Using Diversity as a Priority Function for Resource Allocation on MOEA/D

Anonymous Author(s)

ABSTRACT

The key characteristic of the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) is that the multi-objective problem is decomposed into multiple single-objective subproblems. In standard MOEA/D, all subproblems receive the same computational effort. However, as each subproblem relates to different areas of the objective space, it is expected that some subproblems are more difficult than others. Resource Allocation techniques allocates computational effort proportional to each subproblem's difficulty. This difficulty is estimated by a priority function. Using Resource Allocation, MOEA/D could spend less effort on easier subproblems and more on harders ones, improving efficiency. In this paper, we investigate different priority functions. We propose that using diversity as the priority criteria results in better allocation of computational effort. We propose two new priority functions: objective space diversity and decision space diversity. We compare the proposed diversity based priority with previous approaches on the DTLZ and UF benchmarks, as well as on a real world problem about selecting a landing site for lunar exploration. The proposed priority functions based on diversity performed better in terms of HV, IGD and percentage of non-dominated solutions. Decision space diversity was better on the benchmarks, while objective space diversity excelled on the Lunar Landing problem.

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:

Anonymous Author(s). 2019. Using Diversity as a Priority Function for Resource Allocation on MOEA/D. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Multi-objective Optimization Problems (MOP) are maximization (or minimization) problems characterized by multiple, conflicting objective functions. It arises in real world applications that require a compromise among multiple objectives. Thet of optimal trade off solutions in the decision space is the *Pareto Set*, and the image of this set in the objective space is the *Pareto Front*. Finding a good approximation of the Pareto Front is a hard problem for which multiple Evolutionary Algorithms have been proposed.

The Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [13] is an effective algorithm for solving MOPs. The key idea of the MOEA/D is that the multi-objective optimization problem is decomposed into a set of single objectives subproblems All subproblems are then solved in parallel.

In the original MOEA/D, all subproblems are treated uniformly, in the sense that all of them receive the same computational effort. However, it has been observed that some subproblems are harder than others, and take more effort to converge to an optimal solution [17]. Because of this, *Resource Allocation* approaches have been proposed to allocate different amount of computational effort to different subproblems, based on an estimation of the relative difficulty of each subproblem [8, 14, 17]. The most popular approach for estimating subproblem difficulty is the Relative Improvement, which calculates how much a subproblem has improved in recent iterations.

Here, we propose a new approach for estimating difficulty and calculating priority in Resource Allocation for MOEA/D. Our approach uses the idea of *diversity* in decision space and in objective space to calculate the priority of solutions. Our motivation for this choice is that the quality of a MOP solution set is often evaluated by the diversity in the objective space. If we assign higher priority for regions with lower diversity, we are encouraging the algorithm to spend more computational effort in regions that are not yet well explored.

In this paper, we define a priority function based on diversity on the objective space using the MRDL, proposed by Gee [6]. The MRDL is an online diversity metric based on a geometrical perspective and indicates the loss of diversity related to a solution to the whole population. We also define a priority function based on diversity on the decision space using the 2-Norm. It compares solutions by measuring the norm of the difference of these solutions. We understand that these priority functions are able to monitor diversity during the execution of the algorithm guiding the search behavior of the algorithm.

We compare the new approach with the Relative Improvement or not using priority functions at all. The results show that focusing on diversity leads to better results on the metrics Hypervolume (HV) and Inverted Generational Distance (IGD) and also lead to a higher percentage of non-dominated solutions. The diversity on the decision space shows better performance on the benchmark function, generally being better than using the relative improvement. It came to our surprise that the diversity on the objective space barely improve the results of not using priority functionsm in the artificial benchmarks. On the other hand, using diversity on the objective space as a priority function excels in the Lunar Landing problem.

2 BACKGROUND

2.1 Priority functions

We define priority functions as one way of establishing preferences [3],[7] among solutions for the allocation of resources. 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]; MOEA/D-AMS [4]. 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, $u = \{u_1, u_2, ..., u_N\}$ for every subproblem $i = 1, ..., N$, is defined as

$$u_i = \frac{\text{old function value} - \text{new function value}}{\text{old function value}}. \tag{1}$$

This equation, the relative improvement (R.I.), 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.

The priority function used EAG-MOEA/D [1] and MOEA/D-CRA [8] differ from the ones that we mentioned previously. In their case, the framework keeps two populations: one working population, and one external archive. The priority function is based on the contribution to the external archive from each subproblem in the search process.

