.

B. Adaptation Strategies

Various objective reduction strategies are described in the following paragraphs. We distinguish between approaches with a fixed size of the reduced objective set (Paragraph III-B.1), objective reduction methods with dynamically changing objective set sizes (Paragraph III-B.2) and algorithms changing the number of considered objectives adaptively with respect to changes in the hypervolume indicator (Paragraph III-B.3). All mentioned types of objective reduction within SIBEA have their counterpart: Instead of considering the reduced objective set solely, the aggregation-based versions consider an additional objective as an aggregation of the remaining objectives. Paragraph III-B.4 explains in detail this concept of objective reduction plus aggregation.

1) Fixed Objective Set Size: As the simplest objective reduction method, integrated into SIBEA, we fix the number $\mathtt{k}$ of considered objectives in advance. This allows to easily adjust the computation time of the algorithms, i.e., the smaller $\mathtt{k}$, the faster the hypervolume computation. On the other hand, it is not easy to control the quality of the computed Pareto front approximations by changing the parameter $\mathtt{k}$. A low $\mathtt{k}$ will often yield worse results with respect to the population's hypervolume. In preliminary experiments, a reduction to 2 to 5 objectives was reasonable. When considering more than 5 objectives, the hypervolume computation becomes too time-consuming while considering at least two objectives seems to be required for optimization.

Besides a random version, where the objective sets of fixed size $\mathtt{k} = 3$ are always chosen randomly, the greedy $\mathtt{k}$-EMOSS algorithm is used to compute an objective set with predefined size and the smallest possible δ-error. To circumvent high running times for the hypervolume computation within SIBEA, we set the number of considered objectives for the $\mathtt{k}$-EMOSS version to $\mathtt{k} = 3$, and $\mathtt{k} = 4$ respectively.

2) Dynamically Changing the Objective Set Size: To avoid the difficult choice of the objective set size in the methods described above, the three methods, described in this section, dynamically choose the number of considered objectives. Two methods choose the objective set size randomly. More precisely, a geometrically distributed random number with $p = 0.5$ is drawn as the number $\mathtt{k}$ of considered objectives and adjusted to $\mathtt{k} = k$ if $\mathtt{k} > k$. Thus, the expected objective set size is 2. We consider two different versions where the objectives are either chosen randomly or according to the greedy algorithm for $\mathtt{k}$-EMOSS. The third method chooses the objective set (and its size) according to a given δ error by applying Algorithm 2 on the current population. A drawback of this reduction method is the δ-error as additional parameter since it is not clear how to choose δ in advance. While the current population is scaled to $[0,1]^k$ before the objective vectors are used as input for Algorithm 2, we consider two versions with $\delta = 0.8$ and $\delta = 0.9$.

3) Adaptively Increasing Objective Sets: Starting with one objective, we increase in this type of objective reduction the objective set size adaptively every G generations, dependent on a hypervolume improvement. If and only if the hypervolume indicator of the entire population increased within the last G generations in at most $G/10$ generations, the objective set size is increased by one. The idea behind this is that SIBEA can optimize the selected objectives as long as it can improve the population easily. If the algorithm gets stuck, we increase the number of considered objectives to improve the hypervolume of the population and guarantee that high-dimensional Pareto-optimal fronts can be found by the algorithm. In contrast to the PCA-based objective reduction method in [7] where the objective sets become only smaller while converging to the Pareto-optimal front, here the opposite happens.

We distinguish between one random version which chooses the larger objective sets always randomly and a second version which uses Algorithm 1 to compute the new objective set. Note, that the objective set is not changed unless the number of objectives is increased.

4) Aggregation of Neglected Objectives: If only a subset of objectives is optimized as proposed above, the omitted objectives are, in general, not optimized simultaneously, i.e., the objective values of the neglected objectives can be arbitrary poor. We would expect that the values of neglected objectives are kind of randomly chosen due to the fact that they are not considered within selection. In order to avoid this behavior, we suggest to optionally aggregate the omitted objectives for all objective reduction methods by summing up the r remaining objective values[1], and using the sum divided by $1000 \cdot r$ as additional objective.

