J Glob Optim (2013) 55:579–595
DOI 10.1007/s10898-012-9966-4

New fitness sharing approach for multi-objective genetic algorithms

Hyoungjin Kim · Meng-Sing Liou

Received: 11 January 2011 / Accepted: 31 July 2012 / Published online: 19 August 2012

Abstract A novel fitness sharing method for MOGA (Multi-Objective Genetic Algorithm) is proposed by combining a new sharing function and sided degradations in the sharing process, with preference to either of two close solutions. The modified MOGA adopting the new sharing approach is named as MOGAS. Three different variants of MOGAS are tested; MOGASc, MOGASp and MOGASd, favoring children over parents, parents over children and solutions closer to the ideal point, respectively. The variants of MOGAS are compared with MOGA and other state-of-the-art multi-objective evolutionary algorithms such as IBEA, HypE, NSGA-II and SPEA2. The new method shows significant performance improvements from MOGA and is very competitive against other Evolutionary Multi-objective Algorithms (EMOAs) for the ZDT and DTLZ test functions with two and three objectives. Among the three variants MOGASd is found to give the best results for the test problems.

Keywords Genetic algorithms · Multi-objective optimization · Niching · Sharing Function

1 Introduction

Multi-objective optimization methods are getting more attention as multiple conflicting objectives are increasingly considered in MDO (Multi-disciplinary Design Optimization) of modern engineering problems. Unlike single objective optimization problems, a solution for a multi-objective optimization problem (MOP) is not a single point but, in general, a set of points known as a Pareto optimal front.

H. Kim (✉)
Science Applications International Corporation, MS 5-10, 21000 Brookpark Road, Cleveland, OH 44135, USA
e-mail: hyoungjinkim1@gmail.com

M.-S. Liou
Aeropropulsion Division, NASA Glenn Research Center, MS 5-10, 21000 Brookpark Road,
Cleveland, OH 44135, USA

Evolutionary Algorithms (EAs) are search algorithms inspired by genetic evolution and natural selection. EAs are well suited to MOPs because they are based on populations rather than a single solution and therefore can be applied to find distributed Pareto optimal solutions in a single run.

Among approaches for EMOAs (Evolutionary Multi-Objective Algorithms), Pareto-based EMOAs have been very popular over the last decades, in which two criteria are adopted for selection of better solutions: a ranking procedure based on the Pareto-dominance concept and a diversity preservation based on Euclidean distances between solutions. Examples of Pareto-based EMOAs are MOGA (Multi-Objective Genetic Algorithm) [1], NSGA-II (Nondominated Sorting Genetic Algorithm-II) [2], SPEA2 (Strength Pareto Evolutionary Algorithm 2) [3], ε-MOEA [4].

Recently, indicator-based EMOAs have been suggested [5–7] by employing a single quality indicator for fitness calculations rather than separately considering convergence and diversity as Pareto-based algorithms do. IBEA (Indicator Based Evolutionary Algorithms) [5], HypE (Hypervolume Estimation algorithm for multi-objective optimization) [6], and SMS-EMOA (S Metric Selection-EMOA) [7] are among those. The indicator-based algorithms have been reported to show superior performances to those of Pareto-based algorithms, especially for MOPs with more than two objectives [5–8]. However, a heavy computational overhead for the calculation of a quality indicator is a major drawback for the indicator-based algorithms [6].

One of the main issues for Pareto-based EMOAs is how to preserve diversity among Pareto optimal solutions. Fonseca and Fleming [1] introduced the concept of niching among solutions through the use of a sharing function, which was originally suggested for single objective genetic algorithms for multi-modal functions [9]. In NSGA-II [2], the solution density is estimated by the volume of a cuboid defined by the nearest neighbors and less crowded solutions are emphasized in the selection process. In the hyperbox approach, the objective space is divided into uniform grid cells, and solutions in less crowded hyperboxes are emphasized [10] or only a single solution is taken from each grid cell [4].

Although the sharing function approach has been widely adopted in many Pareto-based EMOAs [11], its performance for the diversity preservation has also been pointed out as not so significant [12]. Furthermore, a potential defect could appear with the standard sharing approach when combined with elite-preserving strategies such as the best-N selection approach, as will be discussed in Sect. 3. Basically, the weakness comes from the fact that the standard sharing function approach makes degradations to both of two close solutions.

In this paper we propose a modified MOGA by employing a new sharing approach that combines a novel sharing function and a sided degradation into the baseline MOGA. The new sharing approach allows significant performance improvement when it is coupled with a proper preference between two crowded solutions. The modified MOGA, named hereafter as MOGAS (MOGA with a new fitness Sharing), is compared with MOGA and several other state-of-the-art EMOAs for standard test functions with two or three objectives.

The remainder of this paper is organized as follows: In Sect. 2, we describe the baseline MOGA, including the Pareto ranking, the standard sharing function, evolutionary operators and an elite-preserving strategy. The new sharing approach is explained in Sect. 3. Assessment of performance of the proposed method against other prominant methods on standard test functions is presented in Sect. 4. Finally, concluding remarks are made in Sect. 5.

2 Baseline multi-objective genetic algorithms

2.1 Definition of multi-objective optimization problems

MOPs are defined in a general form with all objectives to be minimized as follows:

$$\begin{aligned} &\text{Minimize}\,(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) && \\ &\text{Subject to}\, g_j(\mathbf{x}) \geq 0, && j = 1, 2, \ldots, J; \\ &\quad h_k(\mathbf{x}) = 0, && k = 1, 2, \ldots, K; \\ &\quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i = 1, 2, \ldots, n_{dv}, \end{aligned} \tag{1}$$

where M, J, K and n_{dv} are the number of objective functions, inequality constraints, equality constraints and decision variables, respectively. In this study, unconstrained MOPs having side constraints for decision variables are considered. The dominance relation between two solutions of a MOP is defined by the following two conditions:

Solution a dominates solution b if

(1) $f_j(a) \leq f_j(b)$ for all $j \in \{1, \ldots, M\}$, and
(2) $f_k(a) < f_k(b)$ for at least one $k \in \{1, \ldots, M\}$.

