Indicator-Based Selection in Multiobjective Search

Eckart Zitzler and Simon Künzli*

Swiss Federal Institute of Technology Zurich
Computer Engineering and Networks Laboratory (TIK)
Gloriastrasse 35, CH–8092 Zürich, Switzerland
{zitzler,kuenzli}@tik.ee.ethz.ch

Abstract. This paper discusses how preference information of the decision maker can in general be integrated into multiobjective search. The main idea is to first define the optimization goal in terms of a binary performance measure (indicator) and then to directly use this measure in the selection process. To this end, we propose a general indicator-based evolutionary algorithm (IBEA) that can be combined with arbitrary indicators. In contrast to existing algorithms, IBEA can be adapted to the preferences of the user and moreover does not require any additional diversity preservation mechanism such as fitness sharing to be used. It is shown on several continuous and discrete benchmark problems that IBEA can substantially improve on the results generated by two popular algorithms, namely NSGA-II and SPEA2, with respect to different performance measures.

1 Motivation

In a multiobjective scenario, the goal of the optimization process is often to find a good approximation of the set of Pareto-optimal solutions. The difficulty, though, is that there is no general definition of what a good approximation of the Pareto set is. Each particular definition represents specific preference information that depends on the user. For instance, one could formalize the goal as maximizing the hypervolume of the objective space dominated by the resulting approximation (cf. [11, 18]). In certain scenarios this definition may be appropriate, in others it can be inappropriate because the goal of the optimization process may vary for each decision maker and problem.

In the light of this discussion, one may reconsider the criteria that guided the design of multiobjective evolutionary algorithms (MOEAs) in the last decade. We make two observations here:

1. The basis of most MOEAs is the assumption that there are two conflicting goals: (i) to minimize the distance to the Pareto-optimal set, and (ii) to maximize the diversity within the approximation of the Pareto-optimal set [3]. However, recent studies [10, 18] have shown that this assumption

* Simon Künzli has been supported by the Swiss Innovation Promotion Agency (KTI/CTI) through the project KTI 5500.2.

X. Yao et al. (Eds.): PPSN VIII, LNCS 3242, pp. 832–842, 2004.

is problematic; to our best knowledge, there exists no formal definition of two separate objectives, one for convergence and one for diversity, that is compliant with the Pareto dominance relation. Furthermore, there are also practical problems related to this issue as discussed in [1].

2. In most popular MOEAs, the above assumption is implemented in terms of a Pareto-based ranking of the individuals that is refined by additional density information in objective space. The algorithms, though, differ in various aspects, and therefore each of them realizes a slightly different optimization goal, which is usually not explicitly defined. That means current approaches have not been designed for flexibility with respect to the preference information used; instead, they directly implement one particular type of preference information.

As to the first aspect, the alternative is to use Pareto-compliant formalizations of the decision maker's preferences (cf. [9, 10, 18]). This, in turn, leads to a question that is directly related to the second aspect: How to design MOEAs with respect to arbitrary preference information?

The issue of integrating preference information into multiobjective search has been addressed by different researchers, see [2] for an overview. For instance, Fonseca and Fleming [8] proposed an extended dominance relation that integrates predefined priorities and goals; however, the two observations stated above also apply to the algorithm introduced by them, similarly to many other algorithms used in this context: a diversity preservation mechanism is implemented that implicitly encodes unspecified preference information. In contrast, Knowles [11] presented a multiobjective hill climber that can be combined with arbitrary unary performance measures and does not require niching methods. This approach, though, is – depending on the performance measure used – computationally expensive, and it is not clear how to extend it to population-based multiobjective optimizers that implement both mating and environmental selection.

In this paper, we extend the idea of flexible integration of preference information by Fonseca and Fleming [8] and Knowles [11] and propose a general indicator-based evolutionary algorithm, IBEA for short. The main idea is to formalize preferences in terms of *continuous* generalizations of the dominance relation, which leads to a simple algorithmic concept. Thereby, IBEA not only allows adaptation to arbitrary preference information and optimization scenarios, but also does not need any diversity preservation techniques, in contrast to [8]. In comparison to [11], IBEA is more general, since the population size can be arbitrary, and faster, because it only compares pairs of individuals and not entire approximation sets. As will be shown, the proposed approach can significantly improve the quality of the generated Pareto set approximation with respect to the considered optimization goal – in comparison to prominent Pareto-based MOEAs.

