Abstract—The majority of empirical comparisons of multi-objective evolutionary algorithms (MOEAs) are performed on synthetic benchmark functions. One of the advantages of synthetic test functions is the a-priori knowledge of the optimal Pareto front. This allows measuring the proximity to the optimal front for the solution sets returned by the different MOEAs. Such a comparison is only meaningful if the cardinality of all solution sets is bounded by some fixed $k$. In order to compare MOEAs to the theoretical optimum achievable with $k$ solutions, we determine best possible $\varepsilon$-indicator values achievable with solution sets of size $k$, up to an error of $\delta$. We present a new algorithm with runtime $O(k \cdot \log^2(\delta^{-1}))$, which is an exponential improvement regarding the dependence on the error $\delta$ compared to all previous work. We show mathematical correctness of our algorithm and determine optimal solution sets for sets of cardinality $k \in \{2, 3, 4, 5, 10, 20, 50, 100, 1000\}$ for the well known test suits DTLZ, ZDT, WFG and LZ09 up to error $\delta = 10^{-25}$.

I. INTRODUCTION

Biobjective optimization aims at minimizing a fitness function $F : X \rightarrow \mathbb{R}^2$ which maps from some arbitrary search space $X$ to pairs of real numbers, which describe two (potentially conflicting) objective functions $F_1$ and $F_2$. This is typically done by finding a solution set $S \subseteq X$ such that (i) $S$ approximates well all Pareto optimal points in the objective space $F(X)$ and (ii) $S$ is not too large, say of some fixed size $k = |S|$. We can then compare the solution sets obtained by different biobjective optimization algorithms with a unary indicator function $I : X^k \rightarrow \mathbb{R}$, which maps a solution set to a single real number. Examples for commonly used indicators are the hypervolume indicator, the $\varepsilon$-indicator, generational distance [27], inverted generational distance [11], and several other [29]. The progress of the optimization can then be visualized by plotting this indicator as a function of time (generations or number of fitness evaluations).

For single-criteria test functions with known optima it is very common to present a comparison between different optimizers by measuring the distance to the optimum of the solutions obtained by the different optimizers. For bi- and multi-objective problems this would mean measuring the difference between the optimal solution set with perfect indicator value and the indicator value of the solution sets obtained by different optimizers. As optimal indicator values are typically hard to obtain for bi- and multi-objective problems, only very few comparisons of multi-objective optimizers present their results with respect to the achieved distance to the optimum (e.g. [8, 23]).

What is missing are reliable optimal indicator values for typical test functions, with varying dimension $d$ and solution set size $k$. For the hypervolume indicator, the former website of Eckart Zitzler’s group at ETH [30] presents close-to-optimal hypervolume distributions for the ZDT and DTLZ benchmark suites for $d = 2$ and $k = 2, 3, 4, 5, 10, 20, 50, 100, 1000$. They are obtained by the Maximize function of Maple's Optimization package and are therefore not guaranteed to be optimal. Last year, Glasmachers [16] obtained slightly larger (=better) hypervolume indicator values than [30] for the same set of test functions and additionally presented fronts for $d = 3$ achieved by a gradient-based optimizer. They are again not proven to be optimal. There are a few theoretical results on optimal hypervolume distributions [3, 4, 9, 15], but obtaining provably-optimal distributions for non-trivial test functions is still an open problem.

We do not consider the hypervolume indicator, but instead study the $\varepsilon$-indicator for $d = 2$ dimensions. It measures by how much we have to shift all points of a solution set $S$ in both objectives such that all feasible points in $F(X)$ are dominated. In operations research and theory, this is the most popular measure for multi-criteria optimization [10, 12, 14, 18, 21, 22, 25]. It is also popular in evolutionary multiobjective optimization [6, 19, 26]. The $\varepsilon$-indicator allows us to compare the approximation quality of the solution sets returned by different multi-objective evolutionary algorithms (MOEAs). Our aim is finding the best possible point set of some fixed size $k$, which maximizes the $\varepsilon$-indicator on a fixed biobjective test function $F$. More formally, we consider the following problem. We assume that we are given the Pareto front $f$ of the biobjective optimization problem $F$, for details see Section II.