We will see in the comparison described in Sec. VI whether the aggregation is a useful extension. On the one hand, the consideration of the otherwise neglected objectives may increase the quality of the computed solution sets; on the other hand, an additional objective also increases the computation time for the hypervolume and therefore allows only fewer evaluations.

5) Reference Algorithms SIBEA, NSGA-II and SPEA2: In addition, we use SIBEA without any objective reduction to compare whether the proposed objective reduction methods can improve the hypervolume-based algorithm. For reference, we also use NSGA-II [16] and SPEA2 [17], implemented in the PISA framework [18], together with random search space samples of size μ.

IV. Test Problems DTLZ2_{BZ} and DTLZ7

To compare the different objective reduction methods, we use two test problems based on the DTLZ test suite of [19]. Many functions within the original test function suite, especially the often used DTLZ2 function, have the properties that

1) the projection of the Pareto front to $\mathtt{k} < k$ objectives collapses to one optimal point, i.e., when omitting arbitrary objectives, the search will always converge to one solution. The second drawback is that
2) when we optimize only a subset of $\mathtt{k} < k$ objectives, the neglected objectives are also optimized at the same time due to the usage of a single scaling function $g(\vec{x}_M)$ for all objectives.

To eliminate the mentioned properties, we modify the original DTLZ2 function in two ways. First, we limit the range of the decision variables x_i, i.e., we cut the corners of the non-dominated fronts to circumvent property 1). In detail, we

[1] If the objective reduction method chooses $\mathtt{k}$ objectives out of all k, the remaining $r := k - \mathtt{k}$ objectives are aggregated.

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = (1+g_1(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\cos(\theta_{k-1}), \\
\text{Min} & f_2(\mathbf{x}) = (1+g_2(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\sin(\theta_{k-1}), \\
\vdots & \vdots \\
\text{Min} & f_{k-1}(\mathbf{x}) = (1+g_{k-1}(\vec{x}))\cos(\theta_1)\sin(\theta_2), \\
\text{Min} & f_k(\mathbf{x}) = (1+g_k(\vec{x}))\sin(\theta_1), \\
\text{where} & g_i(\vec{x}) = \sum_{j=k+(i-1)\cdot\lfloor\frac{n-k+1}{k}\rfloor}^{k+i\cdot\lfloor\frac{n-k+1}{k}\rfloor-1} (x_i-0.5)^2 \\
& \qquad \text{for } i = 1,\ldots,k-1, \\
& g_k(\vec{x}) = \sum_{j=k+(k-1)\cdot\lfloor\frac{n-k+1}{k}\rfloor}^{n} (x_i-0.5)^2, \\
& \theta_i = \frac{\pi}{2}\cdot\left(\frac{x_i}{2}+\frac{1}{4}\right) \text{ for } i=1,\ldots,k-1 \\
& 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.
\end{array}
$$

Fig. 1. Definition of the modified DTLZ function DTLZ2_{BZ} with n decision variables and k objectives.

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = x_1, \\
\text{Min} & f_2(\mathbf{x}) = x_2, \\
\vdots & \vdots \\
\text{Min} & f_{k-1}(\mathbf{x}) = x_{k-1}, \\
\text{Min} & f_k(\mathbf{x}) = (1+g(\vec{x}_M))h(f_1,f_2,\ldots,f_{k-1},g), \\
\text{where} & g(\vec{x}_M) = 1 + \frac{9}{|\vec{x}_M|}\sum_{x_i\in\vec{x}_M} x_i, \\
& h(f_1,f_2,\ldots,f_{k-1},g) \\
& \qquad = k - \sum_{i=1}^{k-1}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right] \\
& 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.
\end{array}
$$

Fig. 2. Definition of the non-modified DTLZ7 function with n decision variables and k objectives. As in the original definition, $\vec{x}_M$ is defined as the last $n-k+1$ decision variables $x_k,\ldots,x_n$.

use $x_i/2+1/4$ instead of the decision variables x_i directly as in the original version. To come up with a problem where all single objectives have to be optimized simultaneously to reach the Pareto front, i.e., avoiding drawback 2), we, secondly, introduce one scaling function $g_i(\vec{x})$ for each objective, instead of one single scaling function $g(\vec{x}_M)$. Figure 1 shows the definition of the DTLZ2_{BZ} function.