Together these studies indicate that it is worth monitoring the algorithm behavior and guiding its search, but it is unclear how the priority functions influence into the results since in all the impact of using priority functions was not isolated. That is, in all of the work previously mentioned incremented MOEA/D with priority functions and some extra. For example, in MOEA/D-DRA a 10-tournament selection [14] was used while in MOEA/D-GRA a new replacement strategy was also consider [17]. MOEA/D-AMS proposes an adaptive mating selection mechanism as dynamically adjusts the mating pools of individuals [4]. Finally, both EAG-MOEA/D [1] and MOEA/D-CRA [8] use archive population.

The new priority functions are defined focus in on improving the population diversity. 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 PS (diversity on the decision space) and solutions better spread along the PF (diversity on the objective space).

That said and based on recent the success of addressing the problem of resource allocation using priority functions, we aim to isolate priority functions to analyze the real impact of using them in the MOEA/D framework. In this study, we compare these two new priority functions with the R.I., equation 1, to further understand how priority functions influence the performance of MOEA/D framework.

2.2 Diversity Metric

Over the last two decades, some works in diversity metrics have been successfully applied in different tasks on evolutionary computation. 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, we highlight the ones that calculate the diversity during the execution of the algorithm. We have sigma method [11] (PF lies in the positive objective space); entropy of the solutions by using Parzen window density estimation-[12] (sensitive to kernel width); and maximum relative diversity loss, MRDL, [6] (expensive $O(N^2)$, N being the size of the parent population).

MRDL is an online diversity metric that estimates the diversity loss of a solution to the whole population [6]. High values indicate the existence of similar solutions or that the offspring solution is close to the convergence direction. The further an objective vector of a solution is from the convergence direction, the more it contributes for the diversity of the approximated the Pareto front. The MRDL is the maximum value for Relative Diversity Loss (RDL) of each solution.

In this work, we consider the MRDL as the first priority function since it targets diversity on the objective space. To deal with diversity on the decision space we consider the similarity of decision vectors of consecutive iteractions given by the 2-Norm. The 2-Norm is defined similarly as the R.I., since in both a difference between vectors values from distinct iteractions. However there are two main differences between R.I. and 2-Norm priority function. The first is that while the R.I. considers the function values of solutions the 2-Norm considers its decision values. The second difference is that in R.I. the next step is scalarizing the values between 0 and 1 and in the 2-Norm priority function, the 2-Norm value of that difference is computed prior to scalarizing.

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 $i = 1, ..., N$.
2: **while** *Termination criteria* **do**
3:     **for** 1 to N **do**
4:         **if** *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 $\boldsymbol{u}$ by a priority function.

In this study we use the basic algorithm framework 1with priority functions of MOEA/D-GRA. In contrast to MOEA/D-GRA we only consider the basic algorithm and no other variant. The benefit of using MOEA/D-GRA is that it has a simple code structure and represents well the class of variants of MOEA/D with resource allocation. Therefore, our new functions based on diversity are then easily integrated to the MOEA/D framework.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]. All reproduction procedures and parameters are the same as in MOEA/D-DE [9]. We highlight that the neighborhood is only calculated in the initialization period.

Algorithm 2 2-Norm

```
Input: NEW new incumbent solutions; OLD, previous iteration
  incumbent solutions; N, the population size.
for i=1 to N do
    u[i] = ||New[i] - OLD[i]||
u = scale (u) // between 0 and 1
return u
```

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 we performed a simple linear scaling of the objectives to [0, 1].

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 PF 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 $\Delta T = 20$ for artificial benchmarks, as in MOEA/D-GRA [17], while for the real-world problems, we chose $\Delta T = 2$, by trial and error. A sensitivity analyzes should be performed for deciding suitable initial values for u and for ΔT.

It should also be noticed that once less than 3 or less subproblems would be improved in a given iteraction i, we reset the priority vector $u = 1$ and all subproblems will be chosen for offspring reproduction at the that i iteraction.

3.1 Priority Function - Norm of the difference of current solutions and its parents

The priority function proposed that considers diversity on the objective space is based on the 2-Norm,

$$2\ \mathrm{Norm}_i = ||\text{current solution} - \text{parent solution}||. \tag{2}$$

The idea of the 2-Norm as priority function is that by using diversity on decision space as the priority function more resources are given to incumbent solutions that are similar. Hence more effort is used focusing on modifying solutions that are close in the decision space, leading to a higher exploration of the decision space.

3.2 Priority Function - MRDL

The diversity on objective space as a priority function is based on the Maximum Relative Diversity Loss, MRDL [6]. The idea using MRDL as a priority function is that by measuring diversity on the objective space, more resources are given to incumbent solutions that have similar objective function values between two consecutive iteractions leading to a higher exploration of the objective space. Algorithm 3 gives the details on implementation.

