A Modified Indicator-based Evolutionary Algorithm (*m*IBEA)

Wenwen Li, Ender Özcan, Robert John
School of Computer Science
University of Nottingham
Jubilee Campus, Wollaton Road,
Nottingham, UK, NG8 1BB
Email: wenwen.li@nottingham.ac.uk
ender.ozcan@nottingham.ac.uk
robert.john@nottingham.ac.uk

John H. Drake
Operational Research Group
Queen Mary University of London
Mile End Road,
London, UK, E1 4NS
Email: j.drake@qmul.ac.uk

Aneta Neumann, Markus Wagner
School of Computer Science
The University of Adelaide
Ingkarni Wardli Building,
Adelaide, Australia SA 5005
Email: aneta.neumann@adelaide.edu.au
markus.wagner@adelaide.edu.au

Abstract—**Multi-objective evolutionary algorithms (MOEAs) based on the concept of Pareto-dominance have been successfully applied to many real-world optimisation problems. Recently, research interest has shifted towards *indicator*-based methods to guide the search process towards a good set of trade-off solutions. One commonly used approach of this nature is the indicator-based evolutionary algorithm (IBEA). In this study, we highlight the solution distribution issues within IBEA and propose a modification of the original approach by embedding an additional Pareto-dominance based component for selection. The improved performance of the proposed modified IBEA (*m*IBEA) is empirically demonstrated on the well-known DTLZ set of benchmark functions. Our results show that *m*IBEA achieves comparable or better hypervolume indicator values and epsilon approximation values in the vast majority of our cases (13 out of 14 under the same default settings) on DTLZ1-7. The modification also results in an over 8-fold speed-up for larger populations.**

I. INTRODUCTION

In multi-objective optimisation, where multiple objectives are optimised simultaneously, the goal is to find a set of $Pareto$-optimal solutions known as the Pareto front (PF). The PF consists of a set of solutions that are not dominated by each other, which are termed as *non-dominated* solutions, representing the trade-off that exists between different objectives. This dominance relation, also known as *Pareto dominance* relation, ($\prec$) is defined between solutions x^1 and x^2. We say w.l.o.g., in a minimisation problem that x^1 dominates x^2 ($x^1 \prec x^2$) if and only if $f_i(x^1) \leq f_i(x^2)$ for all k objective functions ($i \in \{1, \ldots k\}$), and $f_i(x^1) < f_i(x^2)$ for at least one objective function. One of the difficulties in multi-objective search is to find a set of solutions to minimise the distance to the true Pareto front (PF) while maintaining the diversity of the solution set in the objective space.

Multi-objective evolutionary algorithms (MOEAs) are widely used to solve various multi-objective optimisation problems [16] and are considered to be general and robust search mechanisms [1]. Evolutionary algorithms (EAs) are a class of stochastic optimisation methods that mimic the process of evolution in nature. Examples of EAs, such as genetic algorithms, evolutionary programming, and evolution strategies [1] operate on a set of solutions using the basic principles of natural evolution: selection, reproduction by means of recombination and mutation. The algorithmic difference between single-objective EAs and MOEAs is that additionally in an MOEA, the multiple objectives of a solution must be transformed into a single fitness value to facilitate the comparison of individual solutions [6]. Thus, MOEAs often vary in the method of this transformation, considering the balance of convergence and diversification during the search. Some MOEAs, such as NSGA-II [7], SPEA2 [18], and AGE [3, 13] incorporate Pareto-based ranking of the individuals and an additional density measurement (*crowding distance* in NSGA-II, *k-th nearest neighbour* in SPEA2, ϵ-*dominance* in AGE) in the objective space. However, between two non-dominated solutions, purely Pareto-based MOEAs are not able to ascertain which solutions have better potential for convergence. IBEA [17] was one of the earliest indicator-based MOEAs proposed in the literature. Originally, it came in two variants: one using ϵ-dominance for guidance, denoted IBEA$_\epsilon$, and another based on hypervolume, denoted IBEA$_{HD}$ ('HD' stands for hypervolume difference) which will be referred to as IBEA from this point onward in this study. IBEA associates a fitness value with each solution based on the selected indicator (hypervolume or ϵ), attempting to guide the search towards the true PF. IBEA was shown to achieve significantly better performance on various benchmark functions than NSGA-II and SPEA2 [17], however the distribution of the solutions found by IBEA has rarely been reported or discussed in detail. In this paper, we propose a modified variant of IBEA$_{HD}$, termed *m*IBEA, which adds a Pareto-based element to this indicator-based method, analysing the distribution of non-dominated solutions found. For further information about MOEAs and indicator-based MOEAs in particular, we refer the interested reader to [5, 6, 16].

The remainder of the paper is structured as follows. We first describe the original IBEA in Section II-A, then present observations of the solution distributions observed using existing MOEAs in Section II-B. The proposed *m*IBEA is introduced in Section III. Experimental results comparing IBEA and *m*IBEA are presented in Section IV. We draw conclusions and provide

ideas for future work in Section V.

II. Indicator-based Evolutionary Algorithm

Since the focus in the paper is on the hypervolume variant of IBEA, i.e. IBEA_{HD}, we first give a detailed description of the original IBEA. We then provide visualisations and observations for the non-dominated solution sets found using IBEA and two other existing MOEAs, one Pareto dominance-based (NSGA-II [7]) and one indicator-based (SMS-EMOA [2]).

A. Description of IBEA

The core idea of IBEA_{HD} is to employ a binary hypervolume indicator in the selection process, when determining which solutions survive to the next generation. The binary hypervolume indicator assigns a real-valued number to two solution sets with respect to a reference point. The formula of $I_{HD}(A, B)$ is defined as space that is dominated by population B, but not by A, shown in Equation (1) [17].

$$I_{HD}(A,B) = \begin{cases} I_H(B) - I_H(A), \forall x^2 \in B \exists x^1 \in A : x^1 \prec x^2 \\ I_H(A+B) - I_H(A), o.w. \end{cases}$$
(1)

where $I_H(A)$ denotes the hypervolume formed by the solution set A. Correspondingly, $I_H(A+B)$ means the hypervolume of the union of solution set A and B. $I_{HD}(A, B)$ is negative if all solutions in B are dominated by solutions in A. Note that $I_H(A) \neq I_H(B)$.