Optimal $\varepsilon$-indicator problem (OPTEPS): For a given biobjective Pareto front $f$, solution set size $k$ and error bound $\delta$, determine the best possible $\varepsilon$-indicator achievable with solution sets of size $k$, up to an additive error of $\delta$. 

Efficient Computation of Two-dimensional Solution Sets Maximizing the Epsilon-Indicator

Karl Bringmann
Institute of Theoretical Computer Science
ETH Zurich
Zurich, Switzerland

Tobias Friedrich
Chair of Theoretical Computer Science
Friedrich-Schiller-Universität Jena
Jena, Germany

Patrick Klitzke
Department of Computer Science
Universität des Saarlandes
Saarbrücken, Germany
We say that set time. Specifically, we assume that we can evaluate error $\delta$ tractable, which makes error bounds of $\Theta(1)$ or $O(\log^{2}(\delta^{-1}))$. More generally, any algorithm following this approach needs runtime $\Omega(\delta^{-1})$. We present an exponential improvement regarding the dependence on the error $\delta$ and present an algorithm with runtime $O(k \cdot \log^{2}(\delta^{-1}))$, which makes errors of $\delta = 10^{-25}$ and smaller easily tractable.

II. PRELIMINARIES

A. Input Assumptions

We assume that all Pareto optimal points $z \in X$ have $F_{1}(z), F_{2}(z) \in [0, 1]$ as input we are given the Pareto front function $f: [0, 1] \rightarrow [0, 1]$ with

$$f(x) = \inf \{F_{2}(z) \mid z \in X \text{ and } F_{1}(z) \leq x\}.$$

Specifically, we assume that we can evaluate $f(x)$ in constant time.

B. Problem Definition

Given a Pareto front $f$ as above and a number $k$, for any set $P \subset [0, 1]$ of size $k$ its $\varepsilon$-indicator is defined as

$$\mathcal{I}_{\varepsilon}(P, f) = \sup_{x \in [0, 1]} \min_{p \in P} \max\{p - x, f(p) - f(x)\}.$$  

We say that $(p, f(p))$ $\varepsilon$-approximates $(x, f(x))$ (or simply $p\varepsilon$-approximates $x$) if $\max\{p - x, f(p) - f(x)\} \leq \varepsilon$. We consider the optimal $\varepsilon$-indicator problem of computing

$$\text{OPT\-EPS} := \inf_{P \subset [0, 1], |P| = k} \mathcal{I}_{\varepsilon}(P, f).$$

We remark that OPT\-EPS is not necessarily attained at some set $P$, but for any $\varepsilon > 0$ there is a set $P$ with $\mathcal{I}_{\varepsilon}(P, f) \leq \text{OPT\-EPS} + \varepsilon$.

We will solve an approximation version of this problem, where we are additionally given an error parameter $\delta$ and want to compute a set $P \subset [0, 1]$ of size $k$ with

$$\mathcal{I}_{\varepsilon}(P^{*}, f) \leq \mathcal{I}_{\varepsilon}(P, f) \leq \mathcal{I}_{\varepsilon}(P^{*}, f) + \delta.$$

C. Justification of Input Assumptions

Let us argue why our input assumptions are justified. For typical test problems one can analytically determine a front function $f$ describing the best possible objective value $F_{2}$ over all points with a fixed objective value $F_{1}$, i.e.,

$$\hat{f}(x) = \inf \{F_{2}(z) \mid z \in X \text{ and } F_{1}(z) = x\},$$

and $\hat{f}$ is defined on an interval $[a, b]$ consisting of (a superset of) all possible values of $F_{1}(z), z \in X$. If $f$ is monotonic decreasing, then it is the desired Pareto front $f$.