Algorithm 3 MRDL

Input: old MRDL (initial value is 0); C, objective function values from the incumbent solutions; P, objective function values from the previous iteraction incumbent solutions; N, the population size.
for i=1 to $|C|$ **do**
  find $h \in |P|$ where ($P_h \geq C_i$) and $||P_h - C_i||$ is minimal.
  $d.conv_y = C_i - P_h$.
  for j=1 to N **do**
    $p = P_j - P_h$
    $c = c_j - c_i$
    $proj_{d.conv_y} * p' = \dfrac{sum(conv_y \cdot p')}{(p' \times p')} * p'$
    $proj_{d.conv_y} * c' = \dfrac{sum(conv_y \cdot c')}{(c' \times c')} * c'$
    $RDL = \dfrac{||p' - proj_{d.conv_y} p'||}{||c' - proj_{d.conv_y} c'||}$
  MDRL[i] = maximum RDL
u = 1 - scale (MRDL - old MRDL) // between 0 and 1
return u, MRDL

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

$$\Gamma_i^{p \to c} = \max_{i=1,\dots,k} \Gamma_{d.conv_y}^{p \to c}. \tag{3}$$

At every iteraction and for each incumbent solution, we need k convergence directions (shown later) and we compute the Relative Diversity Loss (RDL) for each of these k convergence directions. The maximum value among these k convergence directions is chosen as the MRDL for that incumbent solution.

$$\Gamma_{d.conv_y}^{p \to c} = \frac{||p' - proj_{d.conv_y} p'||}{||c' - proj_{d.conv_y} c'||}. \tag{4}$$

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

p' and c' are given by:

$$\begin{aligned} p' &= p - p_r, \\ c' &= c - c_s, \end{aligned} \tag{5}$$

with p_r and c_s being the parent, and offspring objective vectors used to calculate the convergence direction in equation 7. 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.

Algorithm 4 Relative Improvement

```
Input: C, objective function values from the incumbent solutions; P, objective function values from the ΔT previous iteraction incumbent solutions; N, the population size.
for i=1 to N do
    u[i] = (C[i] - P[i])/C[i]
u / (max(u) + 1.0x10^-50)
return u
```

$$proj_{d.conv_y}p\prime = \frac{d.conv_y \cdot p\prime}{(p \times p)^2}p\prime,$$

$$proj_{d.conv_y}c\prime = \frac{d.conv_y \cdot c\prime}{(c \times c)^2}c\prime. \quad (6)$$

While the norm of $p\prime - proj_{d.conv_y}p\prime$ is calculated as follows using the 2-Norm of this difference. The same procedure is done for $c\prime - proj_{d.conv_y}c\prime$.

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. 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.

$$d.conv_y = c_j - p_h \quad (7)$$

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

$$D_c = \{d | \exists c_d \prec p_k, k \in 1, ..., N, d \in [1, ..., |C|]\}z \quad (8)$$

N is the parent population size, $|C|$ is the size of the offspring population C. In equation 8, 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.

$$h = \underset{k \in D_p}{argmin} ||p_k - c_j|| \quad (9)$$

$$D_p = \{k | \exists c_j \prec p_k, k \in 1, ..., N\} \quad (10)$$

D_p in equation 10 denotes the index set of parent solutions which are weakly dominated by c_j (j index comes from equation 8).

3.3 Priority Function - Relative Improvement

Here we give a brief description of the Relative Improvement (R.I.), the priority function used in MOEA/D-DRA, MOEA/D-GRA and many others. This priority function aims to measure subproblem hardness and then it helps allocating more resource to subproblems that have improved more over the next few generations. Algorithm 4 gives the details on implementation of the equation 1. R.I. was first introduced in the context of the unconstrained MOEA competition in the CEC 2009 [14], being the winner of that competition [16]. This competition also introduced the UF benchmark functions.

Algorithm 5 Random

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

3.4 Priority Function - Random

Finally, we describe the last priority function studied in this work. The random priority function is used a base for comparison. Given no information besides the size of the population, we define the vector of priority u at random. Algorithm 5 gives the details on implementation.

4 EXPERIMENTAL DESIGN

The question that we want to answer is how MOEA/D-DE performs when combined with: no priority function (none) and the priority functions 2-norm, MRDL, relative improvement and random. To answer that question we introduce the 4 priority function into MOEA/D-DE and apply these variants to two artificial benchmark problems and the Lunar Landing real-world problem and we compared the performance against the results of the MOEA/D-DE. The first benchmark used is the DTLZ functions [5] with 100 dimensions and k = dimensions - number of objectives +1, where the number of objectives is 2. The second benchmark is the UF functions [16], with 100 dimensions.