Since the original DTLZ7 problem has both a Pareto front which does not collapse to a single point when projected and objective functions to which independent decision variables are associated with, we use the original version from [19] as the second test function in our comparison. Figure 2 shows the formal definition of DTLZ7.

In addition, we finally scale the objective values for both test functions, since in general not all objectives are equally scaled in practical problems. With the scaling

$$
f_i'(\vec{x}) := \begin{cases} \text{maxValue}\cdot\left(\frac{f_i(\vec{x})}{\text{maxValue}}\right)^i & \text{if } i \equiv 0 \bmod 2 \\ \text{maxValue}\cdot\left(\frac{f_i(\vec{x})}{\text{maxValue}}\right)^{1/i} & \text{otherwise} \end{cases},
$$

we change the importance of the different objectives, whereas $\text{maxValue} = 1 + ((n-k+1)/4)$ for DTLZ2_{BZ}, and $\text{maxValue} = 11k$ for DTLZ7. Near the Pareto-optimal front, objectives with even number have larger variances than objectives with odd number. Far from the Pareto-optimal front, odd and even objectives invert their behavior.

V. Experimental Setup

For the experimental comparison of the different objective reduction methods, 21 runs are performed for $T = 20$ minutes and each combination of algorithm and problem on identical linux machines (4 cores, 64bit architecture, 2.6GHz). All algorithms described in Sec. III and also listed in Table I are used. As test problems, we use the described DTLZ2_{BZ} and DTLZ7 functions with $n = 200$ and 5, 7, and 9 objectives each. The implementation of SIBEA is based on the hypervolume indicator in the PISA framework [18] and, thus, uses only a slow hypervolume algorithm, instead of state-of-the-art approaches like in [5]. The implementations of NSGA-II and SPEA2 are also taken from the PISA framework. All algorithms are used with standard settings,

TABLE I
All algorithms used in the experiments.

algorithm	objective set size	objective reduction method	aggregation
SIBEA	complete	—	no
SIBEA	$k = 3$ fix	k-EMOSS	yes/no
SIBEA	$k = 4$ fix	k-EMOSS	yes/no
SIBEA	$k = 3$ fix	random	yes/no
SIBEA	dynamic	0.8-MOSS	yes/no
SIBEA	dynamic	0.9-MOSS	yes/no
SIBEA	dynamic	random	yes/no
SIBEA	dynamic	k-EMOSS	yes/no
SIBEA	adaptive	random	yes/no
SIBEA	adaptive	k-EMOSS	yes/no
NSGA-II	complete	—	no
SPEA2	complete	—	no

whereas the population size is always set to $\mu = 50$ and the objective reduction frequency G equals 50.

To analyze the quality of the produced Pareto front approximations, we compute for all runs the hypervolume indicator of the first generation after the predefined time of $T = 20$ minutes elapses. Note, that for some of the algorithms, e.g., SIBEA without any objective reduction on the high-dimensional problems, the considered generation was completed rather after a day than right after 20 minutes. The reference points for the hypervolume computation are chosen as $(50,\ldots,50)$ for DTLZ2_{BZ} and $(100,\ldots,100)$ for DTLZ7 and the hypervolume is to be maximized.

For comparing the different algorithms, hypotheses have been derived from preliminary experiments which are tested on the performed 21 independent runs. As statistical test, we use the Mann-Whitney test implemented in the MATLAB toolbox (Version 7.1.0, see www.mathworks.com). The Mann-Whitney test is a nonparametric test to confirm the hypothesis that one random variable "systematically" produces larger values than another by ranking all values and comparing the rank sums for both samples. Since one and the same sample is used within multiple tests, we use a Bonferroni correction of the significance level to $0.05/78 \approx 6.4\cdot 10^{-4}$, as we test the hypothesis for all 78 combinations of algorithms within Fig. 5–10 except the random sample.

