

MOEA/D - New Resource Allocation Functions Diversity Based*

XXX[†]
XXX
XXX, XXX
XXX@XXX.XXX

XXX[‡]
XXX
XXX, XXX
XXX@XXX.XXX

XXX[§]
XXX
XXX, XXX
XXX@XXX.XXX

ABSTRACT

Multi-objective Evolutionary Algorithm based on Decomposition, MOEA/D, decomposes multi-objective problems into single-objective subproblems. All subproblems are uniformly treated and there is no priority among them. Each subproblem is related to an area of the theoretical Pareto Front. It is expected that different areas would be more difficult to approximate than others, leading to an unbalanced exploration of the search space. To balance exploration, "Resource Allocation" techniques that prioritize certain subproblems were proposed. Here we investigate how priority functions relate to MOEA/D in terms of performance. We consider four different methods as priority functions: diversity on the objective space, diversity in the decision space, a priority function with random values and the relative improvement, from MOEA/D-DRA. We conducted an experimental analysis on the DTLZ and UF benchmark problems and on the lunar landing real-world problem and compared the famous MOEA/D-DE variant with each four priority functions and without any. The results indicate XXX.

KEYWORDS

ACM proceedings, L^AT_EX, text tagging

ACM Reference Format:

XXX, XXX, and XXX. 2019. MOEA/D - New Resource Allocation Functions Diversity Based. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Multi-objective Optimization Problems (MOPs) are problems with multiple, conflicting objectives. This composition is characterized by a set of conflicting objective functions resulting in a set of optimal compromise solutions.

$$\text{minimize } f(x) = (f_1(x), \dots, f_m(x)), x \in \mathbb{R}^D, \quad (1)$$

where m is the number of objective functions and \mathbb{R}^m is the objective function space. $x \in \mathbb{R}^D = \{x_1, x_2, \dots, x_D\}$ is a D-dimensional

vector which represents a candidate solution with D variables, $f : \mathbb{R}^D \rightarrow \mathbb{R}^m$ is a vector of objective functions.

These objectives often conflict with each other, as there is no $X \in \mathbb{R}^D$ that minimizes all the objectives at the same time. Consequently, the goal of the MOP optimization algorithm is to find the approximate set of solutions that balance the different objectives in an optimal way.

This balance is defined by the concept of "pareto dominance". Given two solutions vectors u, v in \mathbb{R}^D , u Pareto-dominates v , we say that denoted by $f(u) < f(v)$, if and only if $f_k(u) \leq f_k(v), \forall k \in \{1, \dots, m\}$ and $f(u) \neq f(v)$. Likewise, a solution $x \in \mathbb{R}^D$ is considered Pareto-Optimal if there exists no other solution $y \in \mathbb{R}^D$ such that $f(y) > f(x)$, i.e., if x is non-dominated in the feasible decision space. A non-dominated solution exists if no other solution provides a better trade-off in all objectives. As in the study by Zitzler et al. [19] study, weak dominance ($A \geq B$) means that any solution in set B is weakly dominated by a solution in set A. However, this does not rule out equality, because $A \geq A$ for all approximation sets A.

Consequently, the set of all Pareto-Optimal solutions is known as the Pareto-Optimal Set (PS), while the image of this set is referred to as the Pareto-optimal Front (PF).

$$PS = \{x \in \mathbb{R}^D \mid \nexists y \in \mathbb{R}^D : f(y) > f(x)\}, \quad (2)$$

$$PF = \{f(x) \mid x \in PS\}. \quad (3)$$

Multi-objective evolutionary algorithms (MOEAs) are one of the most widely used groups of algorithms for finding approximations to the PF of a MOP. They are characterized by their ability to find good approximations to PF in a single run [16]. In recent years, there has been an increasing interest in studying MOEAs and with a primary concern of improving their general performance.

In this study, we are interested in analyzing the Multi-objective Evolutionary Algorithm based on Decomposition framework, MOEA/D [13]. It represents a class of population-based meta-heuristics for solving Multi Objective Problems. In this framework, each individual has a specific weight vector which is used to decompose the original multi-objective problem into simpler, single-objective subproblems by means of scalarizations. Each subproblem is then evaluated and its utility value is calculated by an aggregation function given the related weight vector.