The Lunar Landing problem is a real-world problem that simulates the selection of landing sites for lunar landers [10]. In lunar exploration it is critical to suitable landing site of the rovers. Good landing sites ensure enough sunshine providing enough energy power supply for the rovers while in a region with scientifically interesting materials with little difficulties to the exploration. In this minimization problem, the two decision variables are the longitude and latitude with three objectives: the number of continuous shade days, the total communication time (in reality, this is a maximization problem that was inverted with the goal of consistency), and the inclination angles. This problem is considered to be a severe constrained problem, due to the presence of two craters. In values, the two constraints are defined as continuous shade days being < 0.05 while inclination angles being < 0.3

For every combination, 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 set to 21. For each execution, the number of functions evaluations is 70000. Since the Lunar Landing is a severe constrained one, we chose the population size $N = 5050$ and the number of functions evaluations is 60000.

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, DTLZ and UF benchmark problem. Before calculating the HV value, the objective function was scaled between 0 and 1. The reference point for the HV calculation was set to $(1, 1)$. However, for the real-world Lunar Landing problem, we only perform statistical tests on the hypervolume (HV) metric values using reference point as $(1, 0, 1)$. 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. For reproducibility the code is made available at XXX.

5 RESULTS

Figure 1 shows box-plot that exemplify the results found in the UF benchmark and DTLZ functions as well as the Lunar Landing problem in terms of the HV values while Figure 2 does the same but in terms of the IGD values, but only to the artificial benchmarks. In them we can see that for the artificial benchmarks MRDL as a priority function is slightly better than MOEA/D-DE, considering both HV and IGD values. Norm, R.I. and Random perform the best. On the other hand, MRDL performed the best in the Lunar problem, followed by R.I., Norm and lastly by MOEA/D-DE.

Figure 4 illustrates the PF approximation for the DTLZ4 found by all priority functions and without it, as understood by the results given by the HV and IGD, both Norm and R.I. approximate well the PF. On the other hand it is clear that MRDL did not lead to a better approximation of the PF when compared to the one of MOEA/D-DE. More Figures regarding all experiments and results can be found at the supplementary materials.

5.1 HV Results

Table 1 shows the results for every priority function measured by HV. First we discuss the results for the UF functions, then the results for the DTLZ functions and lastly the results for the Lunar Problem. Norm as priority function had few good results, with the best median in UF3 and UF9. In UF3, there is no statistical significant difference for the R.I. while in UF9, there is no statistical significant difference for the Random. The R.I. had the higher median in five functions. Only in UF8 results there exists a statistical significant difference to the other priority functions. In UF1, UF5 and UF7 statistical significant difference to the results of the Random. Surprisingly, the Random priority function got the higher median in the UF2 (with no statistical difference to R.I.), UF4(with no statistical difference to R.I. or Norm) and UF6 (with no statistical difference to R.I.). Its median is always higher that the one from MRDL.

Now, we discuss to the results on the DTLZ functions. Norm as priority function lead to several good results in median with the best median in the DTLZ2 (no statistical difference to Random), DTLZ5 (no statistical difference to Random or R.I) , DTLZ6 and DTLZ7 (no statistical difference to R.I.) cases. Only in the DTLZ6 the difference in the results had statistical significance. R.I. performed as the best algorithm in terms of median of HV values only in the DTLZ4 (without any significant difference to Random or Norm). Reinforcing our surprise, the Random priority function got the higher median in the DTLZ1 (with no statistical difference to Norm or R.I.), DTLZ2 (with no statistical difference to R.I.), DTLZ3 and DTLZ5 functions(both without statistical difference to Norm or R.I.), with its median being always higher that the one from MRDL.

On the Lunar Problem, the best priority function is the MRDL, the only one with statistical difference to MOEA/D-DE. However, the results of this priority function have no statistical difference to the other priority functions.

5.2 IGD Results

We again start with the UF results and move after to the DTLZ results. In the Table 2 we verify that the Norm had the best median in 7 functions, being statistically different to all others in the UF3, UF5, UF7 and UF9 functions. In the UF2 and UF4 the results of the Norm are not statistically different from the R.I. or the Random. In UF8 the results of the Norm not statistically different from the Random. The R.I. had the highest median in the UF1, UF2, UF4, UF6 and UF10 functions. As in the case of UF2, the results of UF1 and UF4 are not statistically different from the R.I. or the Random. In UF6 the results are not statistically different from the Random while on the UF10, there is no statistical different to the results of the Norm.

Moving to the DTLZ results, we verify that the Norm had the best median in 3 functions, DTLZ2, DTLZ5 (with no statistical difference to Random or R.I.) and DTLZ67 (with no statistical difference to Random). In the DTLZ2 the results had statistical difference. The R.I. perform the best in the DTLZ3 (with no statistical difference to Norm or Random), DTLZ5 and DTLZ7(both with no statistical difference to Norm). Only in the DTLZ1 Random had the highest values(with no statistical difference to Norm or R.I.).

