

Weighted Preferences in Evolutionary Multi-Objective Optimization

Tobias Friedrich¹ and Trent Kroeger² and Frank Neumann²

¹ Max-Planck-Institut für Informatik, Saarbrücken, Germany

² School of Computer Science, University of Adelaide, Adelaide, Australia

Abstract. Evolutionary algorithms have been widely used to tackle multi-objective optimization problems. Incorporating preference information into the search of evolutionary algorithms for multi-objective optimization is of great importance as it allows one to focus on interesting regions in the objective space. Zitzler et al. have shown how to use a weight distribution function on the objective space to incorporate preference information into hypervolume-based algorithms. We show that this weighted information can easily be used in other popular EMO algorithms as well. Our results for NSGA-II and SPEA2 show that this yields similar results to the hypervolume approach and requires less computational effort.

1 Introduction

Evolutionary algorithms are very powerful problem solvers especially when dealing with multi-objective optimization problems [5, 6]. Many successful methods have been developed in evolutionary multi-objective optimization (EMO) during recent years. Popular evolutionary algorithms for multi-objective optimization are (among many others) NSGA-II [8] and SPEA2 [17], as well as hypervolume-based approaches such as SMS-EMOA [3] and MO-CMA-ES [11, 12].

Recently, there has been significant interest in strategies for introducing user preferences into EMO methods. The goal is to give specific regions of the objective space a higher priority. Consequently, more solutions should be computed and maintained in the population for highly preferred regions of the objective space. For NSGA-II a reference point based approach has been proposed by [7]. They give the example that in the problem of maximizing throughput and minimizing latency, a decision maker may have a clue that throughput should be about 99.9%. Several authors [1, 9, 14] also used reference points to guide multi-objective particle swarm algorithms. Wickramasinghe and Li [15] use preferred areas. Auger et al. [2] present an approach of sampling the weighted hypervolume to incorporate user-defined preferences into the search for problems with many objectives. Thiele et al. [13] extended this such that at each iteration, a decision maker is asked to give preference information in terms of his reference point. Furthermore, Hu et al. [10] changed the crowding distance assignment in NSGA-II in order to achieve a non even spread of the points along the Pareto front.

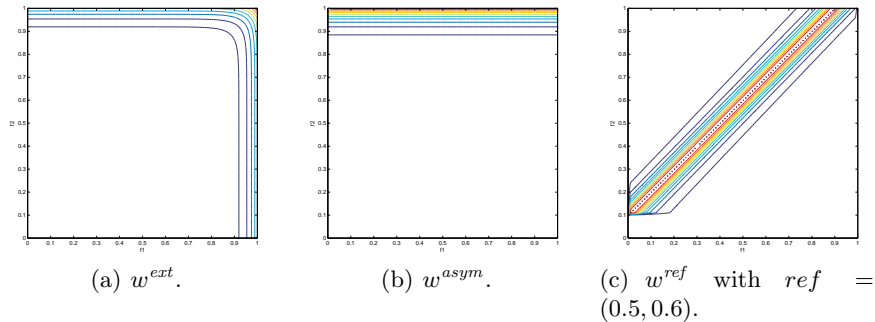


Fig. 1. Contour plots of the weight distribution functions

We consider the case where there is a weight function on the objective space. This preference information can be used by the weighted hypervolume indicator instead of the standard hypervolume indicator. Zitzler et al. [18] show that this weight integration is very well suited for incorporating user preferences. Furthermore, they compare their results to the ones obtained by NSGA-II and SPEA2 and show that these two algorithms do not perform well as they do not take into account the preference information.

In this paper, we present a neat and simple approach to use the preference information given by weightings on the objective space in classical algorithms such as NSGA-II and SPEA2. Up to now, this weight information has only been used by hypervolume-based approaches that have the drawback of needing a runtime exponential in the number of dimensions [4]. We present very simple approaches to incorporate the weight information on the objective space into a wide range of EMO algorithms. We exemplify this by using NSGA-II and SPEA2 and show that this leads to results similar to the ones of the weighted hypervolume indicator presented in [18]. Furthermore, our algorithms are as efficient as the original implementation of NSGA-II and SPEA2 and do not have to deal with the expensive computations that hypervolume-based algorithms have to face [4].