2 Preliminaries

In the following, we consider a general optimization problem that is defined by a decision space X, an objective space Z, and n objective functions $f_1, f_2, \ldots, f_n$

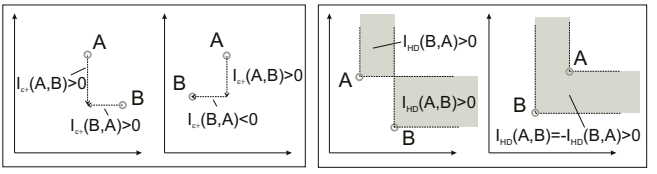

Fig. 1. Illustration of the two binary quality indicators used in this paper where A and B contain one decision vector each (left: $I_{\epsilon+}$-indicator; right: I_{HD}-indicator).

that assign to each decision vector $\boldsymbol{x} \in X$ a corresponding objective vector $\boldsymbol{z} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})) \in Z$. Without loss of generality, it is assumed that all objective functions are to be minimized and that $Z \subseteq \mathbb{R}^n$. Furthermore, the outcome of an MOEA is defined as a set of incomparable decision vectors, i.e., no decision vector dominates[1] any other decision vector in the set. Such a set will also be denoted as Pareto set approximation, and the entirety of all Pareto set approximations is represented by the symbol Ω, where $\Omega \subseteq 2^Z$. The set of all Pareto-optimal solutions is called the Pareto set S with $S \in \Omega$.

We assume that the preferences of the decision maker are given in terms of a binary quality indicator $I : \Omega \times \Omega \rightarrow \mathbb{R}$. A quality indicator in general is a function that maps k Pareto set approximations to a real number; most common are unary quality indicators where $k = 1$ (cf. [18]). Binary quality indicators can be used to compare the quality of two Pareto set approximations relatively to each other. For instance, the binary additive ϵ-indicator $I_{\epsilon+}$ [18] gives the minimum distance by which a Pareto set approximation needs to or can be translated in each dimension in objective space such that another approximation is weakly dominated[2]. Formally, it is defined as follows (cf. Fig. 1 for an illustration):

$$I_{\epsilon+}(A, B) = \min_\epsilon \left\{ \forall \boldsymbol{x}^2 \in B \; \exists \boldsymbol{x}^1 \in A \; : \; f_i(\boldsymbol{x}^1) - \epsilon \leq f_i(\boldsymbol{x}^2) \text{ for } i \in \{1, \ldots, n\} \right\}$$

The reason why we consider binary quality indicators here is that they represent a natural extension of the Pareto dominance relation, and therefore can directly be used for fitness calculation similarly to the common Pareto-based fitness assignment schemes. One requirement, though, is that the considered indicator I is compliant with Pareto dominance as defined as follows.

Definition 1. *A binary quality indicator I is denoted as* dominance preserving *if (i) $\boldsymbol{x}^1 \succ \boldsymbol{x}^2 \Rightarrow I(\{\boldsymbol{x}^1\}, \{\boldsymbol{x}^2\}) < I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})$ and (ii) $\boldsymbol{x}^1 \succ \boldsymbol{x}^2 \Rightarrow I(\{\boldsymbol{x}^3\}, \{\boldsymbol{x}^1\}) \geq I(\{\boldsymbol{x}^3\}, \{\boldsymbol{x}^2\})$ for all $\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3 \in X$.*

We will see later how these properties ensure that the proposed fitness assignment scheme is also Pareto dominance compliant. Note that the $I_{\epsilon+}$-indicator

[1] A decision vector $\boldsymbol{x}^1$ is said to dominate another decision vector $\boldsymbol{x}^2$, written as $\boldsymbol{x}^1 \succ \boldsymbol{x}^2$, if $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)$ for all $i \in \{1, \ldots, n\}$ and $f_j(\boldsymbol{x}^1) < f_j(\boldsymbol{x}^2)$ for at least one $j \in \{1, \ldots, n\}$.