The pseudocode of the original IBEA is given in Alg. 1. IBEA first randomly generates an initial population in Step1, then the following steps loop until the stopping criterion is satisfied. The objective values are scaled and the fitness is assigned to each individual in Step2 and Step3. Step4 performs environmental selection, iteratively removing the worst individual in the population P based on indicator value until μ individuals remain (this step will do nothing in the first iteration of the algorithm as $P = \mu$). Upon removal of each solution, the indicator values of the remaining solutions must be updated. This step continues until the number of solutions in P does not exceed μ. The standard mating selection (Step6) and variation (Step7) steps are performed to generate new individuals and add them to the population P.

B. Solution distribution issues in existing MOEAs

It is common in the MOEA literature to use various quantitative indicators, such as hypervolume, ϵ-approximation and inverted generational distance (IGD) [19], to compare the performance of different MOEAs. However, the distribution of non-dominated solutions over the resultant solution fronts is relatively rarely reported, at least in part due to the lack of quantitative indicators to measure it in the objective space.

One method for measuring the spread of solutions over the objective space we will use here is Generalised Spread (GS) [15]. GS computes the average Euclidean distance of any two consecutive solutions [7] (or nearest neighbours in [15]) in the population, and considers this average in the context of the minimal achieved distances to the extreme points of the true

---

Algorithm 1: (Adaptive) IBEA_{HD} as described in [4, 17]

Inputs:
μ: population size;
N: number of total solution evaluations;
ρ: objective values scaling factor;
κ: indicator value (hypervolume difference) scaling factor
Output: A: Pareto set approximation
Step1: Initialisation: Randomly generate the initial
  population P and set up evaluation counter m.
Step2: Scale objective values:
1) find the lower ($\underline{b_i} = min_{x \in P} f_i(x)$) and upper
  ($\overline{b_i} = max_{x \in P} f_i(x)$) bound of each objective i.
2) increase the upper bound $\overline{b_i}' = \underline{b_i} + \rho * (\overline{b_i} - \underline{b_i})$
3) scale each objective to the interval [0, 1];
  $f_i' = (f_i(x) - \underline{b_i})/(\overline{b_i}' - \underline{b_i})$
Step3: Fitness Assignment:
1) calculate indicator values $I(x^1, x^2)$ using the scaled
  objective values f_i' instead of original objective values
  f_i, and determine the maximum absolute indicator value
  $c = max_{x^1, x^2 \in P}|I(x^1, x^2)|$
2) compute the fitness for all $x^1 \in P$,
  $F(x^1) = \sum_{x^2 \in P \setminus x^1} - \exp^{-I(x^2,x^1)/c*\kappa}$
Step4: Environmental Selection: iterate the following
  three sub-steps until the size of population $P \leq \mu$.
1) choose an individual $x^\star \in P$ with the smallest fitness
  value, i.e., $F(x^\star) \leq F(x)$ for all $x \in P$
2) remove $x^\star$ from P
3) update fitness values of all the remaining individuals
  $x \in P$ as $F(x) = F(x) + \exp^{-I(x^\star,x)/c*\kappa}$
Step5: Termination criteria: if $m \geq N$, return the
  non-dominated solutions in P as A; otherwise, continue.
Step6: Mating Selection: apply binary tournament
  selection operator to P to select two parents from P.
Step7: Variation: perform crossover operator on the two
  parents to generate two offspring, and use an mutation
  operator on one of the offspring. The resulting offspring
  is then added to P. Increment iteration counter
  ($m = m + 1$). Go to Step 2.

PF. The smaller the value of GS, the better spread the resultant front has as the variance is reduced and/or points closer to the extreme points have been found. The ideal spread is achieved when GS equals to 0 as this means that the extreme points are found and the solutions are evenly spread out.

In this study, for the purpose of analysis, we also use a new metric – *border fraction (bf)* to provide a certain quantitative measurement of the solution distribution of different PFs. We define *border fraction* as the fraction of non-dominated points lying in a pre-defined border or area of the resultant PF. More precisely, assume the lower bound of each objective function is 0, the border threshold for each objective is denoted as θ_i, then the border is the volume defined by $(0 < f_1 \leq \theta_1 || 0 < f_2 \leq \theta_2 ||, ..., || 0 < f_k \leq \theta_k)$, where k is number of objectives in the multi-objective problem. Given *ndi* non-dominated solutions in

the population, the border fraction *bf* value is the proportion of non-dominated individuals in the population which lie within in this border. Note that *bf* is not a general metric to measure the solution distribution of a Pareto optimal (or approx.) set.

As a motivating example, we apply three well-known MOEAs, NSGA-II, SMS-EMOA and IBEA, to a sample of three DTLZ problems [8], using the default settings for each method. Figure 1 shows the non-dominated solutions found in 100 independent runs for each MOEA on each problem, except for SMS-EMOA which was run only 10 times due to its high running time. The maximum number of evaluations for each run was 100,000. The *bf* and *ndi* values are also provided on the top right corner of each plot.
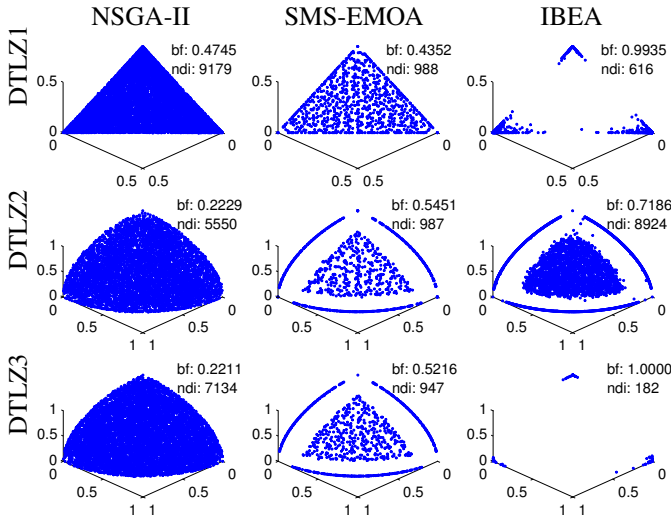

Fig. 1: Comparison of non-dominated solution sets for NSGA-II, SMS-EMOA and IBEA.