The outline is as follows. In Section 2, we introduce some basic concepts of multi-objective optimization and the weight functions used. Section 3 shows how to incorporate weight information into NSGA-II and SPEA2. We report our experimental results in Section 4 and finish with some concluding remarks.

2 Preliminaries

A multi-objective optimization problem is given by a vector-valued objective function

$$f = (f_1, \dots, f_d): S \rightarrow \mathbb{R}^d$$

on a search space S . W.l.o.g. we assume that each function f_i , $1 \leq i \leq d$, should be minimized. We first define a partial order on the objective space. An

objective vector $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ weakly dominates an objective vector $y = (y_1, \dots, y_d) \in \mathbb{R}^d$ ($x \preceq y$) if it is not worse in any objective, i. e., $x \preceq y :\Leftrightarrow x_i \leq y_i$ for $1 \leq i \leq d$.

Let $f(A)$ be the set of objective vectors of the search points in A , i. e., $f(A) = \{f(a) : a \in A\}$. Then, we denote by $\text{Min}(f(A), \preceq)$ the set of minimal objective vectors in $f(A)$ with respect to the partial order \preceq on $f(A)$. The goal in multi-objective optimization is to compute a set X^* with $f(X^*) = \text{Min}(f(S), \preceq)$, where S is the considered search space. $f(X^*)$ is called the Pareto front of the given problem.

Often the size of the Pareto front is large, i. e., exponential with respect to the given input or even infinite in the case of continuous functions. In this case, it is not possible to compute the whole set of minimal elements of $f(S)$ efficiently and $f(X^*)$ should be a smaller subset of them.

So far, there has not been any preference between incomparable solutions. Having to cope with a large Pareto front, we have to decide between incomparable solutions. Basically, all successful evolutionary algorithms have certain diversity mechanisms to deal with this issue. Our goal is to investigate user preference in evolutionary multi-objective optimization. These user preferences give additional information for the search process and distinguish between sets of incomparable solutions.

We assume that we have access to a weight function $w : \mathbb{R}^d \mapsto \mathbb{R}$ which describes the preferences of the decision maker. In principle, w can be an arbitrary function that gives preferences to certain regions of the objective space. We will use the following weight distribution functions on the objective space which have been introduced and investigated by [18] in the context of hypervolume-based algorithms:

- Uniform weight: $w^{uni}(x) = 1$
- Sum of two exponential functions in the direction of the axes:

$$w^{ext}(x) = (e^{20 \cdot x_1} + e^{20 \cdot x_2}) / (2 \cdot e^{20})$$

- Exponential function in the f_2 -direction:

$$w^{asym}(x) = e^{20 \cdot x_2} / e^{20}$$

- Weighted depending on a reference point $ref = (a, b)$:

$$w^{ref}(x) = \begin{cases} c + \frac{(2 - ((2(x_1 - a))^2 + (2(x_2 - b))^2))}{(0.001 + (2(x_1 - a) - 2(x_2 - b))^2)} & \text{if } |x_1 - a| < 0.5 \wedge |x_2 - b| < 0.5 \\ c & \text{otherwise} \end{cases}$$

Note, that the uniform weight does not imply any preferences on the objective space. For NSGA-II and SPEA2 this will imply that we are just running the original versions of these algorithms. The weight distribution for the other three functions are illustrated in Figure 1 for the objective space $[0, 1]^2$ and we will show in the following sections how to generalize NSGA-II and SPEA2 such that they can make use of this preference information.