In other words, solution a dominates solution b if a is not worse than b in all objectives and a is strictly better than b in at least one objective. Solutions that are not dominated by any other solution in the population are said to be non-dominated and form the Pareto optimal set.

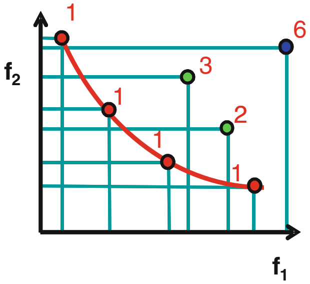
2.2 Pareto ranking

The Pareto ranking is a process for ranking individuals in a population based on their distances from the Pareto optimal front: solutions that are closer to the Pareto front have higher ranks and vice versa. The closeness to the Pareto front is determined by dominancy relations between individuals in the population. In Fonseca and Fleming's MOGA [1], r_i, a rank of each individual i, is assigned as:

$$r_i = 1 + n_i, \tag{2}$$

where n_i is the number of solutions that dominate individual i. Since n_i is zero for non-dominated solutions in the population, nondominated individuals are assigned a rank of 1. A sample illustration of Fonseca and Fleming's Pareto ranking is shown in Fig. 1.

Fig. 1 Fonseca and Fleming's Pareto ranking [1]



Based on the Pareto ranking, a fitness value is assigned to each solution so that solutions in better ranks have larger fitness values by the following expression:

$$F_i = N - 0.5\left(\mu\left(r_i\right) - 1\right) - \sum_{k=1}^{r_i - 1} \mu\left(k\right), \tag{3}$$

where N is the population size, and $\mu(r_i)$ is the number of individuals in rank r_i. The fitness value in Eq. (3) varies from N to 1, depending on r_i, the rank of each individual i and μ, the number of individuals in each rank.

2.3 Standard fitness sharing

In order to impose diversity in a population, MOGA employs the sharing function approach for each solution i. First, a normalized distance between any two individuals i and j of the same rank is calculated as a proximity measure in the objective space as follows:

$$d_{ij} = \sqrt{\sum_{k=1}^{M}\left(\frac{f_k^{(i)} - f_k^{(j)}}{f_k^{\max} - f_k^{\min}}\right)^2}, \tag{4}$$

where $f_k^{(i)}$ and $f_k^{(j)}$ are the kth objective function for solutions $i$and$j$, respectively, and f_k^{max} and f_k^{min} are the maximum and minimum objective function values of the k^{th} objective function for all nondominated individuals in the population. The standard sharing function suggested by Goldberg and Richardson [9] is defined as follows with a normalized niche radius σ_{share}:

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right) & \text{if} \quad d_{ij} < \sigma_{share}; \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where the niche size σ_{share} in the normalized objective space is determined by the following expression [1] :

$$\left(1 + \sigma_{share}\right)^M - 1 = N\left(\sigma_{share}\right)^M. \tag{6}$$

A niche count for individual i is calculated by summing up the sharing function for all individuals of the same rank in the population as follows:

$$nc_i = \sum_{j=1}^{\mu(r_i)} sh\left(d_{ij}\right). \tag{7}$$

The niche count nc_i is always greater than 1 because $sh(d_{ij}) = 1$ holds for $i = j$, and $sh(d_{ij})$ varies from 0 to 1 otherwise. Finally, the assigned fitness is reduced by dividing the fitness F_i given in Eq. (3) by the niche count as follows:

$$F_i' = \frac{F_i}{nc_i}. \tag{8}$$

With the degradation made to each individual in the population, the shared fitness values of solutions in each rank are now re-scaled so that the averaged fitness value before degradation in each rank is restored.

2.4 Selection of mating pairs

After completing the fitness sharing, a pool of N individuals with assigned rankings is available for selection and mating process. The Roulette Wheel Selection (RWS) [12] is the simplest proportionate selection approach, which is analogous to a roulette wheel with the size of each slice being proportional to the fitness of an individual in the population. The N individuals are mapped onto contiguous segments of a straight line. The length of each segment is same as the fitness of the individual. Therefore, the total length of the line equals to the summation' of N fitness values, SUM_F. For the shared fitness values in Eq. (8),SU can be calculated as $SUM_F = \sum_{i=1}^{N} F_i'$.

For the selection of N mating individuals, RWS requires N random numbers to be generated in the range of $[0, SUM_F]$. RWS provides a zero bias between an individual's normalized fitness and its expected probability of reproduction. Minimum spread of the number of offspring of an individual, however, is not guaranteed [13].

In the present study, for the proportionate selection the Stochastic Universal Sampling (SUS) [13] is adopted, which also works on the same mapping onto a straight line with a total length of SUM_F as RWS does, but requires only one random number generation and provides zero bias and minimum spread. In SUS, N mating solutions are selected by the following formula:

$$l_i = l_0 + (i-1) \times (SUM_F/N), \quad \text{for} \quad i = 1, 2, 3, \ldots, N, \tag{9}$$

where l_0 is a random number generated in $[0, SUM_F/N]$. N individuals in the line segment corresponding to l_i's are selected for the mating population. $N/2$ mating pairs are then randomly selected from the N mating solutions without any mating restriction.

2.5 Evolutionary operators

MOGA can be coupled with general evolutionary operators such as crossover and mutation operators for the evolution of population. Since all the test functions in this study have real-valued decision variables, we employ real-parameter operators such as the simulated binary crossover (SBX) [14] for the crossover and the polynomial mutation [15] for the mutation.