Firstly, we can see that the solutions from NSGA-II can cover the whole fronts of all the problems DTLZ1-3. Secondly, SMS-EMOA can reach almost full coverage of the PF for DTLZ1, however, on DTLZ2 and 3 the PFs have clear gaps between the border and central area. Thirdly, similar to SMS-EMOA, IBEA also shows clear gaps (empty space) on the PF of DTLZ2. Also, solutions by the original IBEA are heavily concentrated in the corner points on DTLZ1 and DTLZ3.

The GS measurement of the solution fronts obtained from NSGA-II, SMS-EMOA and IBEA (as illustrated in Figure 1) is shown in Table I. The best GS on each benchmark function is highlighted in bold. Not surprisingly, IBEA performs poorly w.r.t. GS among all the three algorithms on DTLZ1-3. However, interestingly, although SMS-MOEA shows gaps on the resultant PFs of DTLZ2 and 3, the GS indicator still shows that it performs best on both problems. This suggests that under certain situations (e.g., PFs of DTLZ2 and 3 from SMS-EMOA) GS cannot capture the distribution of the PF, i.e., the spread calculated is not equivalent to the distribution.

In the following, we use *bf* and *ndi* to capture the bias within the distribution towards extreme points on the solution fronts. The border threshold is set as 0.03 for DTLZ1 and 0.1 for both

TABLE I: Generalised Spread $Mean_{std}$ values for NSGA-II, SMS-EMOA and IBEA on DTLZ1-3.

Generalised Spread	NSGA-II	SMS-EMOA	IBEA
DTLZ1	$0.5774_{0.30}$	$\mathbf{0.0780_{0.01}}$	$1.3835_{0.38}$
DTLZ2	$0.4951_{0.04}$	$\mathbf{0.2393_{0.02}}$	$0.4705_{0.03}$
DTLZ3	$0.5284_{0.09}$	$\mathbf{0.2388_{0.02}}$	$1.5921_{0.30}$

DTLZ2 and 3. These parameters are determined by the gaps or corner points from IBEA illustrated in Figure 1. We can see that for IBEA, on DTLZ1 and DTLZ3 almost all points lie in the border area of IBEA, and 72% of points are in the border area for DTLZ2. SMS-EMOA shows good PF coverage on DTLZ1, but a biased solution distribution is obvious on the outside 'ring' area and the inside central area in DTLZ2 and DTLZ3. Interestingly, the Pareto dominance-based MOEA NSGA-II shows good PF coverage and fairly uniform solution distributions on DTLZ1-3. In addition, IBEA only identifies 616 non-dominated solutions when solving DTLZ1 and 182 for DTLZ3 after 100 independent runs, which is significantly less than the *ndi* values of NSGA-II on these instances (9179 for DTLZ1 and 7134 for DTLZ3).

To the best of our knowledge, even though these uneven solution distributions have been plotted in articles before, e.g. [2, 12], no further details are discussed in those papers. Although IBEA is often reported as having exceptional indicator value performance [9, 17], the actual distribution of solutions that it obtains is surprisingly uneven. Additionally, mediocre performance [12, 14] is reported on certain benchmark functions, possibly as a result of this uneven distribution. In the following section, we introduce a modified version of IBEA that attempts to overcome this issue.

III. PROPOSED MODIFIED IBEA (*m*IBEA)

The behaviour of IBEA observed in the previous section suggests that the search is trapped in certain regions of the search space when solving DTLZ1 and DTLZ3, and suffers from a lack of diversity in solutions. In other words, it seems that after a certain time point, IBEA is not able to produce solutions with high hypervolume contributions. A poor spread of non-dominated solutions when using IBEA was observed previously by Li et al. [11] in the context of a real world application, the multi-objective wind farm layout optimisation problem. The proposed method in that paper illustrated that the hybridisation of Pareto dominance-based MOEAs and IBEA could improve the overall performance, in terms of both convergence and diversification within the search. It is this observation that the modified IBEA, *m*IBEA, we introduce here is based on.

*m*IBEA excludes dominated solutions at each generation after the new population is created. To achieve this, we use the dominance-based sorting method of NSGA-II, however, other approaches for achieving this can be used as well.

The difference between *m*IBEA and IBEA is to add this one step before the scaling Step2 in IBEA (as described in Section II-A). The effect is that the scaling of the objective scores is no longer affected by dominated solutions that

are far away from the best non-dominated solutions in the population, see Step2.1 of Algorithm 1 where minimum and maximum values across the entire population are determined. The pseudocode of *m*IBEA is given in Algorithm 2.

Algorithm 2: *m*IBEA

Input: μ: population size;
N: number of total solution evaluations;
ρ: objective values scaling factor;
κ: indicator value (hypervolume difference) scaling factor
Output: A: Pareto set approximation
Step1: Initialisation (see Step 1 in Alg. 1);
Step2.1: Use the fast non-dominated sorting of NSGA-II to get non-dominated solutions in P and use the non-dominated solutions as the new P.
  1) rank the solutions in P: Ranking rankedP = new Ranking(P);
  2) get the non-dominated solutions: P = rankedP.getSubfront(0);
Step2.2: Scale objective values (see Step 2 in Alg. 1);
Steps 3-7 are the same as the original IBEA in Alg. 1.

IV. COMPUTATIONAL RESULTS

A. Experimental Design

The following experiments were performed on the full DTLZ problem set (DTLZ1-7) [8]. Our analysis focuses on the three-objective DTLZ1 and 3, as qualitatively good coverage of the true Pareto front can be achieved relatively easily by different algorithms. Besides, these two problems exhibit different properties: DTLZ1 has a planar front in 3D, DTLZ3 has 1/8th of a sphere front in 3D. DTLZ2 and 4 share the same PF as DTLZ3 but differ in the overall domain of the objective values. For example, randomly drawn solutions in the initial population, have objective values of 100-150 in DTLZ1 and 500-1500 in DTLZ3 while the range is significantly smaller e.g., with 2 to 5 in DTLZ2. It is DTLZ1 and 3 where we are expecting the most benefit of the removal of dominated solutions, whereas the insertion of our processing step should be neutral on the other functions.