In the original MOEA/D, each solution of a subproblem have the same amount of computational resource (number of iterations). Each subproblem relates to a region of the PF. Since all of them are uniformly treated, it is expected that different regions of the would be more difficult to find approximations than other areas leading to an unbalanced exploration of the search space.

Although researchers have not studied this problem in much detail, there have been some works that have discussed this matter.

*Produces the permission block, and copyright information

[†]XXX

[‡]XXX

[§]XXX

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '19, July 13–17, 2019, Prague, Czech Republic

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

One way to address this problem is to allocate different number of evaluations to the subproblems based on some priority function. In a few recent works, a priority function (also called utility function) is used to prioritize resources given to subproblems that contribute more to the algorithm's search. In the works of Zhang et al. [14] and Zhou et al. [17] a priority function was proposed aiming to prioritize solutions based on a historical convergence information during different generations. Another approach was implemented in Kang et al. [7], where the priority function was based on the presence of a solution from the main population on a secondary population.

The aim of this study is to explore the relationship between subproblems and priority functions in the context of MOEA/D. Here, we propose a two new functions based to diversity for defining priorities. The first addresses diversity in the decision space using the L2-norm. The second considers the integration of an online diversity metric based on a geometrical perspective [5] as a direct way to define the priority function.

2 RELATED WORKS

2.1 Priority functions

We define priority functions as one way of establishing preferences [3],[6] among solutions for the allocation of resources. Priority functions monitor the diversity of the solutions of an algorithm and may be used to decide how to distribute the computation resources among subproblems, guiding the search behavior of the algorithm. Also, they may be used as one way of deciding computational resources distributions among subproblems by guiding the distribution over generations [1].

Only a few studies have been concerned with resource allocation. We highlight: MOEA/D-DRA [14]; MOEA/D-GRA [17]; and MOEA/D-AMS [4]. Most studies provide little justification for the definition of priority functions and little explanation of how these functions help the MOEA/D search.

According to Zhou and Zhang [17], MOEA/D-GRA may be seen as an generalization of MOEA/D-DRA and MOEA/D-AMS. The reason is that all of these algorithm use a very similar priority function. The priority function values, $u = \{u_1, u_2, \dots, u_N\}$ for every subproblem $i = 1, \dots, N$, is defined as

$$u_i = \frac{\text{old function value} - \text{new function value}}{\text{old function value}}. \quad (4)$$

This equation is based the assumption that if a subproblem has been improved over the last ΔT generations (*old function value*), it should have a high probability of being improved over the next few generations.

In contrast, the priority function used in MOEA/D-CRA [7], is based on if an individual from the main population A is selected to compose the archive population B MOEA/D-M2M [9], a two population MOEA/D.

Together these studies indicate that it is worth monitoring the algorithm behavior and guiding its search, but it is unclear if the priority functions proposed are the most appropriated ones.

Consequently, in this work, we define a priority function based on a diversity. Two approaches are considerer. One metric with a geometrical interpretation of convergence and diversity. This

new function is then integrated to the MOEA/D framework. The resource allocation procedure is similar as the one in MOEA/D-DRA. The benefit of using MOEA/D-DRA is that it is a good representative of a the class of variants of MOEA/D with resource allocation, since it is widely used.

2.2 Diversity Metric

One way to measure diversity is to use metrics that evaluate MOPs solvers. The hypervolume indicator (HV) [18] and the Inverted Generational Distance (IGD) [15] are frequently used as metrics to evaluate such solvers. However they include information about both quality of the solutions and diversity in a single metric.

Among the metrics that only measure diversity, there are mainly two groups. The offline group, that calculate the diversity after the execution of the algorithm, while online group, that calculate the diversity during the execution of the algorithm. We are interested in measuring diversity during the execution of the algorithm, therefore we briefly introduce some studies that are part of the online group.

The online group includes: sigma method [10] (PF lies in the positive objective space); entropy of the solutions by using Parzen window density estimation-[11] (sensitive to kernel width); and maximum relative diversity loss, MRDL, [5] (expensive $O(N^2)$, with N being the size of the parent population).

In this work we chose to focus on using diversity as the desired characteristic that the priority function should focus on.