5.3 Rate Non-dominated Solutions

The results on the Table 3 indicate that the Norm leads to the best rate of non-dominated solutions in the artificial benchmarks, being always the fastest, mean time of the median of every function: 1.17; standard deviation (sd): 2.46. The MRDL priority function improved a little the rate of non-dominated solutions, at the cost of a longer execution time (mean time of the median of every function: 37.53; sd: 1.03), which may be alleviated by more costly problems, such as real-world problems. The same behavior is found in the R.I.(mean time of the median of every function: 39; sd: 5.14) and Random results (mean time of the median of every function: 31.47; sd: 0.41).

On the Lunar Landing problem, MRDL found less feasible solutions than any variant, although it exceed the rate of non-dominated solutions of the MOEA/D-DE, also it was the fastest, mean time of 4.95 and sd of 0.21. MOEA/D-DE had mean time of 6.67 and sd of 1.98. Norm had the highest rate of feasible solutions, but had the second best rate of non-dominated solutions, surpassed by R.I. Both priority functions were the slowest, Norm: mean time of 6.81 with sd of 0.40; R.I.: mean time of 6.95 and sd of 1.02.

5.4 Resource Allocation

Figure 3 illustrates the amount resource allocated by Norm, R.I. and MRDL to every subproblem on UF3 and DTLZ4 problems. We exclude the visuals from MOEA/D-DE since it give the same amount of resource to every problem (200) and of the Random, since it is completely noisy. It is clear from this Figure 3 that, during the execution of the algorithm, the resource allocate to each subproblem is different. This behavior is different given priority functions, illustrating that every priority function allocates different amount of resource given their characteristics. It is also important to highlight that each priority function influences the search differently given

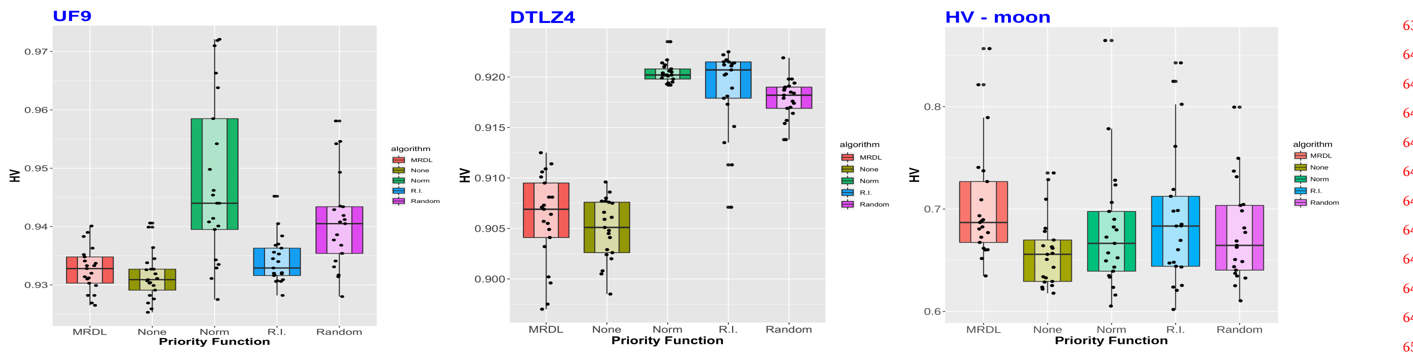

Figure 1: Box plot of HV values on UF9, DTLZ4 and Lunar Landing.None is the MOEA/D-DE with no priority function.

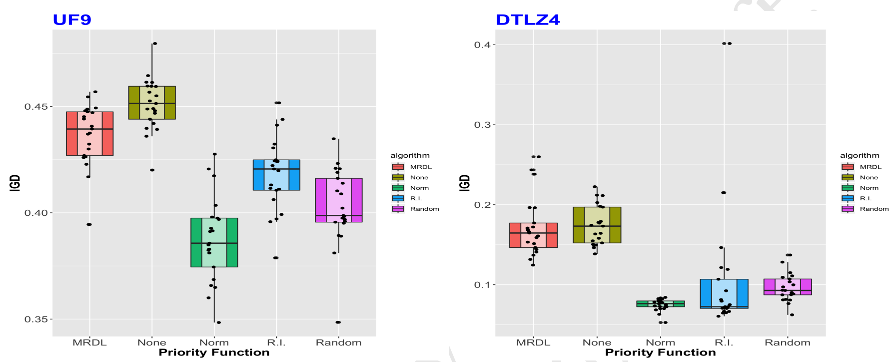

Figure 2: Box plot of IGD values on UF9 and DTLZ4. None is the MOEA/D-DE with no priority function.