[2] A ecision vector $\boldsymbol{x}^1$ weakly dominates another one $\boldsymbol{x}^2$, written as $\boldsymbol{x}^1 \succeq \boldsymbol{x}^2$, if $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$ or the corresponding objective vectors are equal.

is dominance preserving; for instance, the indicator values become negative as soon as $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$ (cf. [18]).

Now, given an arbitrary optimization problem and a corresponding binary quality indicator I, we can define the goal of the optimization process as minimizing $I(A, S)$ for $A \in \Omega$ where S is the Pareto set. If I is dominance preserving, then $I(A, S)$ is minimum for $A = S$; in the case of the additive ϵ-indicator, $I_{\epsilon+}(S, S) = 0$. Note that we do not require here that S is known, it just serves the formalization of the optimization goal.

3 Indicator-Based Selection

Taking the scenario described in Section 2, the question is how I can be integrated in an MOEA to minimize $I(A, S)$, where A is the generated Pareto set approximation. This section deals with this issue.

3.1 Fitness Assignment

The population P represents a sample of the decision space, and fitness assignment tries to rank the population members according to their usefulness regarding the optimization goal. Among the different ways how the exploit the information given by P and I, one possibility is to simply sum up the indicator values for each population member with respect to the rest of population, i.e.: $F'(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})$ This fitness value F', which is to be maximized, is a measure for the "loss in quality" if $\boldsymbol{x}^1$ is removed from the population. For $I_{\epsilon+}$, e.g., $F'(\boldsymbol{x}^1)$ divided by the population size N equals the average ϵ needed to cover $\boldsymbol{x}^1$ by other population members. However, we will use a slightly different scheme in the following that amplifies the influence of dominating population members over dominated ones:

$$F(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} -e^{-I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})/\kappa}$$

We use one property of dominance preserving indicators here, namely that $I(\{\boldsymbol{x}^1\}, \{\boldsymbol{x}^2\}) < I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})$ if $\boldsymbol{x}^1 \succ \boldsymbol{x}^2$. Thereby, the influence of small indicator values contributes much more to the overall fitness than large values. The parameter κ is a scaling factor depending on I and the underlying problem; κ needs to be greater than 0. The following theorem shows that this fitness scheme is compliant with the Pareto dominance relation.

Theorem 1. *Let I be a binary quality indicator. If I is dominance preserving, then it holds that $\boldsymbol{x}^1 \succ \boldsymbol{x}^2 \Rightarrow F(\boldsymbol{x}^1) > F(\boldsymbol{x}^2)$.*

Proof. From Def. 1 and property (i) it follows that the indicator value $I(\{\boldsymbol{x}^1\}, \{\boldsymbol{x}^2\}) < I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})$. Due to property (ii) of Def. 1, we know that $I(\{\boldsymbol{x}^3\}, \{\boldsymbol{x}^1\}) \geq I(\{\boldsymbol{x}^3\}, \{\boldsymbol{x}^2\}), \forall \boldsymbol{x}^3 \notin \{\boldsymbol{x}^1, \boldsymbol{x}^2\}$. Since $-e^{-x/\kappa} > -e^{-y/\kappa}$, if $x < y$ and $\kappa > 0$, it follows that $F(\boldsymbol{x}^1) > F(\boldsymbol{x}^2)$. □

3.2 Example Indicators

We have now seen how the additive ϵ-indicator can be used to assign fitness values to the population members. However, many other dominance preserving indicators can be defined that could be used instead. For instance, the following I_{HD}-indicator is based on the hypervolume concept [17]:

$$I_{HD}(A,B) = \begin{cases} I_H(B) - I_H(A) & \text{if } \forall \boldsymbol{x}^2 \in B\ \exists \boldsymbol{x}^1 \in A\ :\ \boldsymbol{x}^1 \succ \boldsymbol{x}^2 \\ I_H(A+B) - I_H(A) & \text{else} \end{cases}$$

Here, $I_H(A)$ gives the hypervolume of the objective space dominated by A, and accordingly $I_{HD}(A,B)$ measures the volume of the space that is dominated by B but not by A with respect to a predefined reference point $\boldsymbol{Z}$. While the calculation of the $I_{HD}(A,B)$-values is computationally expensive for approximations containing several decision vectors, it is of order $\mathcal{O}(n)$ if two decision vectors are compared. The I_{HD}-indicator will be used in addition to the $I_{\epsilon+}$-indicator later in this paper. A graphical interpretation for I_{HD} can be found on the right hand side of Fig. 1.