Algorithm 1: Crowding assignment for weighted NSGA-II

```

1  $\ell \leftarrow |X|$ ;
2 foreach  $x^j \in X$  do
3    $d(x^j) \leftarrow 0$ ;
4 for  $i \leftarrow 1$  to  $d$  do
5   Sort the solutions in  $P$  such that  $f_i(x^1) \leq f_i(x^2) \leq \dots \leq f_i(x^\ell)$  holds;
6    $d(x^1) \leftarrow d(x^\ell) \leftarrow \infty$ ;
7   for  $j \leftarrow 2$  to  $\ell - 1$  do
8      $d(x^j) \leftarrow d(x^j) + \frac{f_i(x^{j+1}) - f_i(x^{j-1})}{f_j^{\max} - f_j^{\min}}$ ;
9 foreach  $x^j \in P$  do
10   $d(x^j) \leftarrow w(x^j) \cdot d(x^j)$ ;

```

3 Algorithms

In this section, we show how to integrate the user preferences given by weightings on the objective space into NSGA-II [8] and SPEA2 [17]. Both algorithms are based on the Pareto dominance relation and use diversity mechanisms to decide between incomparable solutions. We transfer the diversity mechanisms for incomparable solutions to the weighted case and adjust them such that they can make use of the weight information provided on the objective space.

3.1 Weighted NSGA-II

NSGA-II is a very popular evolutionary multi-objective algorithm. It is based on dominance ranking which ensures that non-dominated solutions are preferred over dominated ones. Furthermore, the algorithm has a diversity mechanism which distinguishes between incomparable solutions by a crowding distance measure. This measure prefers solutions of less crowded regions in the objective space.

We will keep the dominance ranking as in the original algorithm. To incorporate the user preferences, we will change the crowding distance assignment according to the weight information on the objective space. Let X be a set of incomparable solutions then the assignment of a crowding distance, taking into account the weighting on the objective space, is given in Algorithm 1.

As the original NSGA-II, it iterates over all objectives. For each objective f_i , the solutions are sorted in increasing order and the distance of a solution is changed according to its neighboring points for that objective. Solutions that are maximal (or minimal) with respect to one objective obtain an infinite distance which gives strong preference to the extreme points of the Pareto front. Our weighted crowding distance assignment differs from the original crowding distance assignment by the last for-loop. In this loop the crowding distance of each solution is multiplied by the weight that its objective vector has according to the weight distribution on the objective space. Note, that this does not

change the crowding distance assignment of the extreme points as they have already obtained an infinite distance. However, it changes the assignment and the preferences for the other points and gives preferences based on the weighting of the objective space.

3.2 Weighted SPEA2

SPEA2 is another very popular approach. It sorts individuals of a population based on a fitness assignment strategy that incorporates both a coarse-grain evaluation of Pareto dominance that results in an integer raw fitness value and a fine-grained evaluation of density that allows the algorithm to distinguish between solutions with the same raw fitness. The sum of the density and raw fitness values yields the overall fitness of a solution, which is used within both the mating selection and environmental selection functions of the algorithm. The density $D(i)$ of a solution i is calculated as follows:

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

Where σ_i^k is the distance within the objective space from the solution i to its k -th nearest neighbour in the population. Solutions in less crowded regions of the objective space will be assigned lower density values and will be preferred when compared to solutions in more crowded regions of the same Pareto front. In this way, SPEA2 maximises the diversity of solutions within the population. Note that the density function is constructed such that $D(i) \leq 0.5$ and as such, the density cannot affect Pareto dominance relationships between solutions, which have a raw fitness value of integer type.

Our weighted density measure incorporates information from a weight distribution function as follows:

$$D(i) = \frac{1}{(w_i \cdot \sigma_i^k) + 2}$$

Where w_i is the value of the weight distribution function at the point in the objective space corresponding to solution i . Solutions corresponding to higher values of w_i will have a lower density $D(i)$ and will be favoured by the SPEA2 selection functions. Conversely, solutions corresponding to lower values of w_i will be less favoured by the SPEA2 selection functions. It is important to note that for any weight distribution function $w: \mathbb{R}^d \mapsto \mathbb{R}_+$, it still holds that $D(i) \leq 0.5$ and so, Pareto compliance is maintained.