However, a front function $\hat{f}$ might not be monotone, see Figure 1. We argue that then we can derive an approximation $\hat{\hat{f}}$ of the true Pareto front $f$ if $\hat{f}$ is sufficiently well-behaved. Assume that (1) $\hat{f} : [a, b] \rightarrow [c, d]$, (2) $\hat{f}$ has exactly $\ell$ local minima and we know disjoint intervals $I_{1}, \ldots, I_{\ell}$ each containing one local minimum, (3) $\hat{f}$ has curvature bounded by $C$, and (4) $a, b, c, d, \ell, C$ are fixed, i.e, we may omit them in $O$-notation. Let $x_{1}, \ldots, x_{\ell} \in I$ be the local true minima of $f$. (For an illustration see Figure 1.) Then the true Pareto front function $f : [a, b] \rightarrow [c, d]$ of a front function $\hat{f}$ is given by

$$f(x) := \min \{\hat{f}(x), \min_{1 \leq i \leq \ell} \{\hat{f}(x_{i}) \mid x_{i} \leq x\}\}.$$

Thus, we can construct $f$ from $\hat{f}$ if we know $x_{1}, \ldots, x_{\ell}$. However, our assumptions allow to compute an approximation $\hat{\hat{f}}_{i}$ of $\hat{f}$ such that $\hat{f}(x_{i}) \leq \hat{\hat{f}}_{i}(x_{i}) \leq \hat{f}(x_{i}) + \delta$ for all $i$. This can be done using ternary search, which runs in time $O(\log^{3}(\delta^{-1}))$ (using assumptions (3) and (4)). Replacing $x_{i}$ by $\hat{\hat{f}}_{i}$ in the definition of $f$ yields an approximation $\hat{f}$ of $f$ such that $f(x) \leq \hat{f}(x) \leq f(x) + \delta$ for all $x$. It can be shown that OPT\-EPS of $f$ and $\hat{f}$ differ by at most $\delta$. Moreover, the resulting representation of $\hat{f}$ allows to evaluate $f(x)$ in constant time (since we assume that $\ell$ is constant and $\hat{f}$ can be evaluated in constant time).

Finally, after translation and scaling we can assume that $[a, b] = [0, 1]$, so that we indeed end up with a Pareto front $\hat{f} : [0, 1] \rightarrow [0, 1]$ that can be evaluated in constant time, as we assume for our algorithm.

Note that the simplification from this section introduces an error of $\delta$, which adds to the error $\delta$ of our algorithm presented in the next section. Choosing $\delta := \delta'/2$ yields a total error of $2\delta = \delta'$.

III. ALGORITHM

A. Discretization

Consider a Pareto front $f : [0, 1] \rightarrow [0, 1]$, size $k$, and error bound $\delta$. Assume without loss of generality that $\delta^{-1} \in \mathbb{N}$, otherwise decrease $\delta$ slightly to satisfy this requirement. For approximation purposes, we discretize the domain and range of $f$ to $\Delta := \{0, \delta, 2\delta, \ldots, 1 - \delta, 1\}$. That is, instead of considering points $x \in [0, 1]$ we restrict them to $\Delta \subset [0, 1]$, and we let $f^{\delta}(x)$ be $f(x)$ rounded up to a multiple of $\Delta$. We obtain a function $f^{\delta}: \Delta \rightarrow \Delta$. This yields an approximate version of the $\varepsilon$-indicator as

$$\mathcal{I}^{\delta}_{\varepsilon}(P, f^{\delta}) := \max_{x \in \Delta} \min_{p \in P} \max\{p - x, f^{\delta}(p) - f^{\delta}(x)\}.$$
We remark that one can compute $\text{APXEPS}^{(\delta)}$ by solving the \(\varepsilon\)-indicator subset selection problem on \([(x,f^{(\delta)}(x)) \mid x \in \Delta]\) (with the same reference set). This yields a running time of $O(\delta^{-1}\log(\delta^{-1}))$ [7]. We design an exponentially faster algorithm with respect to $\delta^{-1}$ in the next section.