VI. RESULTS

Figures 5–10 show boxplots of the hypervolume indicators derived from the first completed generation after $T = 20$ minutes for all algorithms in Table I on all six problems. For clarity, the results for SIBEA with dynamic objective reduction based on the greedy δ-MOSS algorithm and the adaptive SIBEA versions are omitted. In the following paragraphs, we discuss the main results of the algorithm comparison.

A. SIBEA vs. NSGA-II vs. SPEA2

The experiments with SIBEA confirm, that hypervolume based evolutionary algorithms are sensitive to the number of objectives, i.e., the running time for the hypervolume indicator computation highly depends on the number of objectives. Up to four objectives are manageable with the used population size of 50, whereas the computation time explodes to more than a day per generation for the 9-objective problems. In the allowed time interval of 20 minutes for example, SIBEA managed to reach not more than 16, 5, and 1 generations for DTLZ2_{BZ} with 5, 7, and 9 objectives respectively. Nevertheless, the improvement in the hypervolume indicator values in the first generations is high compared to NSGA-II or SPEA2, cf. Fig. 3 and 4.

This observation leads to the assumption that the hypervolume indicator provides additional information on the search space; the information gain per objective vector evaluation is increased compared to the usage of the weak dominance relation within NSGA-II and SPEA2. If the hypervolume computation can be accelerated, one may expect that an improvement is possible also with respect to a predefined running time as it is the case for our comparison. Since the baseline hypervolume algorithm used within SIBEA is not state-of-the-art, we cannot expect that SIBEA can compete with NSGA-II and SPEA2 compared with respect to a given time interval. However, a realistic comparison of SIBEA with NSGA-II and SPEA2 was never intended to be the focus of this paper; in fact, we emphasize the difference between SIBEA with and without objective reduction strategies.

1) Remarks on the Comparison NSGA-II vs. SPEA2: Since NSGA-II and SPEA2 are not optimizing the hypervolume indicator directly, the fact that the indicator values can increase during the search is not astonishing. While the algorithms themselves improve their population permanently, e.g., with respect to diversity, the populations of consecutive generations will often be incomparable. Therefore, the hypervolume indicator can decrease over time as can be seen in Fig. 3 and 4.

That NSGA-II obtains significantly better results than SPEA2 for all problems except DTLZ7 with 7 and 9 objectives is not surprising if we recall the conditions of the comparison: the available time is the same for all algorithms. Due to more complex computations within SPEA2, it takes more time to perform an entire step of SPEA2 than a generation lasts within NSGA-II. That means, NSGA-II is able to perform more generations than SPEA2 within the same time interval. As NSGA-II is, e.g., able to reach generations of 1500 and more within 20 minutes on DTLZ2_{BZ}, SPEA2 cannot process more than 860 generations within the same time.

B. Objective Reduction with Fixed Objective Set Size

The objective reduction strategies with fixed k turn out to be the best algorithms in the comparison (always significantly better than SPEA2, NSGA-II, and SIBEA without objective reduction and nearly always better than the other methods). Comparing the random version and the one based on the greedy heuristic, the latter is always significantly better on DTLZ2_{BZ}. On DTLZ7, no trend is visible because all three methods come close to the Pareto front. When compared with the aggregation versions, the random and the heuristic based strategy behave completely different. While an additional objective improves the random version on all DTLZ2_{BZ} problems, the greedy algorithm apparently chooses the right objectives on its own; the aggregation objective cannot help but instead makes the hypervolume computation too costly. For example, on DTLZ2_{BZ} with 9 objectives, the random version reaches generation 2081 without aggregation ($-$A) and 1618 with aggregation ($+$A), the 3-EMOSS based strategy reaches just the generations 1770 ($-$A) and 1092 ($+$A) and the 4-EMOSS based algorithm only the generations 436 ($-$A) and 56 ($+$A).

This observation again confirms the above mentioned sensitivity of the hypervolume computation on the number of objectives. A promising alternative—which has to be investigated in future work—would be to simply add the aggregated value to the computed hypervolume indicator value instead of using the aggregation as an additional objective.