The SBX operator mimics the operation of the single point crossover of binary-coded genetic algorithms by using a probability density function. The crossover operator is applied to each decision variable of all mating pairs with a crossover probability of p_c. Two children solutions are obtained from two parents as follows:

$$\begin{aligned} C_i^{(1)} &= 0.5[\left(1+\beta_{qi}\right) P_i^{(1)} + \left(1-\beta_{qi}\right) P_i^{(2)}], \\ C_i^{(2)} &= 0.5[\left(1-\beta_{qi}\right) P_i^{(1)} + \left(1+\beta_{qi}\right) P_i^{(2)}], \end{aligned} \quad \text{for} \quad i = 1, 2, 3, \ldots, n_{dv}, \tag{10}$$

where $C_i^{(1)}$ and $C_i^{(2)}$ are the ith component of solution vectors of the two children solutions. Similarly, $P_i^{(1)}$and $P_i^{(2)}$are the ith component of solution vectors of the parents. A spread factor β_{qi}in Eq. (10) is determined by a uniform random number $u_i \in [0, 1]$:

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_p+1}}, & \text{if } u_i \le 0.5; \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_p+1}}, & \text{otherwise.} \end{cases} \tag{11}$$

If $u_i \le 0.5$, then $\beta_{qi} \le 1$ from Eq. (10), which means that Eq. (10) provides an interpolation so that the offspring are generated in between the two parents. If $u_i > 0.5$, β_q is greater than 1, and Eq. (10) conducts an extrapolation of the two parent values. The distribution parameter $\eta_p$is a nonnegative real number. A larger value of $\eta_p$gives a higher probability of creating offspring near the parents; a smaller value of η_p generates children farther away from the parents.

After the crossover operator is conducted, the polynomial mutation operator is applied to each decision variable of all children solutions obtained by the crossover operator with a mutation probability of p_m. A component of a child solution vector is perturbed as follows:

$$C_i^{perturbed} = C_i + \delta_i (x_i^U - x_i^L) \quad \text{for} \quad i = 1, 2, 3, \ldots, n_{dv}, \tag{12}$$

where δ is calculated by the following formula with a uniform random number $r \in [0,1]$:

$$\delta_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m}} - 1, & \text{if}\, r_i \le 0.5, \\ 1 - (2\,(1 - r_i))^{\frac{1}{\eta_m + 1}}, & \text{otherwise,} \end{cases} \quad \text{for} \quad i = 1, 2, 3, \ldots, n_{dv}. \tag{13}$$

The probability distributions of the crossover and mutation operators are modified so that no child solution is created outside of the allowed range [x^L, x^U]. The implementation details of the evolutionary operators follow those given in PISA [16].

2.6 Elite preservation: best-N strategy

The Best-N strategy can also be referred to as the ($\mu + \lambda$)-selection, in which μ and λ stand for the number of parents and children respectively, and parents compete with their children to be selected for the next generation [17]. In the present study, μ and λ are all equals to N. The procedure for the Best-N strategy is as follows.

(1) After performing the evolutionary operators, N parents and N children solutions are combined to build a solution set of size $2N$.
(2) The combined set goes through the Pareto ranking. Top-ranked solutions in the set are preserved in an external archive, which contains all nondominated solutions found in the evolution history, under the condition that the best ranked solutions are still nondominated in the archive.
(3) Crowded solutions in the $2N$ set are degraded by a fitness sharing approach and then sorted in a descending order for the shared fitness.
(4) The first half of the sorted $2N$ solutions are taken as the best-N solutions which become parents for the next generation.

All the nondominated solutions are stored without any restriction on the size of the external archive. In general cases, there might be a need to limit the upper bound of the archive size. The evolutionary procedure of generating a new solution set is shown in Fig. 2a. And the overall procedure of the elite-preserving MOGA is shown in Fig. 2b. A similar approach of the elite-preserving MOGA has been successfully applied to various engineering MDO problems [18,19]. However, no explicit performance comparisons have been made with other state-of-the-art EMOAs.

Recently, a research work focusing on archival strategies dealing with potential drawbacks of the chosen archiving strategy was presented [20].

3 New fitness sharing method

When the distance between two solutions of the same rank is less than a specified niche size, the standard fitness sharing approach degrades both individuals, as described in Sect. 2.3 to

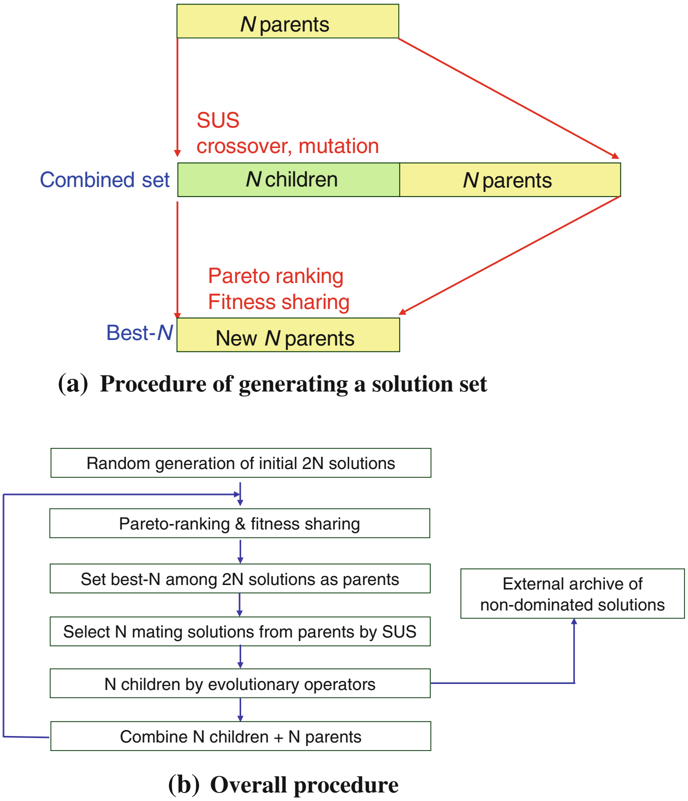
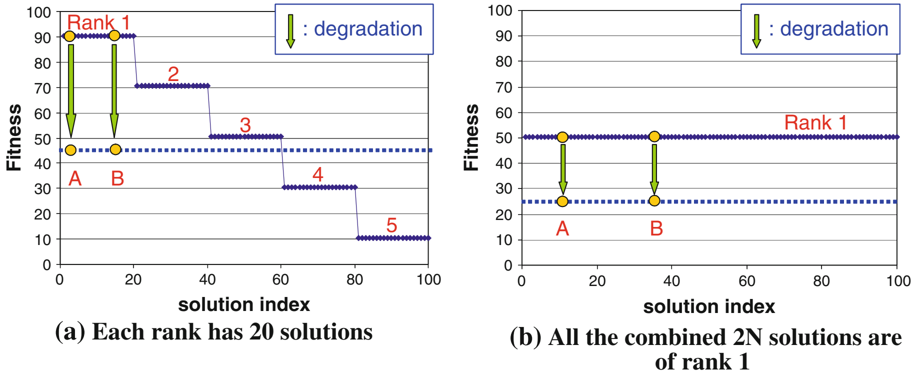