In this work, we chose the maximum relative diversity loss as an online diversity assessment to measure the diversity loss caused by any individual in the population and then guide the resource allocation at each generation.

3 MOEA/D WITH PRIORITY FUNCTIONS

Algorithm 1 MOEA/D with priority functions

- 1: Initialize the weight vectors λ_i , the neighborhood B_i , the priority value u_i every subproblem $i = 1, \dots, N$.
 - 2: **while** Termination criteria **do**
 - 3: **for** 1 to N **do**
 - 4: **if** $\text{rand}() < u_i$ **then**
 - 5: Generate an offspring y for subproblem i .
 - 6: Update the population by y .
 - 7: Evaluate and after ΔT generations, keep updating \mathbf{u} by a priority function.
-

Algorithm 1 describes MOEA/D with priority functions. Except for line 4, in which a subproblem may not be part of the group that is going to be iterated and for line 7, in which the priority function is calculated, the whole procedure is similar to the MOEA/D-DE [14]. Likewise, all reproduction procedures and parameters are the same as in MOEA/D-DE [8]. It is important to highlight that the neighborhood is only calculated in the initialization period.

The decomposition method used is the Simple-Lattice Design (SLD), the scalar aggregation function used is Weighted Sum (WS), the update strategy used is the Restricted update strategy and finally we performed a simple linear scaling of the objectives to the interval $[0, 1]$.

Algorithm 2 "spectral" norm

```

1: Input: NEW new incumbent solutions; OLD, previous iteration
   incumbent solutions; N, the population size.
2: for i=1 to N do
3:   u[i] = calculate the spectral norm of NEW[i] - OLD[i]
4: u = scale (u) // between 0 and 1
5: return u

```

We understand that priority functions provides an important property. It provides ways of designing MOEA/D variants that could focus on a desired characteristics, such as diversity, performance contribution, convergence to a specific region of the Pareto Front or others. This is possible because different methods can be used as priority functions to create the vector u in algorithm 1.

In this work we chose to focus on studying four different characteristics: diversity on the objective space, diversity in the decision space, no information (random values) and the relative improvement, from MOEA/D-DRA as our priority functions. Next we give a brief explanation of why we chose to consider these our methods and we describe the details of how to calculate them.

Independently of the method used to calculate the priority function, we initialize the value of the vector $u = 1$, as in MOEA/D-DRA. As in DRA and GRA we have a learning period, ΔT generations (*old function value*). Here $\Delta T = 20$. A sensitivity analyzes should be performed for deciding suitable initial values for u and for ΔT .

3.1 Priority Functions - Diversity Group

We believe in the idea that diversity is a critical issues of a search process in any multi-objective algorithm. Therefore, we propose to use priority functions that address lack of diversity aiming to make solutions better spread among each other. These two priority functions focus on different aspects of the diversity: solutions better spread along the Pareto Set (diversity on the decision space) and solutions better spread along the Pareto Front (diversity on the objective space).

3.1.1 Diversity on decision space. The priority function proposed that considers diversity on the objective space is based on the "spectral" norm (or 2-norm), which is the which is the largest singular-value (SVD) of the difference between the offspring solution minus the parent solution. Algorithm 2 gives the details on implementation.

By using diversity on decision space as the priority function incumbent solutions that are similar are given more resources. Therefore more effort is used focusing on generating solutions that are different in the decision space, leading to a higher exploration of the decision space.

3.1.2 Diversity on objective space. The priority function proposed that considers diversity on the objective space is based on the Maximum Relative Diversity Loss, MRDL. Algorithm 3 gives the details on implementation.

Prior to scalarizing it between 0 and 1 to fit the algorithm 1 we calculate MRDL for every individual of the population. The following function describes how to calculate the priority function given the MRDL, $\Gamma^{p \rightarrow c}$.