C. Dynamic Objective Reduction

The dynamic objective reduction strategies show mixed performance. What we can conclude is that in many cases, the aggregation version yields worse hypervolume indicator values. It is also noticeable that the dynamic SIBEA versions show a high variance: with the randomly chosen objective set sizes, the dynamic strategies can at any time choose too many objectives such that the hypervolume computation will last too long to achieve competitive indicator values. A third disadvantage of the dynamic objective reduction strategies can be seen in Fig. 3 and 4: the quality of the solutions highly depends on the choice of the objective set size which changes every G generations. Thus, we advise against using the dynamic objective reduction strategies and prefer the k-EMOSS based strategies with fixed size and therefore predictable running time.

D. Objective Reduction Methods Omitted in Plots

Since the remaining objective reduction methods, namely adaptive and dynamic with predefined δ error, perform equally or even worse than SIBEA without any objective reduction, we do not show the results graphically due to space limitations and clarity in the plots. In addition, the indicator values for these objective reduction strategies show extremely high variances in comparison to the other methods.

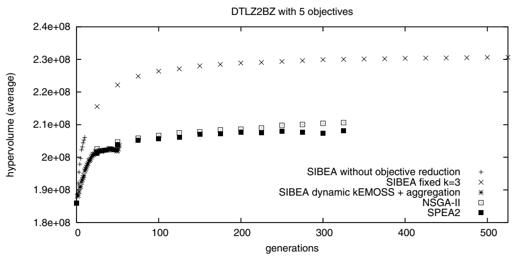

Fig. 3. Course of the averaged hypervolume indicator values for selected algorithms over time on $DTLZ2_{BZ}$ with 5 objectives. Note, that on the x-axis, the number of generations is plotted and not the time in seconds. Therefore, the algorithms reach different numbers of generations within the same time interval. Only the generations, all 21 runs reached, are plotted.

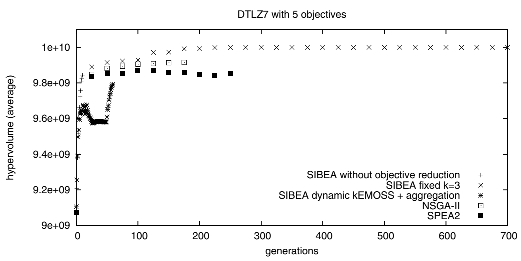

Fig. 4. Course of the averaged hypervolume indicator values for selected algorithms over time on DTLZ7 with 5 objectives. As in Fig. 3, only the generations, all 21 runs reached, are plotted.

One possible reason for the poor results of the dynamic version with given δ error might be a sensitive parameter δ which is hard to adjust. The adaptive SIBEA versions, on the other hand, tend to get stuck in the region which is found when only the first objective is optimized. The only way out would be to optimize always at least two objectives; the conformation, however, would be future work.

VII. Conclusions

In this paper, we have investigated how objective reduction can be integrated into hypervolume-based evolutionary algorithms as SIBEA to save computation time during the hypervolume computation which allows the algorithm to evaluate more solutions within the same time. Various objective reduction strategies choosing the number of considered objectives dynamically, adaptively or fix have been proposed and compared experimentally on two chosen test functions with up to 9 objectives.

It turns out that the SIBEA version without any objective reduction can be improved drastically by incorporation of an objective reduction method based on the greedy `k-EMOSS` algorithm into SIBEA. Also in comparison to NSGA-II and SPEA2, this objective reduction based algorithm beats the state-of-the-art algorithms in terms of the hypervolume indicator significantly. Due to the sensitivity of the hypervolume computation with respect to the number of objectives, the proposed strategy of reducing the number of objectives and aggregating the remaining objectives in addition, cannot further improve the proposed hypervolume based algorithms in general. Nevertheless, the extensive experiments show that further improvements of the proposed algorithms seem possible in the future.