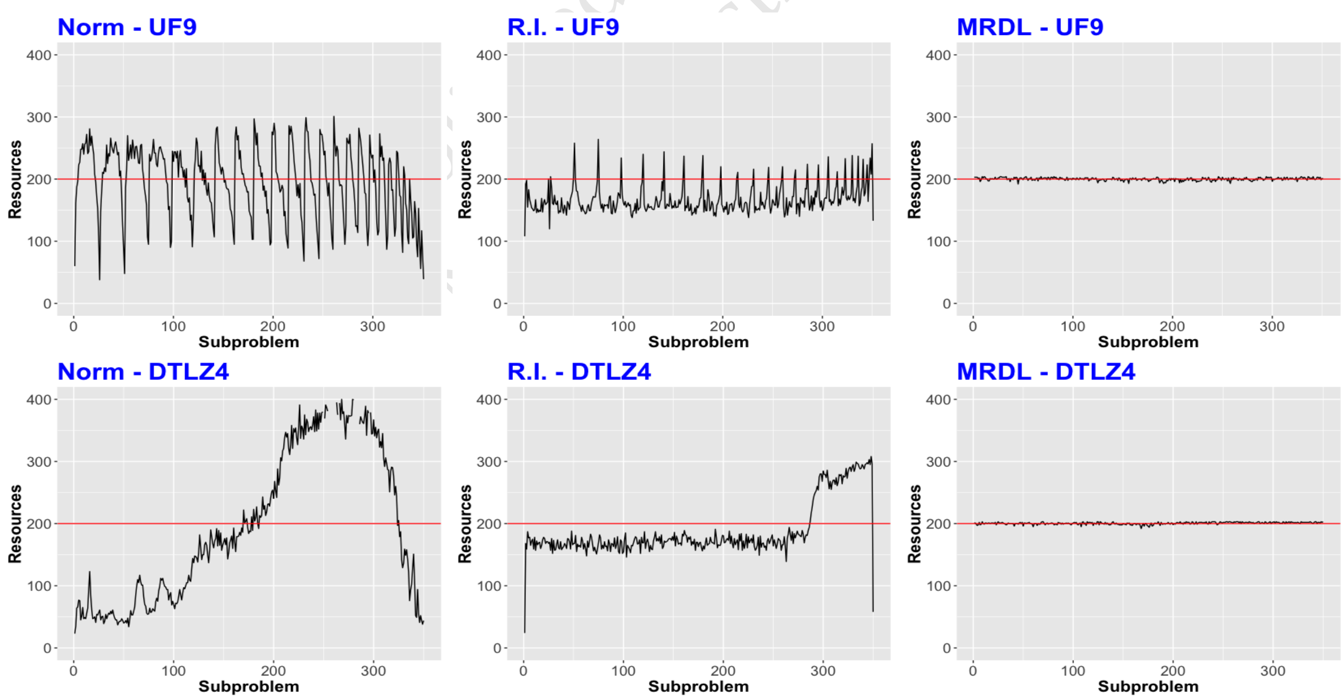

Figure 3: Resource allocation by subproblem - The red line indicates the default amount of resource for each problem, i.e., with no priority function.