N parents
SUS
crossover, mutation
Combined set
N children
N parents
Pareto ranking
Fitness sharing
Best-*N*
New *N* parents
(a) Procedure of generating a solution set
Random generation of initial 2N solutions
Pareto-ranking & fitness sharing
Set best-N among 2N solutions as parents
Select N mating solutions from parents by SUS
N children by evolutionary operators
Combine N children + N parents
External archive of non-dominated solutions
(b) Overall procedure

Fig. 2 Elite-preserving MOGA

preserve diversity. The degradation is made in such a way that fitness values of the two individuals having the same objective vector become half of the original fitness value. Therefore, the sum of the degraded fitness values of the two coinciding individuals is the same as the fitness of one solution before the degradation. This is reasonable in a sense because for a proportionate selection operator, the sum of probabilities for the two degraded solutions to be picked up in the selection process is same as that of a single individual before the degradation.

However, if an elite preserving strategy, the best-N approach for instance, is adopted, the standard sharing method may become problematic because neither of the degraded non-dominated solutions may survive in the reduction process. A degraded shared fitness for a crowded individual may become smaller than the median of fitness values of the combined set of $2N$ solutions while only the best N solutions survive in the reduction process and undergo the selection process by a proportionate operator.

Some illustrative examples of the potential weakness are shown in Fig. 3. In Fig. 3a, the combined $2N$ solutions are assumed to be evenly distributed to five ranks with a population size $N = 50$. If solutions A and B in rank 1 have the same objective vector, they will be degraded to be half of the assigned fitness values, from 90 to 45 for instance. Then it is likely that the two solutions will have little chance to be included in the best N solutions. In Fig. 3b, a converged case is illustrated with all the $2N$ solutions being of rank 1, where even a very little degradation can result in losing both solutions. This is the reason why a sided degradation is desirable, its details will be discussed in what follows.



(a) Each rank has 20 solutions

(b) All the combined 2N solutions are of rank 1

Fig. 3 Illustration of potential defect of the standard sharing function approach. Population size N = 50, and the graphs show combined 2N solutions

Although nondominated solutions in the evolution history are saved in the external archive, losing two very close good solutions in the reduction process is clearly not desirable from a convergence or diversity aspect of MOGA. Therefore, we propose applying a sided degradation to mitigate this deficiency: by keeping the shared fitness of one individual the same and degrading the other individual, so that one solution has a better chance to survive than the other. For this purpose, we define a new sharing function as follows:

$$sh_{new}\left(d_{ij}\right) = \min\left(\left(\frac{d_{ij}}{\sigma_{share}}\right)^{0.1}, 1\right), \tag{14}$$

where d_{ij} is the normalized distance between two solutions calculated by Eq. (4), and the exponent 0.1 is found to give good performance in numerical experiments. It is noted that this definition of the new sharing function gives an opposite trend to the conventional sharing function in Eq. (5). The new sharing function diminishes as two solutions i and j of the same rank cluster together and becomes unity as the distance gets larger than the niche size σ_{share}.

With the new sharing function, the fitness degradation is applied to either of the two solutions, depending on a practitioner's preference strategies. Three different preferences are tested in this study: favoring children over parents, favoring parents over children, and favoring solutions closer to the ideal point. A niche count nc is defined for each solution to accumulate the information of preference. The initial value of nc is set to be 1.

3.1 Favoring children over parents

The combined set of $2N$ solutions is formed in such a way that N children are put in the first half ($i = 1, 2, 3, \ldots, N$) and N parents are put in the second half ($i = N+1, N+2, \ldots, 2N$) of the combined set as shown in Fig. 2a. Then, favoring children over parents can be simply conducted by degrading individuals with larger indices among the two crowded solutions. For individuals i and j in a population such that $i < j$ and *rank(i)* = *rank(j),* only individual j is degraded as follows while individual i is not altered:

$$nc_j := nc_j \cdot sh_{new}(d_{ij}). \tag{15}$$

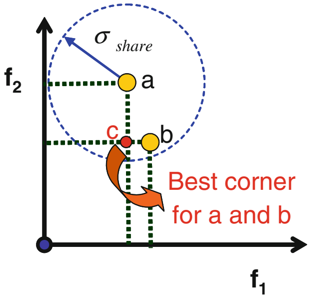


Fig. 4 Distance-based priority for fitness sharing: c is the best corner for solutions a and b. The solution closer to the best corner is b in this case assuming each objective function coordinate is normalized by the difference between the maximum and minimum objective function values of the Pareto solutions. a is degraded while b is kept unchanged following Eq. (18)

3.2 Favoring parents over children

Similarly, favoring parents can be made by degrading individuals of smaller indices. For individuals i and j in a population such that $i < j$ and *rank(i)* = *rank(j),* only individual i is degraded as follows while individual j is kept as the same:

$$nc_i := nc_i \cdot sh_{new}\left(d_{ij}\right). \tag{16}$$

3.3 Favoring solutions closer to the ideal point

Another option is to favor individuals having a smaller distance to the ideal point, which is an M dimensional objective vector composed of best feasible objective values. As a reference value for the distance-based degradation, we introduce a normalized composite objective function:

$$Z\left(\mathbf{x}_i\right) = \sum_{m=1}^{M} \frac{f_m\left(\mathbf{x}_i\right) - f_m^{\min}}{f_m^{\max} - f_m^{\min}}, \tag{17}$$

where $f_m(\mathbf{x}_i)$ is the mth objective function for the solution $\mathbf{x}_i$, and $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of the m^{th} objective function for all nondominated solutions.

For individuals i and j in a population such that $i < j$ and $rank(i) = rank(j)$, we degrade a niche count nc as follows:

$$\begin{aligned} nc_i &:= nc_i \cdot sh_{new}\left(d_{ij}\right) \text{ if } Z\left(\mathbf{x}_i\right) \geq Z\left(\mathbf{x}_j\right); \\ nc_j &:= nc_j \cdot sh_{new}\left(d_{ij}\right) \text{ otherwise.} \end{aligned} \tag{18}$$

Favoring the smaller value of the composite objective Z is equivalent to favoring the closer solution to the best corner formed by the two solutions i and j under consideration as illustrated in Fig. 4. Similarly, in ε-MOEA [4], among solutions in a hypercube, the closest to the best corner of the hypercube was selected as a representative solution of the hypercube.