The settings for the 3D DTLZ problems follow the recommended settings from [8]. We are using the algorithms and test functions as implemented in the jMetal framework [10]. For a fair comparison, all algorithms utilise the same operators with the same corresponding parameters. Specifically, a binary tournament selection operator is used for selection. Simulated Binary Crossover (SBX) and polynomial mutation with probabilities of 0.9 and (1/number of parameters) respectively, are used to create new offspring. The distribution parameters of the crossover and mutation operators are $\eta_c = 20.0$ and $\eta_m = 20.0$, respectively. A population size of 100 is used with the total number of solution evaluations limited to 100,000. Other algorithmic parameters are set as the default settings from the original IBEA [17]. 100 runs of each algorithm are performed. The version of IBEA that we are using for

comparison is the one provided by Brockhoff [4], which has fixed the hypervolume calculation bug in the original IBEA.

In the following, we will assess the performance of *m*IBEA qualitatively by inspecting the plots, quantitatively using performance indicators and by measuring the runtime. We analyse experimentally the effect that our patch has on IBEA. We do this on DTLZ1-7 and by varying the scaling parameter ρ and the population size μ.

B. Visualisation of Resultant Solution Distributions

The resultant populations under the default setting of *m*IBEA, with $\rho = 2.0, \mu = 100$, are illustrated in Figure 2. Compared to the original IBEA presented in Figure 1, the coverage obtained by *m*IBEA on DTLZ1 and DTLZ3 is a clear improvement. More specifically, the 9890 non-dominated individuals (*ndi*) found by *m*IBEA for DTLZ1 spread cross the whole front surface, while the corresponding *ndi* from IBEA is only 616 and only distributed in the extreme points, resulting in much higher *bf* (0.9935) than *m*IBEA (0.5297). Similarly, *m*IBEA improves the original IBEA by generating many more non-dominated solutions (*ndi* = 5292) and solutions between the extreme points (*bf* = 0.7897) and the central area of the front are found, while the original IBEA only obtains 182 non-dominated solutions which only lie in the corner points.

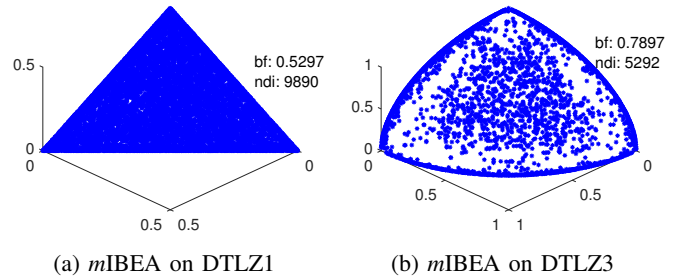(a) *m*IBEA on DTLZ1   (b) *m*IBEA on DTLZ3

Fig. 2: Visualisation of the non-dominated solutions found by *m*IBEA on DTLZ1 and DTLZ3 over 100 independent runs.

In order to provide more quantitative insights into the distribution of solutions found by each algorithm, Figure 3 compares the proportion of non-dominated solutions found in different regions on each DTLZ problem of the PF. For the three highlighted areas the proportion of the total number of non-dominated solutions contained in the area is provided, with the corresponding proportion of the actual PF surface in 3D the area represents given in the bracket. The boundaries from outermost triangle (blue area) to the innermost triangle (green area) of DTLZ1 are 0.03 and 0.1, for DTLZ3 are 0.1 and 0.2.

For example, in Figure 3b, 0.53(0.33) indicates that the fraction of *ndi* observed from *m*IBEA in the blue triangle area of DTLZ1 is 0.53, however the proportion of the total surface this area corresponds to is 0.33. In an ideal solution distribution, the fraction of *ndi* should be very close to the surface proportion, if one is interested in this coverage notion.

Figure 3 also shows that *m*IBEA could find new trade-off solutions between extreme points on both DTLZ problems that

the original IBEA is not able to. In the case of *m*IBEA, 53% and 79% of non-dominated solutions are found in the edge regions of DTLZ1 and DTLZ3 respectively. Interestingly, a decreased fraction of ndi from the outside to the inside of the PF of DTLZ1 is observed, with a 'tiny' fraction (3%) of solutions lie in the middle area (green) area of DTLZ3.

(a) IBEA on DTLZ1   (b) *m*IBEA on DTLZ1
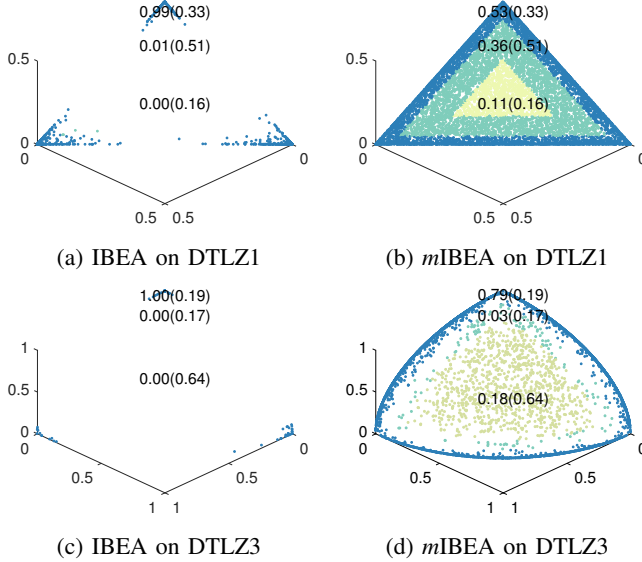
(c) IBEA on DTLZ3   (d) *m*IBEA on DTLZ3

Fig. 3: Solution distributions from IBEA and *m*IBEA on DTLZ1 and DTLZ3

C. A performance analysis of IBEA and m*IBEA using different scaling factor values

In the previous subsection, the objective values scaling factor ρ is fixed to 2.0. As the hypervolume contribution of each solution is calculated based on the scaled objective values (as described in Step2 and Step3 of IBEA, see Section II-A), here we examine the effect of modifying the value of ρ. Within IBEA and *m*IBEA, ρ directly affects the ranking of solutions, therefore influencing the preservation of elite solutions.