For each iteration of the SPEA2 main loop, the environmental selection function involves copying non-dominated individuals from the archive and population at the previous iteration into a new archive. If the nondominated front fits exactly into the archive then the environmental selection step is completed. If there are not enough non-dominated individuals to fill the archive, the remaining places in the archive are filled with dominated individuals according to fitness. If there are too many non-dominated solutions to fit into the archive, then a

truncation procedure is invoked to iteratively remove individuals until the non-dominated solutions fit within the archive. At each iteration of the truncation procedure, a solution i is chosen for removal that has the minimum distance to another individual. If there are several individuals with minimum distance then the second and, if necessary subsequent, smallest distances are considered to break the tie. Our weighted version of SPEA2 incorporates user preferences within the truncation procedure by multiplying the calculated distance from a solution i to its k -th nearest neighbor by w_i to yield a weighted distance. The result of this modification is that solutions that are situated in highly-weighted regions of the objective space will have a relatively high weighted distance and so will be less likely to be removed by the truncation procedure.

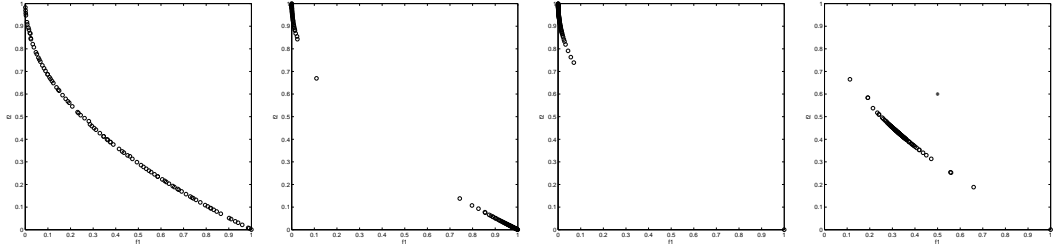
4 Experimental Results

In this section, we report on our experimental results for the weight integration into NSGA-II and SPEA2. We use the same setting as [18] for the weighted hypervolume indicator and examine the classical benchmark functions ZDT1, ZDT3, and ZDT6 [16] and the weight distribution functions defined in Section 2.

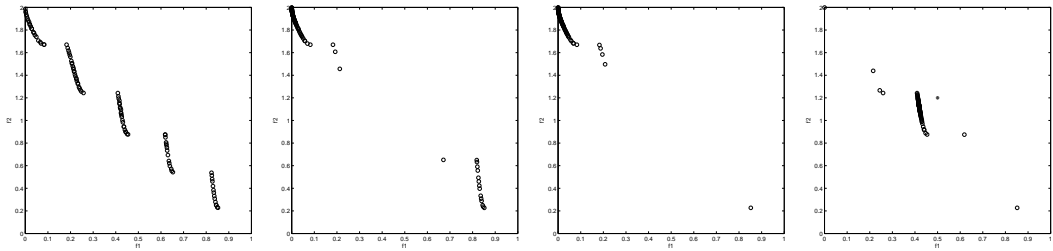
We now examine how the three weight distribution functions defined above influence the search process of NSGA-II and SPEA2 for the three test problems. The functions ZDT1, ZDT3, and ZDT6, are optimized by NSGA-II runs with population size 100 for 25000 generations and SPEA2 runs with population and archive sizes of 100 for 25000 generations. For both algorithms, a crossover probability setting of 0.90 is used and mutation probability is set to 0.03. Each of the test problems was constructed with 30 decision variables.

Figures 2 and 3 show the computed Pareto front approximations for NSGA-II and SPEA2 after 25000 generations for the three ZDT functions and the three weight distribution functions w^{ext} , w^{asym} and w^{ref} . The reference point for w^{ref} is chosen as $ref = (0.5, 0.6)$ for ZDT1 and ZDT6 and as $ref = (0.5, 1.2)$ for ZDT3 which is the same as in [18]. The difference in reference point position is due to the fact that the ZDT3 function has a larger range of values in the f_2 -direction. The computed Pareto front approximation for a uniform weighting scheme w^{uni} is also shown to allow comparisons to the results of the original NSGA-II and SPEA2 variants.