Algorithm 3 MRDL

```

1: Input: old MDRL (default value is 0); C, objective function val-
   ues from the incumbent solutions; P, objective function values
   from the previous iteration incumbent solutions; N, the popu-
   lation size.
2: for i=1 to |C| do
3:   find  $h \in |P|$  where  $(P_h \geq C_i)$  and  $\|P_h - C_i\|$  is minimal.
4:    $d.conv_y = C_i - P_h$ .
5:   for j=1 to N do
6:      $p = P_j - P_h$ 
7:      $c = c_j - c_i$ 
8:      $proj_{d.conv_y} * p' = \frac{sum(conv_y * p')}{crossprod(p')} * p'$ 
9:      $proj_{d.conv_y} * c' = \frac{sum(conv_y * c')}{crossprod(c')} * c'$ 
10:     $RDL = \frac{\|p' - proj_{d.conv_y} p'\|}{\|c' - proj_{d.conv_y} c'\|}$ 
    MDRL[i] = maximum RDL
11: u = scale (MDRL - old MDRL) // between 0 and 1
12: return u, MDRL

```

$$u_i = \Gamma_i^{p \rightarrow c} - \Gamma_j^{p \rightarrow c}, \text{ with } i = 1, \dots, N. \quad (5)$$

MRDL is an online diversity metric that estimates the diversity loss of a solution to the whole population [5]. Following, some features worth highlighting are shown. First, it is an useful metric since if a new offspring generated is identical to any offspring solution in the convergence archive, the metric value will be infinite. Second, high values of this metric indicates the existence of similar offspring solution in the convergence archive or the offspring solution is close to the line of estimated convergence direction. Finally, it may be used as an adaptive technique to adjust parameters, as applied in its original study [5].

The idea of the MRDL is to use the space movement (convergence directions) of a solution on the objective space towards the PF. The further an objective vector of a solution is from the convergence direction, the more it contributes for the diversity of the approximated PF.

To compute the MRDL, we need to estimate k convergence directions (shown later) at every generation. Then we need to compute the Relative Diversity Loss (RDL) for each of these k convergence directions.

To calculate RDL of a solution to the whole population, the following equation is used. It considers every incumbent solution related to a subproblem i , from the whole population.

$$\Gamma_i^{p \rightarrow c} = \max_{j=1, \dots, k} \Gamma_{d.conv_y}^{p \rightarrow c}. \quad (6)$$

RDL is a diversity measurement quantity that indicates the amount of diversity loss of an individual solution between two consecutive generations. High values of RDL imply a reduction of the solution spread.

This reduction is given by a division between the shortest distance of a parent, p , and offspring, c , to the line of convergence direction.

Algorithm 4 Random

```

1: Input: N, the population size.
2: for i=1 to N do
3:   u[i] = random value between 0 and 1
   return u

```

$$\Gamma_{d.conv_y}^{p \rightarrow c} = \frac{\|p' - proj_{d.conv_y} p'\|}{\|c' - proj_{d.conv_y} c'\|} \quad (7)$$

The numerator in 7 is the closest distance between the parent solution (p) to the convergence direction ($c_r - p_r$). While, the denominator in 7 is the closest distance between the offspring solution (c) to the convergence direction ($c_r - p_r$).

p' and c' are given by:

$$\begin{aligned} p' &= p - p_r, \\ c' &= c - c_s, \end{aligned} \quad (8)$$

with p_r and c_s being the parent, and offspring objective vectors used to calculate the convergence direction in equation 11. Index s is equal to index j used to calculate $conv_y$. The same principle is valid for index r .

The vector projection between two vectors is defined as next.

$$proj_{d.conv_y} p' = \frac{d.conv_y \cdot p'}{|p'|^2} p'. \quad (9)$$

While the norm of $p' - proj_{d.conv_y} p'$ is calculated as follows.

$$\|p' - proj_{d.conv_y} p'\| = \text{sqrt}(\text{crossprod}(proj_{d.conv_y} p')). \quad (10)$$

The norm of $c' - proj_{d.conv_y} c'$ is calculated similarly.

To estimate the convergence direction, $d.conv_y$, we need to have an offspring, c_j , that dominates at least one parent. Select a parent, p_h , solution that is closest to this offspring in the objective space.

For every weakly dominated parent, one convergence direction is calculated as in the next equation.

$$d.conv_y = c_j - p_h \quad (11)$$

Index j (for indexing offsprings, c_j) is selected from the set D_c . Index h is explained later.

$$D_c = \{d | \exists c_d < p_k, k \in 1, \dots, N, d \in [1, \dots, |C|]\} \quad (12)$$

N is the parent population size, $|C|$ is the size of the offspring population C . In equation 12, the offspring c_d must weakly dominate at least one parent solution.