A series of ρ values within the range of 1.0 to 1024 are tested. From a set of preliminary experiments, we found that when $\rho \in [1.0, 1.1]$, the PFs on DTLZ functions show no obvious improvement when using either the original IBEA or *m*IBEA. However when $\rho > 1.1$, some improvement was shown by IBEA and significant improvement by *m*IBEA. However, when $\rho \geq 2$, the PFs remain relatively unchanged by both algorithms. Thus, we choose $\rho = 1.0, 1.15, 2.0, 64.0$ to plot the PFs from IBEA and *m*IBEA on DTLZ1 and 3 in Figure 4.

As we can see, the scaling parameter ρ in IBEA has little to no impact on the distribution of the solutions on DTLZ1 and DTLZ3, as the solution sets always collapse to the extreme points of the objective space no matter what the ρ value is. However, ρ can affect the performance of *m*IBEA.

We also compare the performance of IBEA and *m*IBEA w.r.t. the hypervolume and ϵ_+ indicators across the entire DTLZ family in Table III. The higher hypervolume or lower

ϵ_+, the better performance of the algorithm has. The reference point of hypervolume is set as 0.5^d for DTLZ1, where d is objective index, 1.0^d for the rest DTLZ problems, DTLZ7 where the reference point is $(1, 1, 6)$ due to the third objective. Results that are better with statistical significance (Wilcoxon rank-sum test, 5% significance level) are highlighted in bold. It shows that *m*IBEA not only improves the coverage of the resultant fronts, but also the convergence, especially on DTLZ1, DTLZ3, and DTLZ7 under all the chosen ρ values, as well as most ρ values of DTLZ6, while performing similarly on DTLZ2, 4 and 5. Only in one of the 14 cases (with standard $\rho = 2$), *m*IBEA is statistically worse, while it is comparable or statistically better in 13 of 14 cases.

In summary, the observations of PF changes from original IBEA and *m*IBEA suggest the following:

- The original IBEA performs poorly on benchmark functions with large objective spaces and comparatively small fronts, i.e. DTLZ1 and DTLZ3 and the change of objective value scaling factor $\rho(\geq 1)$ does not affect (either positively or negatively) the performance;
- Our *m*IBEA improved IBEA overall on DTLZ1-7 with wider PF coverage and better convergence w.r.t. hypervolume and ϵ_+.

Moreover, the performance comparison between NSGA-II and *m*IBEA using default settings ($\mu = 100$, $\rho = 2$) (shown in Table II) suggests that *m*IBEA still does not beat NSGA-II on most DTLZ problems. Perhaps it is because the still existing gaps (Fig. 4) and non-uniform solution distribution (Fig. 3) on the *m*IBEA fronts. Further investigations are to be conducted.

TABLE II: Performance Comparison of NSGA-II and *m*IBEA

	hypervolume		ϵ_+	
	NSGA-II	mIBEA	NSGA-II	mIBEA
DTLZ1	**0.7572**$_{0.01}$	0.7338$_{0.10}$	**0.0562**$_{0.01}$	0.0771$_{0.07}$
DTLZ2	0.3724$_{0.01}$	**0.4217**$_{0.00}$	0.1272$_{0.02}$	**0.0726**$_{0.00}$
DTLZ3	**0.3748**$_{0.01}$	0.2623$_{0.13}$	**0.1315**$_{0.02}$	0.4514$_{0.31}$
DTLZ4	**0.3472**$_{0.10}$	0.2876$_{0.14}$	**0.1738**$_{0.22}$	0.3863$_{0.34}$
DTLZ5	0.0929$_{0.00}$	**0.0931**$_{0.00}$	0.0109$_{0.00}$	0.0117$_{0.00}$
DTLZ6	**0.0869**$_{0.01}$	0.0833$_{0.01}$	**0.0180**$_{0.01}$	0.0271$_{0.01}$
DTLZ7	**0.2818**$_{0.01}$	0.2693$_{0.04}$	**0.1928**$_{0.31}$	0.7017$_{0.77}$

D. A performance analysis of IBEA and m*IBEA as the population size changes

We also investigate the performance of both the original IBEA and *m*IBEA using larger population sizes under different ρ values. Figure 5 shows that when population size goes up to 1000, population size has little affect on PF coverage for DTLZ1 by the original IBEA. It even deteriorates for DTLZ3 on which the maximum of each objective of DTLZ3 is 1.0 on the true PF. However, the proposed *m*IBEA clearly covers greater parts of the PF surface on both DTLZ1 and DTLZ3 given various ρ values.