Other examples for binary quality indicators that could be used here are described in Hansen and Jaszkiewicz's study [9].

3.3 Basic Algorithm

Based on the above fitness assignment scheme, we propose a general indicator-based evolutionary algorithm (IBEA) that performs binary tournaments for mating selection and implements environmental selection by iteratively removing the worst individual from the population and updating the fitness values of the remaining individuals. Its running-time complexity is $\mathcal{O}(\alpha^2)$ with regard to the population size α. Details of the algorithm are given below; note that it represents only the basic version of IBEA (denoted B-IBEA in the following), an extended version will be specified later.

Algorithm 1 (Basic IBEA)

Input: α *(population size)*
N *(maximum number of generations)*
κ *(fitness scaling factor)*
Output: A *(Pareto set approximation)*

Step 1: ***Initialization**: Generate an initial population P of size α; set the generation counter m to 0.*

Step 2: ***Fitness assignment**: Calculate fitness values of individuals in P, i.e., for all $\boldsymbol{x}^1 \in P$ set $F(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} -e^{-I(\{\boldsymbol{x}^2\},\{\boldsymbol{x}^1\})/\kappa}$.*

Step 3: ***Environmental selection**: Iterate the following three steps until the size of population P does not exceed α:*
1. *Choose an individual $\boldsymbol{x}^* \in P$ with the smallest fitness value, i.e., $F(\boldsymbol{x}^*) \leq F(\boldsymbol{x})$ for all $\boldsymbol{x} \in P$.*
2. *Remove $\boldsymbol{x}^*$ from the population.*
3. *Update the fitness values of the remaining individuals, i.e., $F(\boldsymbol{x}) = F(\boldsymbol{x}) + e^{-I(\{\boldsymbol{x}^*\},\{\boldsymbol{x}\})/\kappa}$ for all $\boldsymbol{x} \in P$.*

Step 4: ***Termination**: If $m \geq N$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in P. Stop.*

Step 5: ***Mating selection**: Perform binary tournament selection with replacement on P in order to fill the temporary mating pool P'.*

Step 6: ***Variation**: Apply recombination and mutation operators to the mating pool P' and add the resulting offspring to P. Increment the generation counter ($m = m + 1$) and go to Step 2.*

3.4 Simulation Results

The proposed algorithm was tested on several well-known benchmark problems: the 2-dimensional knapsack problem instance from [17] with 100 items, a network processor application comprising problem instances with two (EXPO2), three (EXPO3), and four (EXPO4) objectives (cf. [14]), and four continuous test functions, namely ZDT6 [15] and KUR [12] with two objectives as well as DTLZ2 and DTLZ6 [6] with three objectives each[3]. For all problems, the population size α was set to 100 and the maximum number of generations N to 200. Overall, 30 runs with different initial populations were carried out per algorithm and per benchmark problem.

To assess the performance values, we have compared the solutions found by the two new algorithms B-IBEA$_{\epsilon+}$ and B-IBEA$_{HD}$ with NSGA-II [5] and SPEA2 [16]. The performance comparison was carried out using the quality indicators $I_{\epsilon+}$ and I_{HD}, i.e., we have computed 30 indicator values $I(A, R)$ for different seeds for all the tested algorithms. In this formula, A stands for the output that the evolutionary algorithm produced; the reference set R was determined by merging all solutions found by all the different algorithms into one set and keeping the non-dominated solutions. R was used instead of the Pareto set S, because S is usually unknown.