Index h (for indexing parents, p_h) comes from the following two equations.

$$h = \underset{k \in D_p}{\text{argmin}} \|p_k - c_j\| \quad (13)$$

$$D_p = \{k | \exists c_j < p_k, k \in 1, \dots, N\} \quad (14)$$

D_p in equation 14 denotes the index set of parent solutions which are weakly dominated by c_j (j index comes from equation 12).

Algorithm 5 Relative Improvement

```

1: Input: C, objective function values from the incumbent solutions; P, objective function values from the  $\Delta T$  previous iteration incumbent solutions; N, the population size.
2: for i=1 to N do
3:   u[i] = (C - P)/C
   return u / (max(u) + 1.0x10-50)

```

3.2 Priority Function - Random

Does it help? When we consider more sophisticated priority functions, are we making any improvement?

Using no information. We select solutions at random. if it is best

bigger population might be worse, since if a subpopulation better results were found - to much effort for nothing smaller population might be better, since if a subpopulation better results were found - effort resource better spent subpopulation from a bigger population is better than smaller population with the subpopulation size?

Algorithm 4 gives the details on implementation.

3.3 Priority Function - Relative Improvement

Commonly used. Using (higher) improvement over generations from DRA and GRA. 4 Algorithm 5 gives the details on implementation.

4 EXPERIMENTAL DESIGN

The question that we want to answer is how the proposed algorithm, MOEA/D-RAD, performs when compared with MOEA/D-DE and MOEA/D-DRA. This comparison is conducted by applying these three algorithms to the first 9 Black-Box Optimization Bi-Objective Benchmark (bbob-biobj) test functions [12].

For the three algorithms, most of the parameters are as follows. The population size $N = 350$, the update size $nr = 2$, the neighborhood size $T = 20$, and the neighborhood search probability $\delta_p = 0.9$. The DE mutation operator value is $\phi = 0.5$. The Polynomial mutation operator values are $\eta_m = 20$ and $p_m = 0.03333333$ [2].

The number of executions is 21. At each execution, the number of functions evaluations is 70000.

We perform statistical tests on the hypervolume (HV) metric values and Inverted Generational Distance (IGD) for measuring the quality of a set of obtained non-dominated solutions found by the algorithms on the both DTLZ and UF benchmark problems. Before calculating the HV value, the objective function was scaled between 0 and 1. The reference point for the HV calculation was set to (1, 1). For the real-world lunar landing problem while for the lunar, we only perform statistical tests on the hypervolume (HV) metric values and the reference point for the HV calculation was set to (1, 0, 1). Higher values of the HV indicate better approximations while lower values of the IGD indicate better approximations.

In order to verify any statistical difference in the average performance given the different algorithms, the Pairwise Wilcoxon Rank Sum Tests was used, with confidence interval $\alpha = 0.05$ and with the Hommel adjustment method.