E. Running time of IBEA and mIBEA

As a positive side-effect, the focus on non-dominated solutions within *m*IBEA results in significant speed-ups of the runtime. Table IV shows that *IBEA* on average gives an over
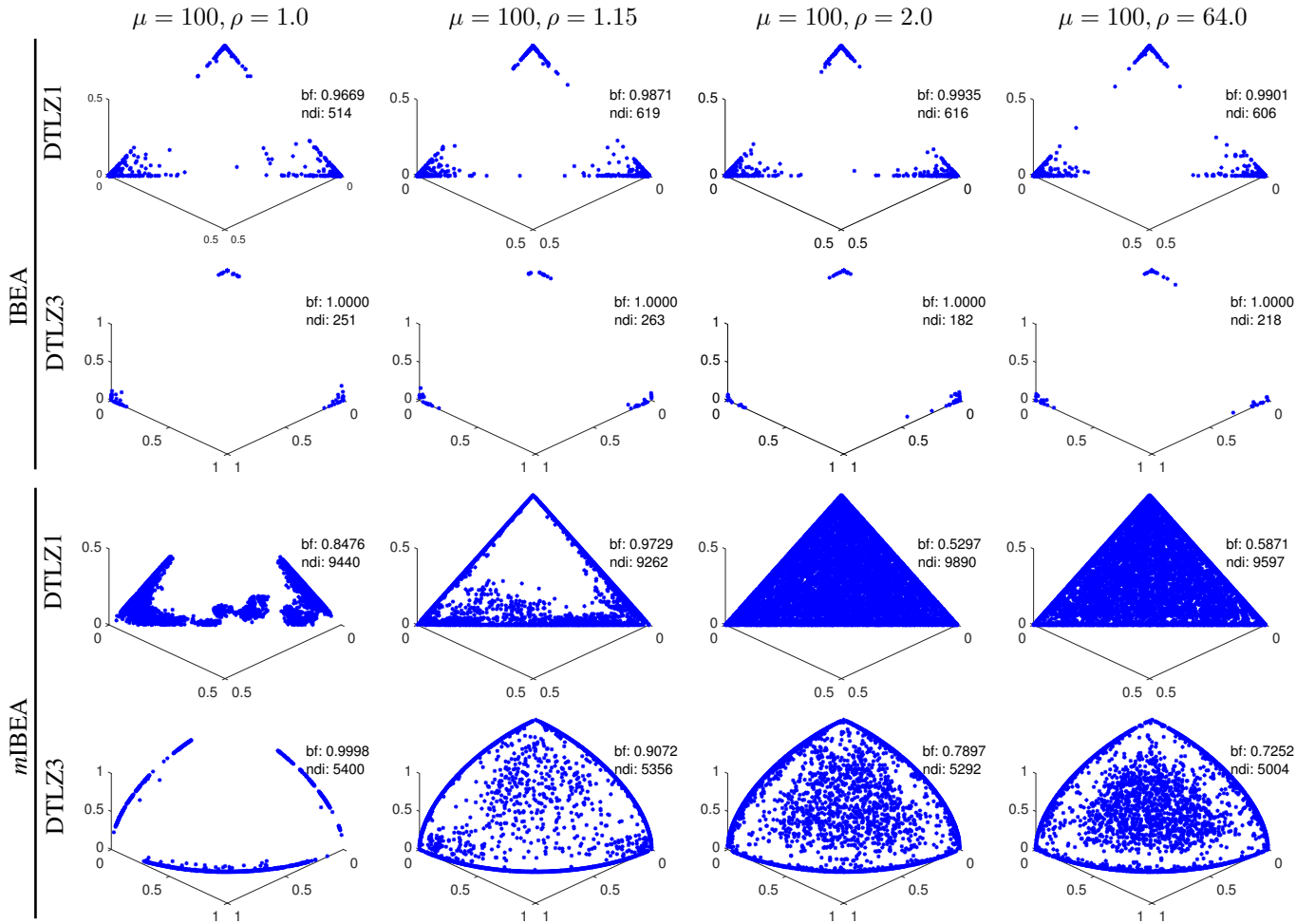
Fig. 4: PFs of DTLZ1 and DTLZ3 generated by *m*IBEA with different Rho values (1.0, 1.15, 2.0, and 64). 100 runs are performed for each experiment. The threshold of border fraction (*bf*) for DTLZ1 is 0.03, and 0.1 for DTLZ3.

8-fold speed-up compared to IBEA using population size of $\mu = 1000$ (the ρ is set as default value 2.0). The reason is that Step2 in *m*IBEA (Section III) removes dominated solutions in every generation. Therefore, fewer solutions are persevered in the population, reducing the calculation time for indicator values i.e., hypervolume difference in the original IBEA. As the indicator value calculation is pairwise (Step3 in Section II-A), n^2 indicator value calculations are required, where n is the current population size.

V. DISCUSSION AND FUTURE WORK

The proposed *m*IBEA introduces the ranking from Pareto dominance-based multi-objective evolutionary algorithm into the indicator-based algorithm during the elite solution preservation process. The empirical results from *m*IBEA over the entire DTLZ benchmark functions show that *m*IBEA significantly improves the original IBEA on the coverage of resultant Pareto fronts, as well as hypervolume and ϵ_+. Also, over 8-fold speed-ups are obtained when using a larger population size. Also note that the proposed *m*IBEA does not introduce any additional parameter to the original IBEA.

However, some gaps are still observable from original IBEA and *m*IBEA. Similar distributions can be observed from the hypervolume-based SMS-EMOA, although these are better than IBEA. We have performed an initial experiment on the SMS-EMOA's scaling parameter *offset*. However, no clear changes have been observed under different *offset* values. It suggests that the gaps from original IBEA, *m*IBEA and SMS-EMOA may be a consequence of using the hypervolume as an indicator in optimisation when faces are encountered that are (near-)parallel to at least one objective's axis.