For the results obtained using B-IBEA$_{\epsilon+}$, B-IBEA$_{HD}$, NSGA-II and SPEA2, we can observe in the comparison that B-IBEA$_{\epsilon+}$ and B-IBEA$_{HD}$ perform significantly better than the other algorithms regarding both performance indicators and for appropriately chosen parameter κ. Although for the variation parameter settings described above, the choice for the parameter κ does not influence the performance of the algorithm much, we found other parameter settings that indicate that the optimal choice of κ can vary and is dependent on the problem and the indicator used. This is, e.g., the case for ZDT6 if both mutation and recombination probability are set to 1. In Figure 2a (top), the influence of different values κ for the performance of B-IBEA$_{\epsilon+}$ on the problem ZDT6 is given. The performance of B-IBEA$_{HD}$ not only depends on the choice of κ but also on the choice of the reference point. In Figure 2a (bottom), we can see that for a particular choice of both κ and the reference point, the performance of B-IBEA$_{HD}$ for problem ZDT6 is better than SPEA2 and NSGA-II, but for other choices for κ and the reference point the performance is substantially worse. We do not give all the results for the basic versions of IBEA in a table due to space limitations.

[3] For the continuous problems, the individuals are coded as real vectors, where the SBX-20 operator is used for recombination and a polynomial distribution for mutation [4]. The recombination and mutation probabilities were set to 1.0 and to 0.01, resp., according to [7]. For the knapsack problem, an individual is represented by a bit string, recombination is performed as one-point crossover with probability 0.8 according to [17], and point mutations are performed with bit-flip probability 0.04, as suggested in [13]. For the design-space exploration problems EXPO, the representation of individuals and the operators are described in [14]. The recombination probability was set to 0.5 and the probability for mutation was set to 0.8. (These are the same parameter settings as proposed in [14]).

The question that arises inspecting the results obtained so far is how we can improve the algorithms such that (i) the same κ value can be used for different problems and indicators, and (ii) B-IBEA$_{HD}$ becomes less sensitive to the choice of the reference point for I_{HD}.

4 Improving Robustness

4.1 Adaptive IBEA

The values for the indicators $I(A, B)$ can be widely spread for different problems. This makes it difficult to determine an appropriate value for κ. We can ease this task by adaptively scaling the indicator values such that they lie in the interval $[-1, 1]$ for all points in the population. Thereby, we can use the same value κ for all the problems.

To tackle the problem of determining a good reference point for the I_{HD} indicator, we propose to use adaptive scaling not only for the indicator values, but also for the objective values. After scaling, the objective values lie in the interval $[0, 1]$. Like this, we can choose the worst values for each objective found in the population as reference point to calculate I_{HD}, i.e. the reference point would be set to 1 for all objectives. If we use this strategy, the only problem remaining is that the corner points found in a population do not add to the hypervolume. To overcome this problem, for the reference point we used a value of 2 for all objectives in the experiments with IBEA$_{HD}$.

Algorithm 2 (Adaptive IBEA)

...

Step 2: ***Fitness assignment:*** *First scale objective and indicator values, and then use scaled values to assign fitness values:*

1. *Determine for each objective f_i its lower bound $\underline{b_i} = min_{\boldsymbol{x}\in P} f_i(\boldsymbol{x})$ and its upper bound $\overline{b_i} = max_{\boldsymbol{x}\in P} f_i(\boldsymbol{x})$.*
2. *Scale each objective to the interval $[0, 1]$, i.e., $f_i'(\boldsymbol{x}) = (f_i(\boldsymbol{x}) - \underline{b_i})/(\overline{b_i} - \underline{b_i})$).*
3. *Calculate indicator values $I(\boldsymbol{x}^1, \boldsymbol{x}^2)$ using the scaled objective values f_i' instead of the original f_i, and determine the maximum absolute indicator value $c = max_{\boldsymbol{x}^1,\boldsymbol{x}^2\in P}|I(\boldsymbol{x}^1, \boldsymbol{x}^2)|$.*
4. *For all $\boldsymbol{x}^1 \in P$ set $F(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2\in P\setminus\{\boldsymbol{x}^1\}} -e^{-I(\{\boldsymbol{x}^2\},\{\boldsymbol{x}^1\})/(c\cdot\kappa)}$.*

Step 3: ***Environmental selection:*** *...*

1. *...*
2. *...*
3. *Update the fitness values of the remaining individuals, i.e., $F(\boldsymbol{x}) = F(\boldsymbol{x}) + e^{-I(\{\boldsymbol{x}^*\},\{\boldsymbol{x}\})/(c\cdot\kappa)}$ for all $\boldsymbol{x} \in P$.*

...