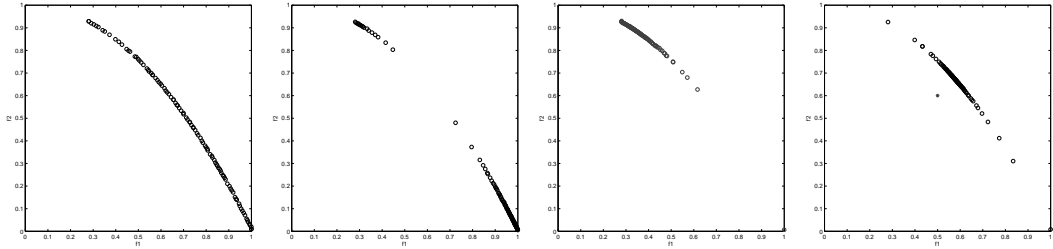
Charts of the experiments show that for test functions ZDT1 and ZDT6, the weighted versions of NSGA-II and SPEA2 were highly successful in directing solutions towards regions of the objective space in accordance with all three of the weight distribution functions trialled. For the weighting scheme w^{ext} , both algorithms yielded a set of solutions that were concentrated near the boundary regions of the Pareto front. Similarly, the use of the weighting scheme w^{asym} resulted in a set of solutions concentrated near the boundary of the Pareto front in the f_2 -direction. When the w^{ref} weight distribution function was used, both algorithms yielded results that were concentrated in a the region of the Pareto front that was closest to the specified reference point.



(a) NSGA-II on ZDT1 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .

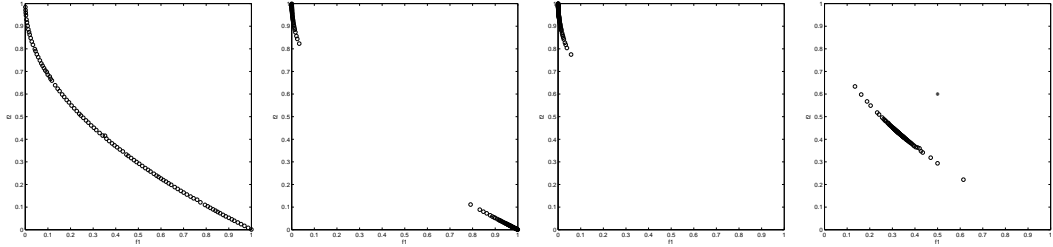


(b) NSGA-II on ZDT3 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .

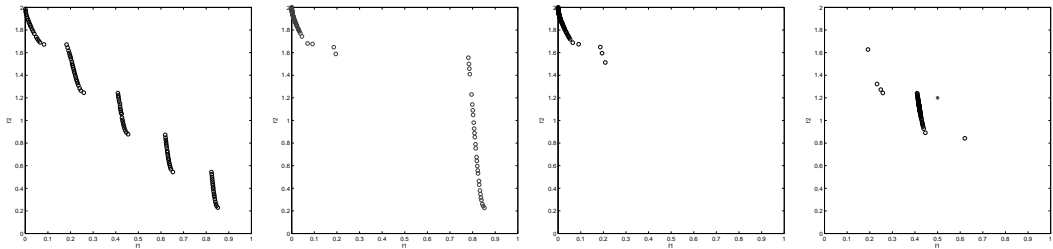


(c) NSGA-II on ZDT6 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .

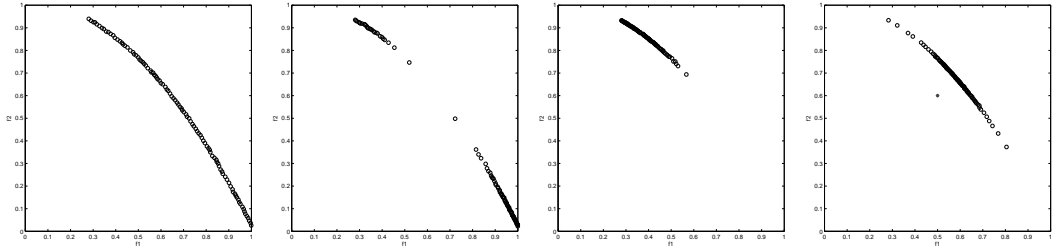
Fig. 2. Experimental results for NSGA-II.