### B. Algorithm Description

We will now present an exact algorithm for $\text{APXEPS}^{(\delta)}$, which is also an approximation algorithm for $\text{OPTEPS}$ by Lemma 1. It runs in time $O(k\log^2(\delta^{-1}))$.

Assume that we can check for any given $\varepsilon \in \Delta$ whether $\text{APXEPS}^{(\delta)} \leq \varepsilon$; we call this check an $\varepsilon$-test. From its definition one can see that $\text{APXEPS}^{(\delta)}$ is restricted to be a multiple of $\delta$ between 0 and 1. Thus, if we can perform an $\varepsilon$-test in time $T(k,\delta)$, then we can exactly compute $\text{APXEPS}^{(\delta)}$ by a binary search over $\varepsilon \in \Delta$ in time $O(\log(|\Delta|) T(k,\delta)) = O(\log(\delta) T(k,\delta))$. See Algorithm 1.

### Algorithm 1: approximation algorithm for OPTEPS in 2 dimensions, $O(k\log^2(\delta^{-1}))$ time

**Input:** integer $k$, Pareto front $f: [0,1] \rightarrow [0,1]$, desired precision $\delta$

1. Binary search for the minimal $\varepsilon \in \Delta = \{0,\delta,2\delta,\ldots,1\}$ with $\text{APXEPS}^{(\delta)} \geq \varepsilon$, which we decide using $\varepsilon$-tests (see Algorithm 2).
2. Return $\varepsilon$.

Hence, it remains to show that we can perform $\varepsilon$-tests in time $O(k\log(\delta^{-1}))$. To this end, we use a modified version of the $\varepsilon$-test described in [7]. Using their original description, the discretized front $\{(x,f^{(\delta)}(x)) \mid x \in \Delta\}$ would be explicitly stored, resulting in a time complexity of $\Omega(\delta^{-1})$. Instead, we implicitly store the discretized front, which is possible because the domain $\Delta$ is of an easy form and $f^{(\delta)}(x)$ can be evaluated in constant time.

### Algorithm 2: $\varepsilon$-test that checks whether $\text{APXEPS}^{(\delta)} \leq \varepsilon$, $O(k\log(\delta^{-1}))$ time

**Input:** integer $k$, Pareto front $f: [0,1] \rightarrow [0,1]$, error parameter $\delta$, value $\varepsilon \in \Delta$

1. $P := \emptyset$
2. $r := 1$
3. for $\ell \Leftarrow 1$ to $k$ do
4. let $x \in \Delta$ be the minimal with $f^{(\delta)}(x) \leq f^{(\delta)}(r) + \varepsilon$
5. $P := P \cup \{x\}$
6. $r := x - \varepsilon - \delta$
7. if $r < 0$ then return “$\text{APXEPS}^{(\delta)} \leq \varepsilon$”
8. return “$\text{APXEPS}^{(\delta)} > \varepsilon$”

The algorithm iteratively chooses a set of points $P$, see Algorithm 2. We denote by $r \in \Delta$ the rightmost point on the front which is not yet $\varepsilon$-approximated by the chosen points.

In any iteration, we choose the leftmost point $x \in \Delta$ that can
Figure 2: $\varepsilon$-optimal point distribution for DTLZ1, WFG3.

Figure 3: $\varepsilon$-optimal point distribution for DTLZ2–4, WFG4–9.
Figure 4: $\varepsilon$-optimal point distribution for DTLZ7.

Figure 5: $\varepsilon$-optimal point distribution for WFG1.
Figure 6: $\varepsilon$-optimal point distribution for WFG2.

Figure 7: $\varepsilon$-optimal point distribution for ZDT1, ZDT4.
Figure 8: $\epsilon$-optimal point distribution for ZDT2.