The algorithms IBEA$_{\epsilon+}$ and IBEA$_{HD}$ denote the adaptive versions of the basic algorithms. For these versions, the choice of κ only marginally depends on the problem and the indicator under consideration. The changes in the initial algorithm are shown in Algorithm 2. For the experiments discussed in Section 4.2, we have used $\kappa = 0.05$ for all the problems and algorithms. Preliminary tests have shown that this value for κ produced good results on the problems considered. Furthermore, the value for κ was chosen such that in the implementation no numerical problems occur, because smaller values led to fitness values larger than the maximum allowed double value in the PISA-specification ($= 10^{99}$).

Table 1. Comparison of different MOEAs for the $I_{\epsilon+}$-indicator using the Wilcoxon rank test. The "P value" columns give the adjusted P value of the corresponding pairwise test that accounts for multiple testing; it equals the lowest significance level for which the null-hypothesis (the medians are drawn from the same distribution) would still be rejected. The "T" columns give the outcome of the test for a significance level of 5%: either the algorithm corresponding to the specific row is significantly better (↑) resp. worse (↓) than the algorithm associated with the corresponding column or there is no significant difference between the results (⇌).

		SPEA2		NSGA-II		$SPEA2_{adap}$		$IBEA_{\epsilon,adap}$	
		P value	T	P value	T	P value	T	P value	T
ZDT6	NSGA-II	$5.6073 \cdot 10^{-4}$	↑						
	$SPEA2_{adap}$	$> 5\%$	⇌	$8.1975 \cdot 10^{-6}$	↓				
	$IBEA_{\epsilon,adap}$	$8.1014 \cdot 10^{-9}$	↑	$2.0023 \cdot 10^{-5}$	↑	$1.9568 \cdot 10^{-9}$	↑		
	$IBEA_{HD,adap}$	0.0095	↑	$> 5\%$	⇌	$5.4620 \cdot 10^{-5}$	↑	$1.3853 \cdot 10^{-5}$	↓
DTLZ2	NSGA-II	$3.0199 \cdot 10^{-10}$	↓						
	$SPEA2_{adap}$	$> 5\%$	⇌	$3.0199 \cdot 10^{-10}$	↑				
	$IBEA_{\epsilon,adap}$	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑		
	$IBEA_{HD,adap}$	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$5.5329 \cdot 10^{-7}$	↓
DTLZ6	NSGA-II	$8.1014 \cdot 10^{-9}$	↓						
	$SPEA2_{adap}$	$> 5\%$	⇌	$6.1210 \cdot 10^{-9}$	↑				
	$IBEA_{\epsilon,adap}$	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑		
	$IBEA_{HD,adap}$	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.5923 \cdot 10^{-4}$	↓
KUR	NSGA-II	$> 5\%$	⇌						
	$SPEA2_{adap}$	$> 5\%$	⇌	$> 5\%$	⇌				
	$IBEA_{\epsilon,adap}$	$3.0199 \cdot 10^{-10}$	↓	$3.0199 \cdot 10^{-10}$	↓	$6.6955 \cdot 10^{-10}$	↓		
	$IBEA_{HD,adap}$	$3.0199 \cdot 10^{-10}$	↓	$3.0199 \cdot 10^{-10}$	↓	$4.9752 \cdot 10^{-10}$	↓	$> 5\%$	⇌
Knap.	NSGA-II	$> 5\%$	⇌						
	$SPEA2_{adap}$	$> 5\%$	⇌	$> 5\%$	⇌				
	$IBEA_{\epsilon,adap}$	$> 5\%$	⇌	$> 5\%$	⇌	$> 5\%$	⇌		
	$IBEA_{HD,adap}$	$> 5\%$	⇌	$> 5\%$	⇌	$> 5\%$	⇌	$> 5\%$	⇌
EXPO2	NSGA-II	$> 5\%$	⇌						
	$SPEA2_{adap}$	$> 5\%$	⇌	0.0189	↑				
	$IBEA_{\epsilon,adap}$	$1.0837 \cdot 10^{-8}$	↑	$2.6753 \cdot 10^{-9}$	↑	$6.4048 \cdot 10^{-8}$	↑		
	$IBEA_{HD,adap}$	$1.9638 \cdot 10^{-7}$	↑	$1.2260 \cdot 10^{-8}$	↑	$6.6261 \cdot 10^{-7}$	↑	$> 5\%$	⇌
EXPO3	NSGA-II	$> 5\%$	⇌						
	$SPEA2_{adap}$	$> 5\%$	⇌	$> 5\%$	⇌				
	$IBEA_{\epsilon,adap}$	$4.3165 \cdot 10^{-8}$	↑	$5.0801 \cdot 10^{-8}$	↑	$3.1159 \cdot 10^{-7}$	↑		
	$IBEA_{HD,adap}$	$2.4189 \cdot 10^{-7}$	↑	$1.5732 \cdot 10^{-7}$	↑	$1.1653 \cdot 10^{-6}$	↑	$> 5\%$	⇌
EXPO4	NSGA-II	$> 5\%$	⇌	-					
	$SPEA2_{adap}$	$> 5\%$	⇌	$9.4209 \cdot 10^{-4}$	↓				
	$IBEA_{\epsilon,adap}$	$1.8546 \cdot 10^{-10}$	↑	$6.9754 \cdot 10^{-10}$	↑	$1.8390 \cdot 10^{-10}$	↑		
	$IBEA_{HD,adap}$	$1.9883 \cdot 10^{-10}$	↑	$1.0221 \cdot 10^{-9}$	↑	$1.9716 \cdot 10^{-10}$	↑	$> 5\%$	⇌