Finally, the assigned fitness F_i given in Eq. (3) is reduced to a shared fitness by multiplying the fitness value and the niche count, nc, calculated by one of Eqs. (15), (16) and (17) as follows:

$$\tilde{F}_i = nc_i \cdot F_i. \tag{19}$$

Unlike the standard sharing approach in Sect. 2.3, no scaling is made on the shared fitness after the degradation in Eq. (19) so that the relative magnitudes of fitness of non-degraded solutions of different ranks remain the same. After the fitness sharing is finished, extreme solutions having the minimum function value for each objective function among the best ranked solutions in the population are given the maximum shared fitness value so that they have the biggest chance to be selected as parents of the next generation.

The new sharing method can be applied to either the phenotypic or genotypic space. If one is interested in obtaining a diverse distribution in the Pareto front, a phenotypic sharing should be used. If one is more interested in different designs that might have similar vector positions in the objective space, a genotype sharing would be preferable. In this study, we have more interest in the phenotypic space because our main purpose is to validate the performance of proposed algorithms, which is measured in the objective space. The modified MOGA utilizing the new sharing function approach is called hereafter MOGAS (MOGA with a new fitness Sharing).

4 Experimental results

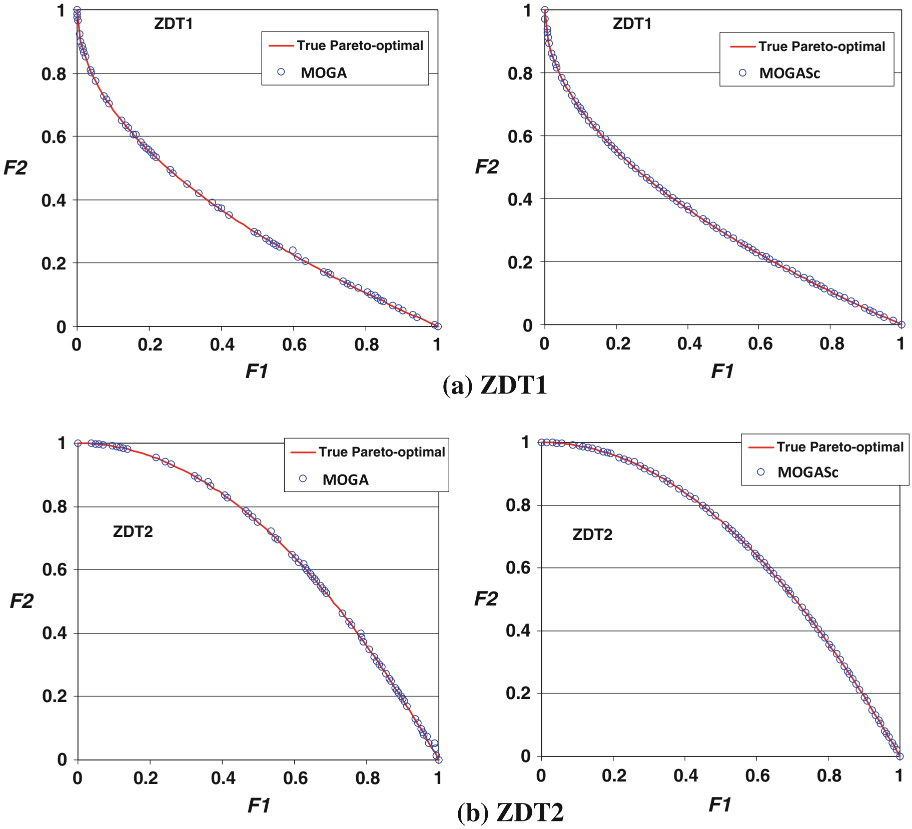
In this section, we shall show performances of the new algorithm MOGAS in three variant forms; MOGASc represents preferring children in the fitness degradation procedure, MOGASp represents favoring parents over children, and MOGASd represents favoring solutions with the smaller scalarized objective functions. The results will also be compared with those by the state-of-the-art EMOAs such as IBEA, HypE, NSGA-II, and SPEA2 available in PISA [16,23] as well as MOGA.

PISA is a platform and programming language independent interface for search algorithms. It splits an optimization problem into two modules. One module contains all parts for the optimization problem such as problem representation and function evaluation. The other module includes the selection process that is independent of the optimization problem. These two modules are implemented as separate programs and communicate through a text-based interface. PISA includes a library of modules for many optimization problems, selection modules of multi-objective optimizers and performance assessment modules. It is very convenient and useful for performance evaluation and comparison of state-of-the-art EMOAs.

As test functions for validation of the proposed algorithm, ZDT (Zitzler-Deb-Thiele's) functions with real decision variables [21] and DTLZ functions [22] with two and three objectives are adopted. The test functions have been widely used in the literature for performance evaluation of EMOAs. The number of design variables, $\mathrm{n_{dv}}$ is 10 for the ZDT4 function and 100 for all other test functions for clear performance comparisons among the tested EMOAs.

For evolutionary operators, as mentioned earlier in Sect. 2.5, we use the SBX recombination [14] with $\eta_\mathrm{c} = 15$ and $p_c = 1$, and the polynomial mutation operator [15] with $\eta_\mathrm{m} = 20$ and $p_m = 1/n_{dv}$. The population size and number of generations are set as 100 and 200, respectively. Therefore, the total number of function evaluations is 20,000.

There are many performance metrics available for measuring convergence and diversity of Pareto-front solutions by multi-objective optimization algorithms. The hypervolume indicator [24] measures the hypervolume in the objective space that is weakly dominated by an approximate set, hence the larger the indicator, the better the performance. The hypervolume indicator has recently become a popular performance measure because it is the only indicator known to be consistent with the Pareto-dominance. Here we use the *hyp_ind* program contained in the performance assessment tools of PISA [16,23]. *hyp_ind* calculates the difference in hypervolumes between a reference set and an approximate set under consideration, thus the smaller the difference, the better the performance. The reference set is defined here as a combined set of nondominated solutions by all the algorithms under comparison. Using



(a) ZDT1

(b) ZDT2

Fig. 5 Comparison of non-dominated solutions in a population of the 200th generation for the ZDT1 and ZDT2 functions between MOGA and MOGASc

the difference between the reference set and a solution set allows clear comparisons among different EMOAs.