Function – Priority Function	None	MRDL	Norm	R.I.	Random
Lunar	0.656 (0.034)	**0.687 (0.057)**∗	0.666 (0.060)	0.683 (0.067)	0.664 (0.047)
UF1	0.861 (0.011)	0.863 (0.015)	0.833 (0.022)	**0.877 (0.013)**∗	0.874 (0.015)∗
UF2	0.750 (0.009)	0.750 (0.005)	0.762 (0.010)∗	0.772 (0.008)∗	**0.773 (0.008)**∗
UF3	0.844 (0.044)	0.860 (0.043)	**0.944 (0.018)**∗	0.918 (0.029)∗	0.909 (0.037)∗
UF4	0.364 (0.005)	0.366 (0.003)	0.372 (0.003) ∗	0.371 (0.004)∗	**0.373 (0.004)**∗
UF5	0.629 (0.022)	0.663 (0.024)∗	0.754 (0.034)∗	**0.811 (0.015)**∗	0.810 (0.016)∗
UF6	0.661 (0.020)	0.660 (0.014)	0.662 (0.020)	0.686 (0.014)∗	**0.689 (0.015)**∗
UF7	0.803 (0.010)	0.801 (0.010)	0.818 (0.012)∗	**0.837 (0.005)**∗	0.834 (0.006)∗
UF8	0.894 (0.004)	0.900 (0.004)∗	0.914 (0.005)∗	**0.922 (0.003)**∗	0.916 (0.004)∗
UF9	0.931 (0.004)	0.932 (0.004)	**0.944 (0.014)**∗	0.932 (0.004)	0.940 (0.008)∗
UF10	0776 (0.017)	0.786 (0.017)	0.835 (0.035)∗	**0.861 (0.033)**∗	0.839 (0.026)∗
DTLZ1	0.989 (0.003)	0.991 (0.004)∗	0.997 (0.002)∗	0.998 (0.002)∗	**0.998 (0.001)**∗
DTLZ2	0.910 (0.002)	0.912 (0.002)∗	**0.922 (0.001)**∗	0.921 (0.001)∗	**0.922 (0.001)**∗
DTLZ3	0.960 (0.015)	0.969 (0.016)∗	0.992 (0.009)∗	0.991 (0.009)∗	**0.993 (0.006)**∗
DTLZ4	0.905 (0.003)	0.907 (0.004)∗	0.920 (0.001)∗	**0.921 (0.004)**∗	0.918 (0.002)∗
DTLZ5	0.895 (0.003)∗	0.898 (0.002)∗	**0.910 (0.001)**∗	0.908 (0.002)∗	**0.910 (0.001)**∗
DTLZ6	0.837 (0.035)	0.860 (0.021)∗	**0.999 (>0.000)**∗	0.999 (0.001)∗	0.999 (0.001)∗
DTLZ7	0.325 (0.056)	0.339 (0.048)∗	**0.688 (0.005)**∗	0.688 (0.006)∗	0.660 (0.011)∗

Table 1: HV median and standard deviation, in parenthesis. The best values found by a priority function is in bold while the priority function with a Star ∗ is statistically different from None (the MOEA/D-DE with no priority function.)

Function – Priority Function	None	MRDL	Norm	R.I.	Random
UF1	0.140 (0.013)	0.128 (0.015)	0.109 (0.016)∗	**0.090 (0.012)**∗	0.093 (0.014)∗
UF2	0.082 (0.006)	0.080 (0.007)	0.060 (0.005)∗	0.060 (0.005)∗	**0.060 (0.004)**∗
UF3	0.260 (0.012	0.257 (0.009)	**0.168 (0.025)**∗	0.183 (0.335)∗	0.214 (0.030)∗
UF4	0.100 (0.003)	0.100 (0.023)	**0.095 (0.002)**∗	0.095 (0.003)∗	**0.095 (0.002)**∗
UF5	1.759 (0.080)	1.648 (0.091)∗	**0.972 (0.056)**∗	1.056 (0.064)∗	1.085 (0.073)∗
UF6	0.121 (0.027)	0.120 (0.017)	0.100 (0.016)∗	**0.078 (0.014)**∗	0.079 (0.016)∗
UF7	0.125 (0.018)	0.127 (0.015)	**0.061 (0.006)**∗	0.068 (0.005)∗	0.074 (0.005)∗
UF8	0.286 (0.012)	0.279 (0.010)∗	**0.229 (0.014)**∗	0.257 (0.020)∗	0.232 (0.006)∗
UF9	0.451 (0.012)	0.439 (0.015)∗	**0.385 (0.020)**∗	0.420 (0.017)∗	0.400 (0.018)∗
UF10	3.693 (0.20)	3.456 (0.229)∗	2.377 (0.241)∗	**2.364 (0.272)**∗	2.639 (0.253)∗
DTLZ1	381.50 (125.13)	337.46 (164.94)	231.00 (086.40)∗	222.46 (105.68)∗	**205.85 (093.83)**∗
DTLZ2	0.158 (0.013)	0.143 (0.010)∗	**0.072 (0.007)**∗	0.095 (0.013)∗	0.085 (0.010)∗
DTLZ3	1248.4 (300.24)	1046.8 (405.65)	572.2 (312.88)∗	**495.2 (267.59)**∗	557.2 (234.31)∗
DTLZ4	0.1732 (0.024)	0.165 (0.037)	0.076 (0.007)∗	**0.072 (0.077)**∗	0.093 (0.017)∗
DTLZ5	0.152 (0.015)	0.139 (0.010)∗	**0.076 (0.007)**∗	0.084 (0.010)∗	0.080 (0.008)∗
DTLZ6	15.971 (2.148)	14.895 (1.347)∗	**0.007 (0.001)**∗	0.508 (0.423)∗	0.664 (0.585)∗
DTLZ7	1.033 (0.153)	1.012 (0.130)	0.044 (0.010)∗	**0.042 (0.013)**∗	0.105 (0.029)∗

Table 2: IGD median and standard deviation, in parenthesis. The best values found by a priority function is in bold while the priority function with a Star ∗ is statistically different from None (the MOEA/D-DE with no priority function.)