References

[1] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization," in *Evolutionary Multi-Criterion Optimization (EMO 2007)*, ser. LNCS, vol. 4403. Springer, 2007, pp. 742–756.
[2] M. Emmerich, N. Beume, and B. Naujoks, "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," in *Evolutionary Multi-Criterion Optimization (EMO 2005)*, ser. LNCS, vol. 3410. Springer, 2005, pp. 62–76.
[3] L. While, "A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume," in *Evolutionary Multi-Criterion Optimization (EMO 2005)*, ser. LNCS, vol. 3410. Springer, 2005, pp. 326–340.
[4] L. While, P. Hingston, L. Barone, and S. Huband, "A Faster Algorithm for Calculating Hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.
[5] N. Beume and G. Rudolph, "Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem," Sonderforschungsbereich 531 Computational Intelligence, Universität Dortmund, Tech. Rep. CI-216/06, 2006.
[6] H. Liu and H. Motoda, Eds., *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, 1998.
[7] K. Deb and D. Saxena, "Searching For Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems," in *Congress on Evolutionary Computation (CEC 2006)*. IEEE Press, 2006, pp. 3352–3360.
[8] I. T. Jolliffe, *Principal component analysis*. Springer, 2002.
[9] D. Brockhoff and E. Zitzler, "Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization," in *Parallel Problem Solving from Nature (PPSN IX)*, ser. LNCS, vol. 4193. Springer, 2006, pp. 533–542.
[10] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
[11] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," in *Parallel Problem Solving from Nature (PPSN V)*, 1998, pp. 292–301.
[12] ——, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[13] L. While, L. Bradstreet, L. Barone, and P. Hingston, "Heuristics for Optimising the Calculation of Hypervolume for Multi-objective Optimisation Problems," in *Congress on Evolutionary Computation (CEC 2005)*. IEEE Press, 2005, pp. 2225–2232.
[14] C. M. Fonseca, L. Paquete, and M. L.-I. nez, "An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator," in *Congress on Evolutionary Computation (CEC 2006)*. IEEE Press, 2006, pp. 1157–1163.
[15] E. Zitzler, D. Brockhoff, and L. Thiele, "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration," in *Evolutionary Multi-Criterion Optimization (EMO 2007)*, ser. LNCS, vol. 4403. Springer, 2007, pp. 862–876.
[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," in *Parallel Problem Solving from Nature (PPSN VI)*, ser. LNCS, vol. 1917. Springer, 2000, pp. 849–858.
[17] 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 2001)*, 2002, pp. 95–100.
[18] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA — A Platform and Programming Language Independent Interface for Search Algorithms," in *Evolutionary Multi-Criterion Optimization (EMO 2003)*, ser. LNCS, vol. 2632. Springer, 2003, pp. 494–508.
[19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. Springer, 2005, pp. 105–145.

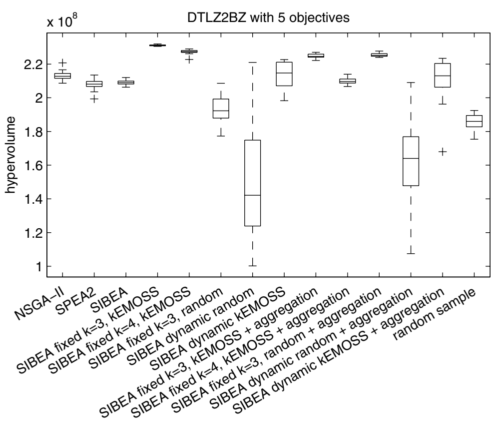

Fig. 5. Boxplot comparing selected algorithms on $\mathrm{DTLZ2_{BZ}}$ with 5 objectives. The hypervolume is upper bounded by $3.125 \cdot 10^8$.

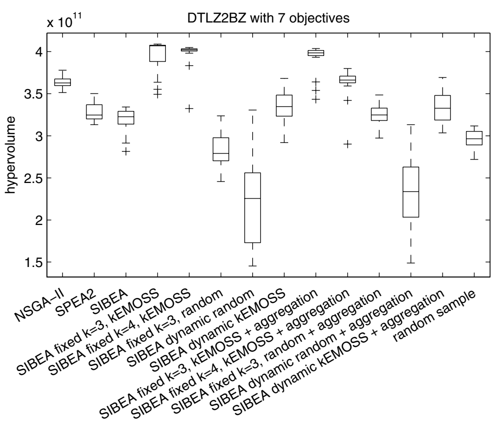

Fig. 6. Boxplot comparing selected algorithms on $\mathrm{DTLZ2_{BZ}}$ with 7 objectives. The hypervolume is upper bounded by $7.8125 \cdot 10^{11}$.