Figure 9: $\epsilon$-optimal point distribution for ZDT3.
sets of fixed cardinality $\varepsilon$-approximate $r$, which amounts to the constraint $f^{(\delta)}(x) \leq f^{(\delta)}(r) + \varepsilon$. This point $x$ can be easily found by a binary search over $\Delta$. The algorithm then updates $r$ by deleting all the points that are $\varepsilon$-approximated by the chosen point $x$, which are all points $y \in \Delta$ with $x - \varepsilon \leq y \leq r$, i.e., we set $r := x - \varepsilon - \delta$. We repeat this step until all points on the front are $\varepsilon$-approximated by the chosen points, i.e., $r < 0$, or the number of iterations exceeds $k$, in which case $\varepsilon$ was to small.

The correctness is a direct consequence of the $\varepsilon$-test of [7]. Essentially, we argue that some point that $\varepsilon$-approximates $(r, f^{(\delta)}(r))$ has to be chosen, and the point $x$ that we choose is such that it approximates a superset of all other such points, i.e., it is the best possible point approximating $(r, f^{(\delta)}(r))$.

### C. Runtime

The binary search for $\varepsilon$ in Algorithm 1 takes $O(\log(\delta^{-1}))$ steps, i.e., $O(\log(\delta^{-1}))$ invocations of an $\varepsilon$-test. In each loop of the $\varepsilon$-test (Algorithm 2) we perform a binary search taking $O(\log(|\Delta|)) = O(\log(\delta^{-1}))$ time. Since the loop is executed at most $k$ times, we obtain an overall time complexity of $O(k \log(\delta^{-1}))$ for the $\varepsilon$-test and $O(k \log^2(\delta^{-1}))$ for Algorithm 1.

### IV. Experiments

Our experiments have been conducted on a Mac Book Air with 1.7 GHz Intel Core i7 using the GNU Multiple Precision Arithmetic Library [1] and the MPFR C++ library [2]. We approximated the optimal $\varepsilon$-indicator OPTEPS values up to an error of $\delta = 10^{-25}$ for the well known test suite DTLZ [13], ZDT [28], WFG [17] and LZ09 [20] and for solution set sizes $k \in \{2, 3, 4, 5, 10, 20, 50, 100, 1000\}$. Some of the results are shown in Figures 2–9. Note that our plots show the front function $f$ and not the Pareto front $f$, see Figure 1 for the difference. As a reference for the description of the front functions we used [5]. All results, plots, and the complete source code can be found at http://docs.theinf.uni-jena.de/code/epssubset.zip.

### V. Conclusions

We present the first efficient algorithm for finding solutions sets of fixed cardinality $k$ which maximize the $\varepsilon$-indicator. Our new algorithm is proven to have an asymptotic runtime of $O(k \log^2(\delta^{-1}))$ for an error bound of $\delta$. This is an exponential improvement over all previous algorithms which have to select the optimal subset from equidistantly chosen points and have a runtime of $\Omega(1/\delta)$. We implemented the algorithm, make the source code available, and provide optimal point sets of varying sizes for several common benchmark suites.

Our work enables practitioners to plot the distance to optimum when presenting experiments on EMOAs on these test functions. This allows to judge whether an EMOA eventually gets stuck in local optima or not: If we observe that an EMOA on a test problem $F$ with population size $k$ converges to a value $v$ (i.e., its $\varepsilon$-indicator converges to $v$), then we can now check whether $v$ is indeed optimal for $F$ and $k$.

Finding optimal solution sets of bounded cardinality is a notoriously difficult problem. We could only present an efficient algorithm for the $\varepsilon$-indicator in two dimensions. It is an open problem how to obtain similar results in higher dimensions and for other quality measures like the hypervolume indicator. In dimension three and above already the subset selection problem [7, 16, 24, 26] is open for the hypervolume and $\varepsilon$-indicator.

### References