In order to evaluate performances of the stochastic algorithms, each algorithm is run 30 times. For statistical tests, we are adopting the Kruskal Wallis test [25], which is a nonparametric test for differences among multiple independent samples without assuming a normal distribution of data. A confidence level was set as 95%; in other words, the null hypothesis is rejected when the p value is less than or equal to the significance level of 0.05. We use a program for the Kruskal Wallis test included in PISA [16,23].

4.1 Results for the ZDT functions

Figure 5 reveals the defect of the standard sharing approach in MOGA, where nondominated solutions in the combined $2N$ solutions of the final (200th) generation are presented for the ZDT1 and ZDT2 functions. A nonuniform distribution of the Pareto-front solutions by MOGA is evident, while on the other hand MOGASc gives a much more even distribution of solutions. Similar trends are also observed for other functions. Although actual performance evaluations are made for all the nondominated solutions kept in the external archive, the trend of the solution distributions in the last generation supports our arguments made in Sect. 3 about the potential defect of the standard sharing approach when combined with the best-N strategy.

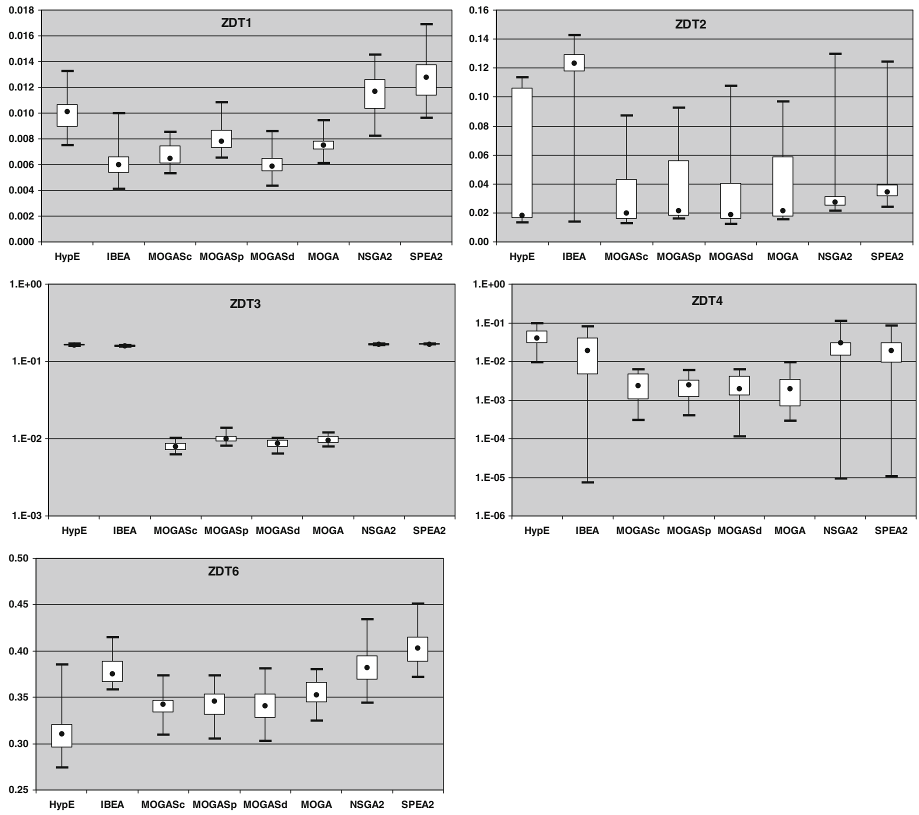


Fig. 6 Box plots of hypervolume indicator for the ZDT functions (smaller value is better). A dot in the box represents the median value

Table 1 The number of dominating algorithms for each algorithm in statistically significant way on various ZDT functions (smaller value is better)

	HypE	IBEA	MOGASc	MOGASp	MOGASd	MOGA	NSGA-II	SPEA2
ZDT1	5	0	2	4	0	3	6	7
ZDT2	0	7	0	1	0	0	3	5
ZDT3	5	4	0	2	1	2	6	7
ZDT4	7	4	0	0	0	0	5	5
ZDT6	0	5	1	1	1	4	5	7
Average	3.2	4.0	0.6	1.6	0.4	1.8	5.0	6.2

The box plots of the hypervolume indicator are depicted in Fig. 6 for the ZDT test functions, representing respectively the minimum, first quartile, third quartile and maximum values. A dot in the box indicates the second quartile or the median value of multiple runs.

Table 1 shows the number of algorithms that are better than each algorithm in a statistically significant way according to the Kruskal Wallis test. The ranks shown in Table 1 can also be used to compare two specific algorithms: an algorithm with a better (smaller) rank outperforms an algorithm with a poorer (larger) rank in a statistically significant way. Algorithms with the same rank do not outperform each other in terms of the statistical test.

For ZDT3 and ZDT4, MOGA and all the three variants of MOGAS give similar performances and outperform other EMOAs as shown in Fig. 6. For ZDT6, HypE performs the best, followed by the MOGAS algorithms. Overall, variants of MOGAS and MOGA are ranked from the 1st to 4th for the ZDT functions in terms of the averaged number of dominating algorithms in Table 1. The best algorithm for the ZDT functions is MOGASd. The second best algorithm is MOGASc. MOGASp is not superior to MOGASd and MOGASc in any test function. MOGA is inferior or similar to MOGASd and MOGASc for all the ZDT functions and superior or similar to MOGASp for all the ZDT functions except ZDT6. Indicator-based algorithms such as IBEA and HypE perform worse than MOGAS and MOGA, but better than NSGA-II and SPEA2.

4.2 Results for the DTLZ functions with two objectives

Box plots are depicted in Fig. 7 for the two-objective DTLZ functions. Table 2 shows the number of algorithms that are significantly better than each algorithm according to the Kruskal Wallis test.