(a) SPEA2 on ZDT1 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .



(b) SPEA2 on ZDT3 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .



(c) SPEA2 on ZDT6 with weight functions w^{uni} , w^{ext} , w^{asym} , w^{ref} .

Fig. 3. Experimental results for SPEA2.

Similar behaviour was observed for the ZDT3 test function when the weighted NSGA-II and SPEA2 algorithms were executed using weighting schemes w^{asym} and w^{ref} . However, both algorithms produced less definitive results when using the ZDT3 test function and the w^{ext} weighting scheme. In particular, it appears that for the weighted SPEA2 algorithm, application of the w^{ext} weight distribution function led to a poor approximation of the true Pareto front in some regions.

Despite this, the experimental results clearly demonstrate that the proposed weighted versions of NSGA-II and SPEA2 can be used successfully to guide solutions towards areas of the objective space according to an arbitrary weight distribution function. Importantly, these approaches are implemented in such a way that they can be used to incorporate user preferences without compromising Pareto compliance of the algorithms. Furthermore, weights are introduced to the diversity measures of each algorithm in such a way as to modify, but not completely destroy its diversity characteristics.

5 Conclusions

The integration of user preference into EMO methods is an important research topic as it allows the user of an EMO algorithm to focus on interesting regions of the objective space. Different models for the integration of user preference have been proposed. Incorporating user preferences by weight information on the objective space has been shown to work very well for hypervolume algorithms [18]. We have presented a simple and very effective alternative way to use these user preferences in other state-of-the-art approaches such as NSGA-II and SPEA2. Our experimental results show that the weight integration into these algorithms performs very well, produces similar results as the weighted hypervolume indicator, and requires less computational effort than the hypervolume approach.

Bibliography

- [1] R. Allmendinger, X. Li, and J. Branke. Reference point-based particle swarm optimization using a steady-state approach. In *Proc. Simulated Evolution and Learning (SEAL '08)*, volume 5361 of *Lecture Notes in Computer Science*, pages 200–209, 2008.
- [2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Proc. 11th Annual conference on Genetic and evolutionary computation*, pages 555–562, 2009.
- [3] N. Beume, B. Naujoks, and M. T. M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181:1653–1669, 2007.
- [4] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry: Theory and Applications*, 43:601–610, 2010.

- [5] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.
- [6] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.
- [7] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proc. 8th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO '06)*, pages 635–642, 2006.
- [8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.
- [9] S.-l. Ho, S. Yang, and G. Ni. Incorporating a priori preferences in a vector pso algorithm to find arbitrary fractions of the pareto front of multiobjective design problems. *IEEE Trans. Magnetics*, 44:1038–1041, 2008.
- [10] Q. Hu, L. Xu, and E. D. Goodman. Non-even spread NSGA-II and its application to conflicting multi-objective compatible control. In *Proc. Genetic and Evolutionary Computation Conference Summit (GEC '09)*, pages 223–230, 2009.
- [11] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15:1–28, 2007.
- [12] T. Suttorp, N. Hansen, and C. Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75:167–197, 2009.
- [13] L. Thiele, K. Miettinen, P. J. Korhonen, and J. M. Luque. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436, 2009.
- [14] U. K. Wickramasinghe and X. Li. Integrating user preferences with particle swarms for multi-objective optimization. In *Proc. 10th Annual conference on Genetic and evolutionary computation*, pages 745–752, 2008.
- [15] U. K. Wickramasinghe and X. Li. Using a distance metric to guide pso algorithms for many-objective optimization. In *Proc. 11th Annual conference on Genetic and evolutionary computation*, pages 667–674, 2009.
- [16] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2): 173–195, 2000.
- [17] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proc. Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100, 2002.
- [18] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Proc. Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO '07)*, volume 4403 of *LNCS*, pages 862–876. Springer, 2007.