4.2 Simulation Results

In Fig. 2 (b), the comparison results for the problems DTLZ6 and EXPO2 are shown. For both problems, the proposed algorithms $IBEA_{\epsilon+}$ and $IBEA_{HD}$ perform significantly better than SPEA2 and NSGA-II with respect to the performance indicators $I_{\epsilon+}$ and I_{HD}. Note that these IBEA versions all work with the same value for κ.

In addition to SPEA2, NSGA-II and the proposed $IBEA_{\epsilon+}$ and $IBEA_{HD}$, we have implemented an adaptive version of SPEA2 to see the impact of adaptive objective-value scaling as such. The performance of the adaptive version of SPEA2 is comparable to the original algorithm on the test problems, and the Wilcoxon rank test returns false for all the problems investigated, i. e. the

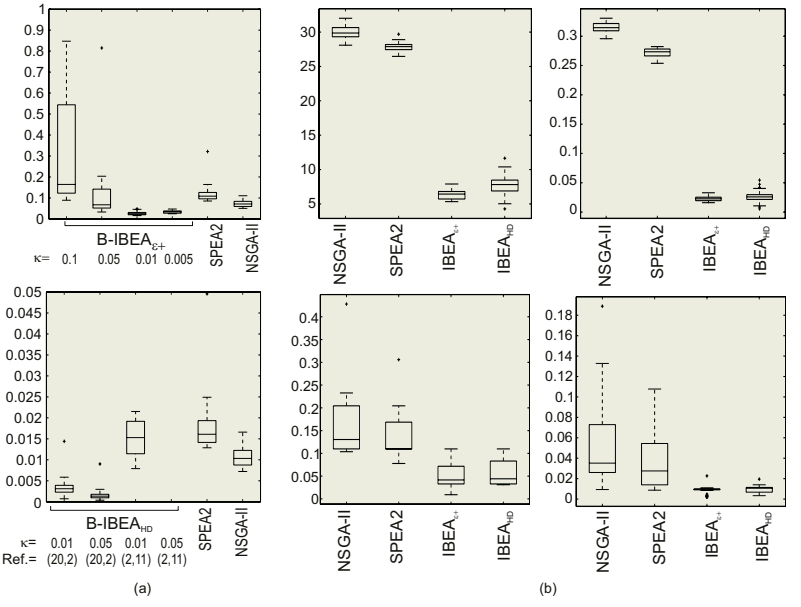

Fig. 2. (a) (top) The indicator values $I_{\epsilon+}$ for SPEA2, NSGA-II and B-IBEA$_{\epsilon+}$ for different values of κ. For all the different algorithms one outlier was removed from the result sample for improved readability. (bottom) The indicator values I_{HD} for SPEA2, NSGA-II and B-IBEA$_{HD}$ for different values of κ and the reference point. In the 4^{th} column, no values are given because they are about 10 times greater than the values given. **(b)** Performance comparison for adaptive IBEA$_{\epsilon}$, IBEA$_{HD}$, SPEA2 and NSGA-II solving problems DTLZ6 (top) and EXPO2 (bottom). On the left, values for $I_{\epsilon+}$, on the right for I_{HD} are given.

distributions of $I(A, R)$ for SPEA2 and the adaptive version of SPEA2 are not significantly different.

An overview of the results for $I_{\epsilon+}$ is given in Table 1. We do not give a table with results for I_{HD} due to space limitations. Overall, we can see that for the continuous problems DTLZ2, DTLZ6 and ZDT6, the proposed algorithms IBEA$_{\epsilon+}$ and IBEA$_{HD}$ perform significantly better than SPEA2 or NSGA-II; only for KUR, the latter provide better performance than the two IBEA instances. For the discrete knapsack problem, the significance tests return false, i.e. the indicator value distributions generated by the different search algorithms are statistically not different from each other. In contrast, the indicator-based algorithms show significantly better performance for the design-space exploration problem EXPO in two, three and four dimensions.

5 Conclusions

Every MOEA implementation inevitably makes assumptions about the decision maker's preferences which are usually hard coded in the algorithm. These preferences, though, may vary for each user and application. Therefore, we have

- argued that ideally MOEAs would be designed and evaluated with regard to the specific preferences of the user, formalized in terms of a performance measure, and