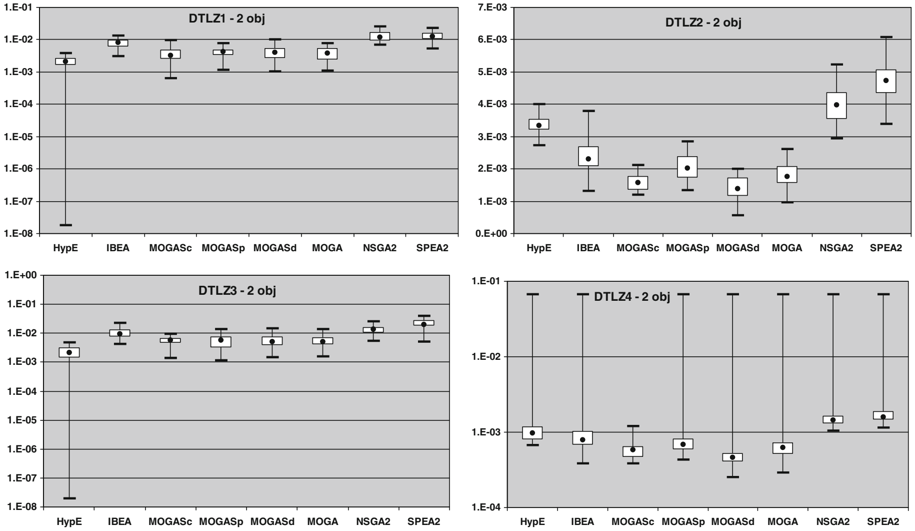


Fig. 7 Box plots of hypervolume indicator for the DTLZ functions with two objectives (smaller value is better)

Table 2 Ranking of algorithms: the number of dominating algorithms for each algorithm on the two-objective DTLZ functions (smaller value is better)

	HypE	IBEA	MOGASc	MOGASp	MOGASd	MOGA	NSGA-II	SPEA2
DTLZ1	0	5	1	1	1	1	6	6
DTLZ2	5	4	1	3	0	2	6	7
DTLZ3	0	5	1	1	1	1	6	7
DTLZ4	4	4	0	2	0	1	6	6
Average	2.3	4.5	0.8	1.8	0.5	1.3	6.0	6.5

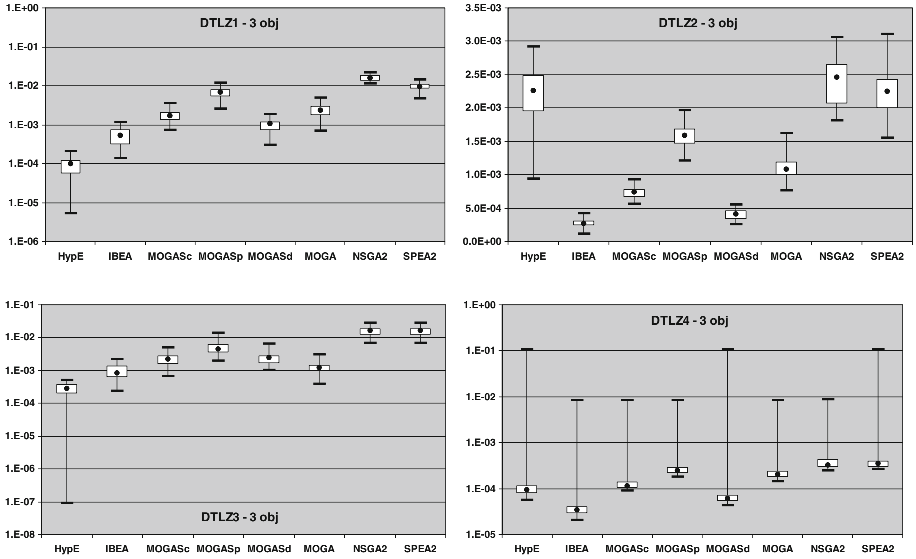


Fig. 8 Box plots of hypervolume indicator for the three-objective DTLZ functions (smaller value is better)

Table 3 Ranking of algorithms: number of dominating algorithms for each algorithm on the three-objective DTLZ functions (smaller value is better)

	HypE	IBEA	MOGASc	MOGASp	MOGASd	MOGA	NSGA-II	SPEA2
DTLZ1	0	1	3	5	2	4	7	6
DTLZ2	5	0	2	4	1	3	7	5
DTLZ3	0	1	3	5	3	2	6	6
DTLZ4	2	0	2	5	1	4	6	6
Average	1.8	0.5	2.5	4.8	1.8	3.3	6.5	5.8

As in the ZDT functions, the indicator-based algorithms such as IBEA and HypE perform better than the Pareto-based algorithms such as NSGA-II and SPEA2 in the two-objective DTLZ functions. HypE shows the best performance for multimodal problems (DTLZ1 and DTLZ3) but performs worse for unimodal problems (DTLZ2 and DTLZ4) than MOGA and MOGAS, which outperform other algorithms in terms of the average number of dominating algorithms, as shown in Table 2. The best algorithm is MOGASd, and the second best is MOGASc as was for the ZDT functions. MOGASc is not superior to MOGASd, and MOGASp is inferior or similar to MOGASd, MOGASc and MOGA in any two-objective DTLZ function.

4.3 Results for the DTLZ functions with three objectives

Box plots for the three-objective DTLZ test functions are depicted in Fig. 8. Table 3 shows the number of algorithms that are significantly better than each algorithm. Similar to the two-objective DTLZ functions, HypE is the best scheme again for the multimodal problems (DTLZ1 and DTLZ3) but not as good for the unimodal problems (DTLZ2 and DTLZ4). IBEA is the best algorithm for the unimodal problems and second best for the multimodal problems. Therefore, in overall, the best algorithm for the three-objective DTLZ functions is

IBEA. The current results are consistent with those reported in the literature in that the indicator-based algorithms have better performance than the Pareto-based algorithms for MOPs with more than two objective functions [5–8].

MOGASp again is inferior to MOGASd, MOGASc and MOGA in all the three-objective DTLZ functions.

MOGASd is found to be the best among the Pareto-based algorithms and shows very close performances to IBEA.