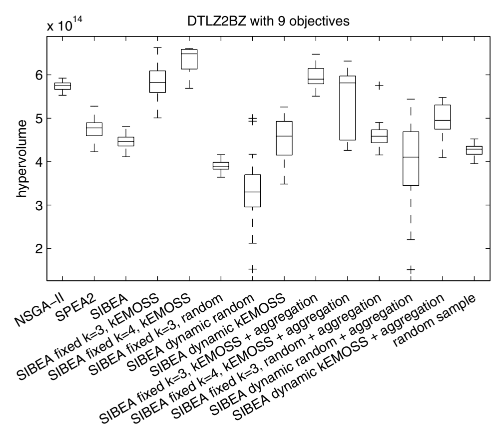

Fig. 7. Boxplot comparing selected algorithms on $\mathrm{DTLZ2_{BZ}}$ with 9 objectives. The hypervolume is upper bounded by $1.954 \cdot 10^{15}$.

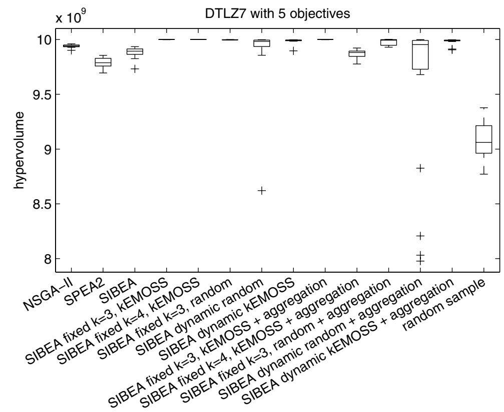

Fig. 8. Boxplot comparing selected algorithms on DTLZ7 with 5 objectives. The hypervolume is upper bounded by $1 \cdot 10^{10}$.

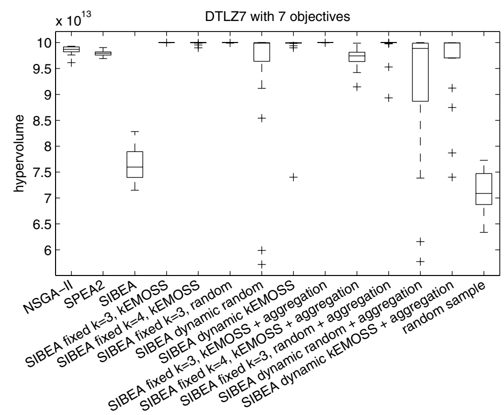

Fig. 9. Boxplot comparing selected algorithms on DTLZ7 with 7 objectives. The hypervolume is upper bounded by $1 \cdot 10^{14}$.

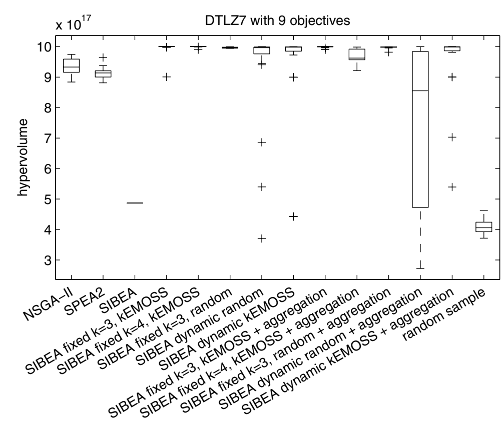

Fig. 10. Boxplot comparing selected algorithms on DTLZ7 with 9 objectives. The hypervolume is upper bounded by $1 \cdot 10^{18}$. Note, that for SIBEA without objective reduction, the computation of the hypervolume was too expensive, such that only one run was completed, needing 6435 minutes or more than 4 days to complete the first generation.