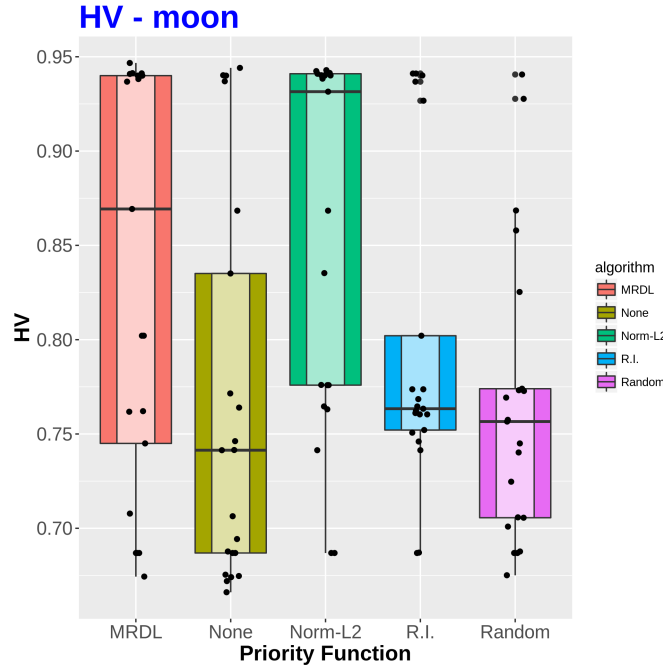


Figure 1: HV values of the last iteration on the Lunar Landing Problem

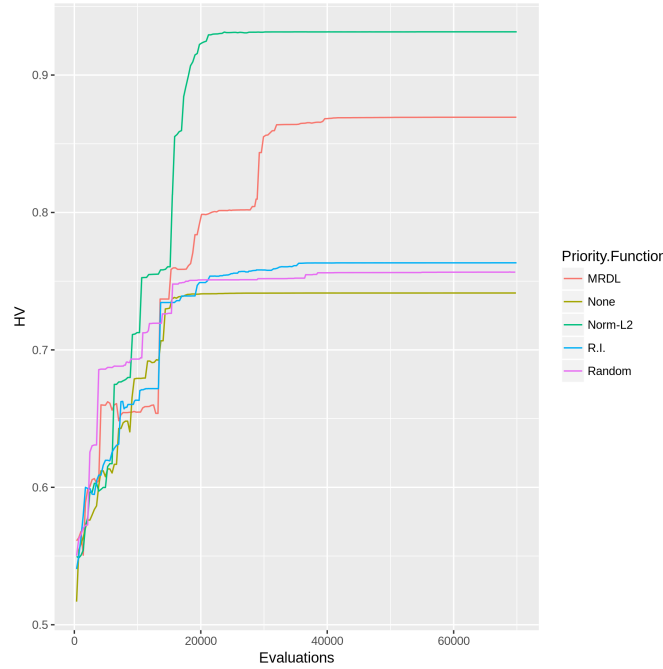


Figure 2: Evolution of the HV on the Lunar Landing Problem

5 RESULTS

6 CONCLUSION

REFERENCES

- [1] Xinye Cai, Yexing Li, Zhun Fan, and Qingfu Zhang. 2015. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 508–523.
- [2] Felipe Campelo, Lucas S Batista, and Claus Aranha. 2018. The MOEA/D Package-A Component-Based Framework for Multiobjective Evolutionary Algorithms Based on Decomposition. *arXiv preprint arXiv:1807.06731* (2018).
- [3] V Chankong and YY Haimes. 1983. *Multiobjective Decision Making: Theory and Methodology* North Holland. New York (1983).
- [4] Tsung-Che Chiang and Yung-Pin Lai. 2011. MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 1473–1480.
- [5] Sen Bong Gee, Kay Chen Tan, Vui Ann Shim, and Nikhil R Pal. 2015. Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 542–559.
- [6] Sven Ove Hansson. 2005. *Decision Theory A Brief Introduction*. Department of Philosophy and the History of Technology, Royal Institute of Technology (KTH) Stockholm. (2005).
- [7] Qi Kang, Xinyao Song, MengChu Zhou, and Li Li. 2018. A Collaborative Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018).
- [8] Hui Li and Qingfu Zhang. 2009. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on evolutionary computation* 13, 2 (2009), 284–302.
- [9] Hai-Lin Liu, Fangqing Gu, and Qingfu Zhang. 2014. Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems. *IEEE Trans. Evolutionary Computation* 18, 3 (2014), 450–455.
- [10] Sanaz Mostaghim and Jürgen Teich. 2003. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 26–33.
- [11] Kay Chen Tan, Chi Keong Goh, AA Mamun, and EZ Ei. 2008. An evolutionary artificial immune system for multi-objective optimization. *European Journal of Operational Research* 187, 2 (2008), 371–392.
- [12] Tea Tusar, Dimo Brockhoff, Nikolaus Hansen, and Anne Auger. 2016. COCO: the bi-objective black box optimization benchmarking (bbob-biobj) test suite. *arXiv preprint arXiv:1604.00359* (2016).
- [13] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.
- [14] Qingfu Zhang, Wudong Liu, and Hui Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 203–208.
- [15] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. 2008. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* 12, 1 (2008), 41–63.
- [16] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagarathnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1 (2011), 32–49.
- [17] Aimin Zhou and Qingfu Zhang. 2016. Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 52–64.
- [18] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*. Springer, 292–301.
- [19] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.