In the future, we will explore ways to systematically examine the reasons for the gaps and improve *m*IBEA by redirecting its attention away from extreme solutions, without introducing arbitrary decisions, magic numbers or explicitly defined weights or preferences in the objective space.
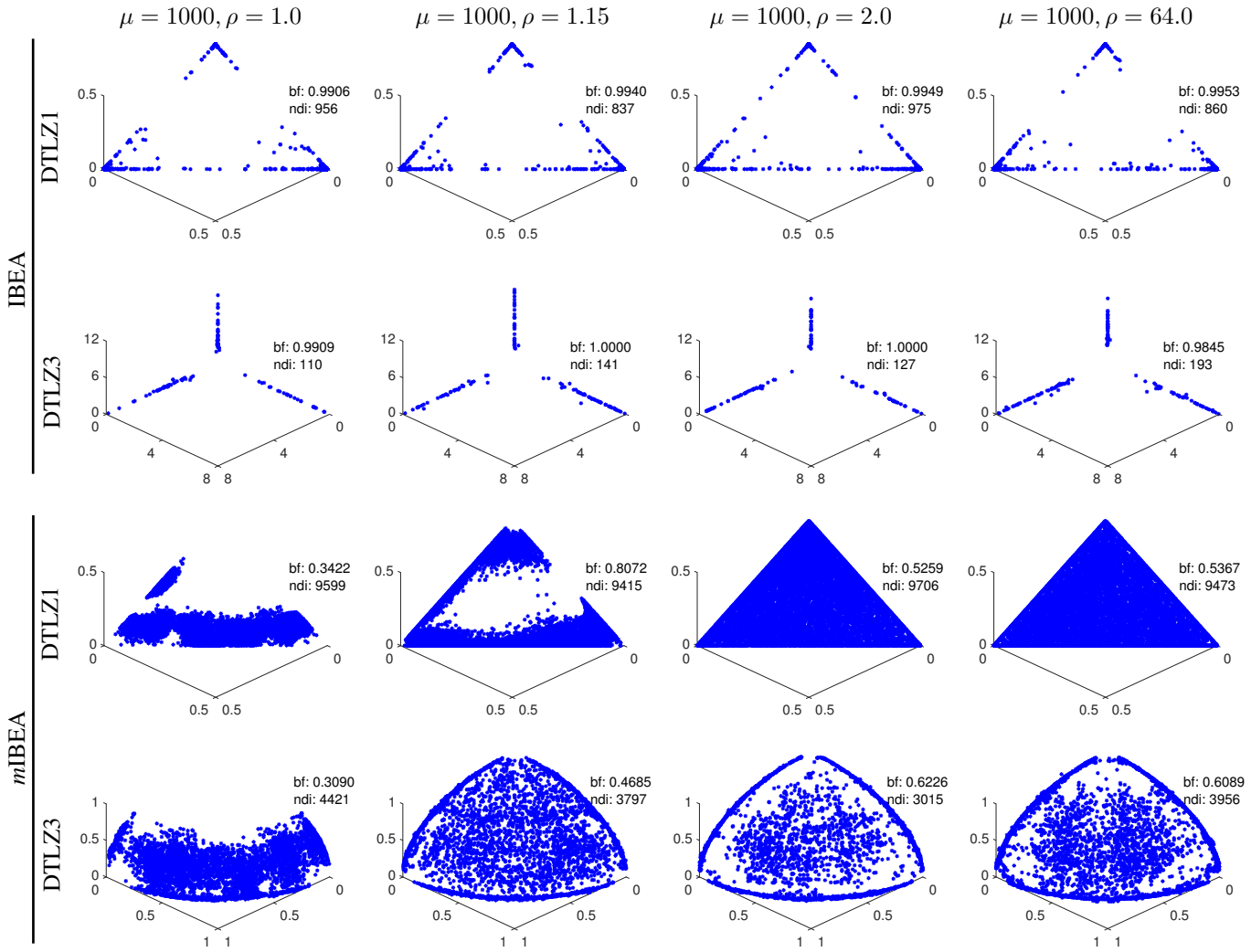
Fig. 5: PFs of DTLZ1 and DTLZ3 generated by IBEA using population size 1000 with different ρ values. 100 runs are performed for each experiment. The threshold of border fraction (bf) for DTLZ1 is 0.03 and DTLZ3 is 0.1.

REFERENCES

[1] T. Back, U. Hammel, and H.-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1): 3–17, 1997.

[2] N. Beume, B. Naujoks, and M. Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181 (3):1653–1669, 2007.

[3] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1198–1203. AAAI, 2011.

[4] D. Brockhoff. A bug in the multiobjective optimizer ibea: Salutary lessons for code release and a performance reassessment. In *Int. Conference on Evolutionary Multi-Criterion Optimization*, pages 187–201. Springer, 2015.

[5] S. Chand and M. Wagner. Evolutionary many-objective

[6] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

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

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. *Scalable test problems for evolutionary multiobjective optimization*. Springer, 2005.

[9] F. E. C. Duran, C. Cotta, and A. J. Fernández-Leiva. A comparative study of multi-objective evolutionary algorithms to optimize the selection of investment portfolios with cardinality constraints. In *European Conference on the Applications of Evolutionary Computation*, pages 165–173. Springer, 2012.

[10] J. J. Durillo and A. J. Nebro. jmetal: A java framework

optimization: A quick-start guide. *Surveys in Operations Research and Management Science*, 20(2):35 – 42, 2015.

TABLE III: The subtables show the hypervolume and the epsilon $mean_{std}$ values of IBEA and mIBEA with different ρ values but same $\mu = 100$ on DTLZ1-7.

DTLZ1

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.1885_{0.08}$	$\mathbf{0.2368_{0.05}}$	$\mathbf{0.2883_{0.02}}$	$0.3420_{0.05}$
1.15	$0.1809_{0.07}$	$\mathbf{0.3796_{0.08}}$	$0.2910_{0.02}$	$\mathbf{0.2658_{0.04}}$
1.41	$0.1699_{0.07}$	$\mathbf{0.4178_{0.12}}$	$0.2938_{0.02}$	$\mathbf{0.2709_{0.06}}$
2	$0.1773_{0.08}$	$\mathbf{0.7338_{0.10}}$	$0.2919_{0.02}$	$\mathbf{0.0771_{0.07}}$
64	$0.1514_{0.08}$	$\mathbf{0.7487_{0.10}}$	$0.2978_{0.02}$	$\mathbf{0.0632_{0.07}}$
1024	$0.1550_{0.08}$	$\mathbf{0.7480_{0.11}}$	$0.2977_{0.02}$	$\mathbf{0.0635_{0.07}}$

DTLZ2

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.0084_{0.01}$	$\mathbf{0.1144_{0.02}}$	$0.9905_{0.01}$	$\mathbf{0.7452_{0.05}}$
1.15	$0.4250_{0.00}$	$0.4251_{0.00}$	$0.0612_{0.00}$	$0.0610_{0.00}$
1.41	$0.4234_{0.00}$	$0.4236_{0.00}$	$0.0683_{0.00}$	$0.0678_{0.00}$
2	$0.4215_{0.00}$	$\mathbf{0.4217_{0.00}}$	$0.0724_{0.00}$	$0.0726_{0.00}$
64	$0.4196_{0.00}$	$0.4195_{0.00}$	$0.0730_{0.00}$	$0.0725_{0.00}$
1024	$0.4195_{0.00}$	$0.4194_{0.00}$	$0.0734_{0.00}$	$0.0726_{0.00}$