Benchmark – Priority Function	None	MRDL	Norm	R.I.	Random
Lunar (Feasible (%))	0.1291 (0.08)	0.0745 (0.13)	0.1113 (0.16)	0.0929 (0.19)	0.0550 (0.10)
Lunar (Non-dominated (%))	0.0016 (0.01)	0.0018 (0.01)	0.0059 (0.09)	0.0083 (0.09)	0.0030 (0.02)
UF (Non-dominated (%))	0.34 (0.04)	0.35 (0.04)	0.84 (0.06)	0.58 (0.10)	0.69 (0.05)
DTLZ (Non-dominated (%))	0.10 (0.03)	0.13 (0.03)	0.97 (0.05)	0.68 (0.19)	0.66 (0.13)

Table 3: Mean of the percentage of the median values and mean of the median values of the standard deviation (in) parenthesis of non-dominated solutions on UF and DTLZ benchmarks. None is the MOEA/D-DE with no priority function.

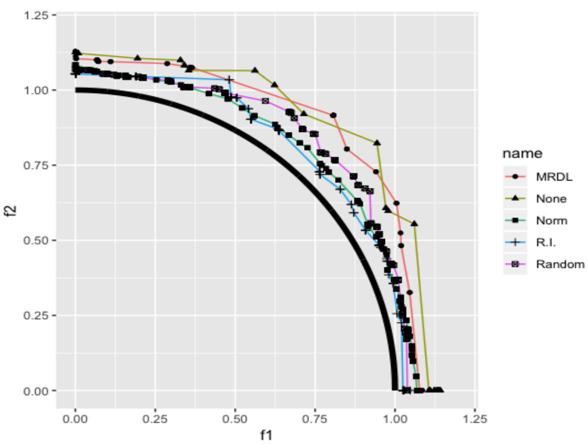

Figure 4: Pareto Front approximations of all priority functions and MOEA/D-DE (None).

the context of the MOP. It called our attention the results form the priority functions Norm and R.I. in the UF9 function, since it appears to be that they prioritized subproblems in an opposite way. In the DTLZ4 they appear to prioritize similar subproblems, focusing the search around the subproblem 280. The distribution of resource is less abrupt in the case of Norm, however R.I. had better results. MRDL influences weakly the distribution of resource allocation, which might indicate its poor performance. In both cases shown the its distribution got closer to 200 resources per subproblem, the rate of not using any priority function.

6 CONCLUSION

In this paper, we have proposed two new priority functions, related to diversity. One based on the MRDL focus on diversity on the objective space while the Norm focus on diversity on the decision space. We then compared the results with the priority function from the MOEA/D-DRA, R.I. and the MOEA/D-DE variant.

To summarize the results, using a priority functions that is based on diversity, as the Norm gave very good results in the artificial benchmark functions, even better than the R.I. a common priority function from the literature while the MRDL performed the best in the Lunar Landing Problem. Therefore, we suggest exploring priority function based on diversity is a preeminent direction of study that should be further explore.

Although the MRDL had a good result in the Lunar problem, we expected that MRDL would give the best results in terms of improvement in performance in the artificial benchmarks. The reason might be that this method considers the diversity of a solution against all the population while the two best priority functions consider only the a relationship of the current solution against its parent. More effort should be directed to address the question of why random as priority function performed so well.

From the results on two artificial benchmark functions and one real-world problem, we understand that using priority functions related to diversity is one promising way of finding better results. More studies need to be conducted to understand how diversity really affects the improvement of performance. We also understand that in priority functions improve the performance of MOEA/D-DE in both HV and IGD metrics values as well as at the number of non-dominated solutions.

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 MOEADr 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] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*. Springer, 105–145.

[6] 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.

[7] Sven Ove Hansson. 2005. Decision Theory A Brief Introduction. Department of Philosophy and the History of Technology, Royal Institute of Technology (KTH) Stockholm. (2005).

[8] 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).

[9] 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.

[10] Takeshi Hoshino Tatsuaki Hashimoto Takeshi Watanabe Tomoaki Tatsuaki Akira Oyama Mari Nishiyama, Hisashi Otake. 2015. Selection of Landing Sites for Lunar Lander with "KAGUYA" Data Using Multi-Objective Optimization (In Japanese with English abstract). In *Space Science Informatics Symposium FY2014*.

[11] 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.

[12] 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.

[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] Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, Wudong Liu, and Santosh Tiwari. 2008. Multiobjective optimization test instances for the CEC 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report* 264 (2008).

[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.