- proposed a general indicator-based evolutionary algorithm (IBEA) that, contrarily to existing population-based MOEAs, allows to adapt the search according to arbitrary performance measures. For two different performance measures, this approach has be shown to generate significantly better results on six of eight benchmark problems in comparison to SPEA2 and NSGA-II, while no statistically significant performance difference could be observed on one of the test function.

IBEA as well as the other MOEAs and the benchmark problems considered in the paper can be downloaded as precompiled, ready-to-use components from the PISA Website `http://www.tik.ee.ethz.ch/pisa/`.

References

1. P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.
2. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, New York, 2002.
3. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.
4. K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.
5. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, *PPSN VI*, LNCS Vol. 1917, pages 849–858. Springer, 2000.
6. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *CEC 2002*, pages 825–830. IEEE Press, 2002.
7. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham et al., editors, *Evolutionary Computation Based Multi-Criteria Optimization: Theoretical Advances and Applications*. Springer, 2004. To appear.
8. C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part ii: Application example. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1):38–47, 1998.
9. M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations of the non-dominated set. Technical report, Institute of Mathematical Modeling, Technical University of Denmark, 1998. IMM Technical Report IMM-REP-1998-7.
10. J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *CEC 2002*, pages 711–716, Piscataway, NJ, 2002. IEEE Press.
11. J. D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, University of Reading, 2002.
12. F. Kursawe. A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Männer, eds., *PPSN I* , pages 193–197. Springer, 1991.

13. M. Laumanns, E. Zitzler, and L. Thiele. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 181–196. Springer, 2001.
14. L. Thiele, S. Chakraborty, M. Gries, and S. Künzli. Design space exploration of network processor architectures. In M. Franklin et. al., eds, *Network Processor Design Issues and Practices, Vol. 1*, chapter 4, Morgan Kaufmann, Oct. 2002.
15. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
16. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. Giannakoglou et al., editors, *EUROGEN 2001*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.
17. E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
18. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.