DTLZ3

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.0866_{0.02}$	$\mathbf{0.4211_{0.00}}$	$\mathbf{0.4218_{0.00}}$	$0.8057_{0.08}$
1.15	$0.0009_{0.00}$	$\mathbf{0.1722_{0.08}}$	$\mathbf{0.4218_{0.00}}$	$0.7372_{0.14}$
1.41	$0.0004_{0.00}$	$\mathbf{0.2115_{0.12}}$	$\mathbf{0.4223_{0.00}}$	$0.6313_{0.028}$
2	$0.0011_{0.01}$	$\mathbf{0.2623_{0.13}}$	$\mathbf{0.4217_{0.00}}$	$0.4514_{0.31}$
64	$0.0009_{0.00}$	$\mathbf{0.3331_{0.09}}$	$0.4218_{0.00}$	$\mathbf{0.1955_{0.12}}$
1024	$0.0009_{0.00}$	$\mathbf{0.3257_{0.09}}$	$0.4221_{0.00}$	$\mathbf{0.2075_{0.12}}$

DTLZ4

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.1913_{0.13}$	$\mathbf{0.2215_{0.14}}$	$0.6555_{0.19}$	$\mathbf{0.6206_{0.19}}$
1.15	$0.2501_{0.14}$	$0.2732_{0.15}$	$0.4840_{0.33}$	$0.4205_{0.35}$
1.41	$0.2680_{0.14}$	$0.2764_{0.14}$	$0.4413_{0.33}$	$0.4154_{0.34}$
2	$0.3023_{0.12}$	$0.2876_{0.14}$	$0.3596_{0.31}$	$0.3863_{0.34}$
64	$0.2573_{0.15}$	$0.2530_{0.14}$	$0.4548_{0.35}$	$0.4688_{0.34}$
1024	$0.2592_{0.14}$	$0.2613_{0.14}$	$0.4577_{0.33}$	$0.4497_{0.34}$

DTLZ5

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.0030_{0.00}$	$\mathbf{0.0158_{0.01}}$	$0.9961_{0.01}$	$\mathbf{0.9772_{0.01}}$
1.15	$0.0082_{0.01}$	$\mathbf{0.0252_{0.01}}$	$0.9879_{0.02}$	$\mathbf{0.9550_{0.04}}$
1.41	$0.0936_{0.00}$	$0.0937_{0.00}$	$0.0226_{0.00}$	$\mathbf{0.0202_{0.00}}$
2	$0.0930_{0.00}$	$\mathbf{0.0931_{0.00}}$	$0.0129_{0.00}$	$\mathbf{0.0117_{0.00}}$
64	$0.0917_{0.00}$	$0.0916_{0.00}$	$0.0110_{0.00}$	$0.0114_{0.00}$
1024	$0.0916_{0.00}$	$0.0916_{0.00}$	$0.0109_{0.00}$	$0.0113_{0.00}$

DTLZ6

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$\mathbf{0.0086_{0.03}}$	$0.0031_{0.01}$	$\mathbf{0.7863_{0.27}}$	$1.0080_{0.02}$
1.15	$\mathbf{0.0407_{0.05}}$	$0.0136_{0.01}$	$\mathbf{0.5806_{0.49}}$	$0.9760_{0.04}$
1.41	$0.0823_{0.01}$	$\mathbf{0.0853_{0.01}}$	$0.0476_{0.02}$	$\mathbf{0.0322_{0.02}}$
2	$0.0789_{0.01}$	$\mathbf{0.0833_{0.01}}$	$0.0464_{0.02}$	$\mathbf{0.0271_{0.01}}$
64	$0.0768_{0.01}$	$\mathbf{0.0844_{0.01}}$	$0.0491_{0.02}$	$\mathbf{0.0211_{0.01}}$
1024	$0.0787_{0.01}$	$\mathbf{0.0828_{0.01}}$	$0.0468_{0.01}$	$\mathbf{0.0238_{0.01}}$

DTLZ7

ρ	hypervolume		ϵ_+	
	IBEA	mIBEA	IBEA	mIBEA
1	$0.0243_{0.01}$	$\mathbf{0.0579_{0.02}}$	$3.2996_{0.05}$	$\mathbf{3.1711_{0.08}}$
1.15	$0.1886_{0.09}$	$\mathbf{0.2617_{0.06}}$	$2.0409_{0.68}$	$\mathbf{1.5644_{0.23}}$
1.41	$0.2115_{0.07}$	$\mathbf{0.2774_{0.05}}$	$1.6374_{0.96}$	$\mathbf{0.7536_{0.58}}$
2	$0.2512_{0.05}$	$\mathbf{0.2693_{0.04}}$	$1.0914_{1.06}$	$\mathbf{0.7017_{0.77}}$
64	$0.2456_{0.03}$	$\mathbf{0.2762_{0.02}}$	$1.3796_{0.99}$	$\mathbf{0.4551_{0.64}}$
1024	$0.2466_{0.03}$	$\mathbf{0.2741_{0.03}}$	$1.3396_{0.97}$	$\mathbf{0.5318_{0.81}}$

TABLE IV: Average running time comparison (in minutes).

	Running Time Comparison (in minutes)			
	$\mu = 100$		$\mu = 1000$	
	IBEA	mIBEA	IBEA	mIBEA
DTLZ1	0.15	0.08	7.86	1.22
DTLZ2	0.15	0.10	7.62	1.46
DTLZ3	0.15	0.06	8.10	0.24
DTLZ4	0.14	0.11	7.98	1.56
DTLZ5	0.16	0.11	8.01	0.97
DTLZ6	0.15	0.11	8.04	0.21
DTLZ7	0.16	0.10	8.03	1.15

for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[11] W. Li, E. Özcan, and R. John. Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimisation. *Renewable Energy*, 2016.

[12] T. Tušar and B. Filipič. Differential evolution versus genetic algorithms in multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 257–271. Springer, 2007.

[13] M. Wagner, K. Bringmann, T. Friedrich, and F. Neumann. Efficient optimization of many objectives by approximation-guided evolution. *European Journal of Operational Research*, 243(2):465 – 479, 2015.

[14] L. Zhao. *Performance of Multiobjective Algorithms on Varying Test Problem Features Using the WFG Toolkit*. PhD thesis, Citeseer, 2007.

[15] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In *IEEE Congress on Evolutionary Computation*, pages 892–899. IEEE, 2006.

[16] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[17] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842. Springer, 2004.

[18] E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: improving the strength pareto evolutionary algorithm, 2001.

[19] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.