5 Concluding remarks

The fitness sharing method has been widely used in EMOAs for diversity preservation among solutions. In the standard sharing approach two close solutions that share a niche in the objective space are evenly degraded in their fitness to a shared fitness. When this standard sharing is combined with the best-N elite preserving strategy, a drawback of losing both solutions is discovered in the reduction process. To cope with this problem, we have suggested an improved approach, MOGAS, in which a new sharing function and sided fitness degradations are employed. Three variants of the new approach have been tested with regard to the preference in the fitness sharing to children (MOGASc), to parents (MOGASp), or to the ones closer to the ideal point in the objective space (MOGASd).

Standard multi-objective test functions, ZDT with two objectives and DTLZ with two and three objectives, were used to validate the effectiveness of the proposed method. Experimental results were compared to those of MOGA and other state-of-the-art EMOAs such as HypE, IBEA, NSGA-II and SPEA2. Some conclusions from the experiments can be summarized as follows:

- For all the test problems considered, MOGA and MOGAS show better performances than NSGA-II and SPEA2.
- Among MOGA and variants of MOGAS, MOGASd gives the best performance. MOGASc is superior or similar to MOGASp for all test functions considered. Hence, MOGASd and MOGASc are preferred over MOGASp for the test cases.
- MOGA is outperformed by MOGASd and MOGAScfor almost all test functions except the three-objective DTLZ3 function.
- For multimodal DTLZ functions with two or three objectives, the best algorithm among the considered ones was HypE, which, however, shows relatively poor performances for unimodal DTLZ problems.
- On the other hand, IBEA is showing relatively poor performances for test functions with two objectives, but is outstanding for three objective problems; the best for unimodal DTLZ functions and the second best for multimodal DTLZ functions.
- Relative performance behaviors among Pareto-based algorithms such as MOGA, MOGAS, NSGA-II and SPEA2 show little change for the number of objectives of the DTLZ functions compared to the indicator-based methods such as IBEA and HypE.

Through the experiments, the proposed fitness sharing method was shown to have significantly improved performances over the standard sharing method, especially when an effective preference is chosen, such as MOGASd or MOGASc. Also, the elite preserving MOGA was shown to still be very competitive to other state-of-the-art EMOAs.

The new sharing method would be enhanced further if a more effective preference could be found other than the three variants of MOGAS tested in this study. Our future work will also focus on applications of the proposed method to real world engineering problems.

References

1. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multi-objective optimization: formulation, discussion, and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416–423 (1993)
2. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI (PPSN_VI), pp. 849–858 (2000)
3. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithms. TIK report 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich, May (2001)
4. Deb, K., Mohan, M., Mishra, S.: A fast multi-objective evolutionary algorithms for finding well-spread pareto-optimal solutions, KanGAL Report 2003002, Indian Institute of Technology, Kanpur, India (2003)
5. Zitzler, E., Kunzli, S. et al.: Indicator-Based Selection in Multi-objective Search. In: Yao, X. (ed.) 8th Int'l Conference on Parallel Problem Solving from Nature (PPSN VIII), pp. 832–842. Springer, UK (2004)
6. Bader, J., Zitzler, E.: HypE: an algorithm for fast hypervolume-based many-objective optimization. TIK report 286, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, November (2008)
7. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multi-objective selection based on dominated hypervolume. Eur. J. Oper. Res. **181**(3), 1653–1669 (2007)
8. Wagner, T., Beume, N., Naujoks, B.: Pareto- Aggregation-, and Indicator-based Methods in Many-Objective Optimization, Evolutionary Multi-Criterion Optimization. Springer, Berlin (2007)
9. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 41–49 (1987)
10. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the pareto archive evolutionary strategy. Evol. Comput. **8**(2), 149–172 (2000)
11. CoelloCoello Carlos, A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. pp. 109 Springer, Berlin (2007)
12. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. pp. 206–207. Wiley, New York (2001)
13. Baker, J.E.: Reducing bias and inefficiency in the selection algorithm. In: Grefenstette, J.J. (ed.) Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, pp. 14–21 (1987)
14. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**(2), 115–148 (1995)
15. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. Comput. Sci. Inform. **26**(4), 30–45 (1996)
16. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA—a platform and programming language independent interface for search algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) Evolutionary Multi-Criterion Optimization, pp. 494–508. Springer, Berlin (2003)
17. Coello, C.A.C., Pulido, G.T., Montes, E.M.: Current and Future Research Trends in Evolutionary Multi-objective Optimization, Information Processing with Evolutionary Algorithms: from Industrial Applications to Academic Speculations. pp. 213–231. Springer, London (2005)
18. Obayashi, S., Sasaki, D., Takeguchi, Y., Hirose, N.: Multi-objective evolutionary computation for supersonic wing-shape optimization. IEEE Trans. Evol. Comput. **4**(2), 182–187 (2000)
19. Kanazaki, M., Obayashi, S., Nakahashi, K.: Design optimization of exhaust manifold for a car engine using MOGA. Presented at EUROGEN2001, Athens, September. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou K., Fogarty T. (eds.) Evolutionary Methods for Design, Optimisation and Control, c CIMNE, Barcelona (2002)
20. Lopex-ibanez, M., Knowles, J., Laumanns, M. et al.: On Sequential Online Archving of Objective Vectors. In: Takahashi, R. (ed.) Evolutionary Multi-criterion Optimization (EMO 2011) volume 6576 of Lecture Notes in Computer Science, pp. 46–60. Springer, Heidelberg (2011)
21. Zitzler, E., Deb, K., Thiele, L.: Comparison of multi-objective evolutionary algorithms: empirical results. Evol. Comput. J. **8**(2), 125–148 (2000)
22. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. In: Abraham, A., Jain, R., Goldberg, R. (eds.) Evolutionary Multi-objective Optimization: Theoretical Advances and Applications, Chapter 6, pp. 105–145. Springer, Berlin (2005)

23. Knowles, J., Thiele, L. Zitzler, E.: A tutorial on the performance assessment of stochastic multi-objective optimizers. TIK report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, February (2006)
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multi-objective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**, 117–132 (2003)
25. Conover, W.J.: Practical Nonparametric Statistics, 3rd edn. Wiley, New York (1999)