# 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 decision space priority achieved high HV and IGD values, excellent rate of non-dominated solutions on the benchmark problems, and highest rate of feasible solutions among all priority functions in the severely constrained lunar exploration 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. The set of optimal trade off solutions in the decision space is the *Pareto Set (PS)*, and the image of this set in the objective space is the *Pareto Front (PF)*. 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) [14] 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 [18]. 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, 15, 18]. 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 [7]. 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 Norm. It considers diversity of compared 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 and with the standard MOEA/D (with no priority function). The results show that a priority function focused on decision space lead 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. To our surprise, priority function on objective space did not perform so well, barely improving the results compared with no Resource Allocation at all. On the Lunar Landing problem, all methods achieved similar HV values but, the proposed priority function obtained higher proportion of feasible solutions.

# 2 BACKGROUND

## 2.1 Priority functions

We define priority functions (also called utility functions) as one way of establishing preferences [3] between solutions for resource allocation. These functions are used to decide how to allocate computational resources among subproblems by monitoring the algorithm search and guiding the distribution over iterations [1].

Only a few studies have been concerned with resource allocation. We highlight two groups. The first is composed by MOEA/D-GRA [18], MOEA/D-DRA [15] and in the Two-Level Stable Matching-Based Selection in MOEA/D [12]. The other is composed by EAG-MOEA/D [1] and MOEA/D-CRA [8].

According to Zhou and Zhang [18], MOEA/D-GRA could be seen as an extension of MOEA/D-DRA and MOEA/D-AMS [4]. They reason that all these algorithms use a very similar priority function and that MOEA/D-GRA can simulate the behavior of MOEA/D-DRA or MOEA/D-AMS by changing the values of a single parameter.

This priority function is named as the Relative Improvement (R.I.) and defines the priority values of each subproblem $i = 1, \dots, N$, as

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

Where new function value is the value at the current iteration ($T$) and old function value is the value at iteration ($T - \Delta T$).

For MOEA/D-GRA $u_i = \delta_i$, but in MOEA/D-DRA, as well as in the Two-Level Stable Matching-Based Selection in MOEA/D, a second equation is used,

$$u_i = \begin{cases} (0.95 + 0.05 \cdot \frac{\delta_i}{0.001} \cdot u_i), & \text{if } \delta_i > 0.001, \\ 1, & \text{otherwise} \end{cases}$$

The R.I. is based the assumption that if a subproblem has been improved over the last $\Delta T$ iteration (*old function value*), it should have a high probability of being improved over the next iterations.

The priority function used EAG-MOEA/D [1] and MOEA/D-CRA [8] differ from the ones in the MOEA/D-GRA group. In their case, the framework keeps two populations: one working population, and one external archive. This priority function estimates priorities for a subproblem given the number of solutions from that subproblem that are in the external archive.

Together these studies indicate that it is worth monitoring the algorithm behavior and guiding its search, but it is unclear how the choice priority functions influence the results since in all the impact of using priority functions was not isolated. In all Resource Allocation works mentioned above the choice of priority function was just one of multiple changes applied to the base framework. For example, in Zhang et al. used a 10-tournament selection in MOEA/D-DRA [15], while Zhou and Zhang used a new replacement strategy in MOEA/D-GRA [18]. Chiang in MOEA/D-AMS proposes an adaptive mating selection mechanism as dynamically adjusts the mating pools of individuals [4]. Finally, both the studies of Cai and Lai in EAG-MOEA/D [1] and Kang et al. in MOEA/D-CRA [8] used an archive population.

Based on the recent success of Resource Allocation with priority functions, we aim to isolate priority functions by analyzing their impact in MOEA/D. In this study we propose two new priority functions to further understand how priority functions influence the performance of MOEA/D framework. We believe in the idea that diversity is a critical issue in the search process for any multi-objective algorithm. Therefore, we consider using priority functions to address lack of diversity aiming to make solutions better spread among each other.

The two proposed priority functions focus on different aspects of diversity: how to better spread solutions along the Pareto Front (diversity on the objective space) and how to better spread solutions the Pareto Set (diversity on the decision space). For the first we define the MRDL priority function and for the second, the Norm priority function.

## 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) [19] and the Inverted Generational Distance (IGD) [16] 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 those that calculate the diversity during the execution of the algorithm. Those are: the sigma method [11] that requires that the PF lies in the positive objective space; measurement of the entropy of solutions by using Parzen window density estimation-[13], that is sensitive to kernel width; and the maximum relative diversity loss, MRDL, [7], a very expensive method - $O(N^2)$, $N$ being the size of the parent population.

In this work we chose to apply the MRDL as the strategy to measure diversity on the objective space. This is an online diversity metric estimates the diversity loss of a solution to the whole population [7]. 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.

To deal with diversity on the decision space we consider the similarity of decision vectors of consecutive iteractions given by the (2-)Norm. The 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 Norm priority function. The first is that while the R.I. considers the function values of solutions the Norm considers its decision values. The second difference is that R.I. also considers equation 2.1.

# 3 MOEA/D WITH PRIORITY FUNCTIONS

In this study we use the basic framework in algorithm 1 with 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 without a population archive. In consequence any priority functions might be easily integrated to the MOEA/D framework.

**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 iterations, keep updating u by a priority function.
```

**Algorithm 2** 2-Norm

```
Input: X^t decision vectors of solutions; X^{t-1}, decision vectors from the previous solutions; N, the population size.
for i=1 to N do
    u[i] = ||X_i^t - X_i^{t-1}||
u = scale (u) // between 0 and 1
return u
```

This basic algorithm is similar to the MOEA/D-DE [15] with exception of lines 4 and 7. Line 4 allocation computational resources to a solution based on their priority value $u_i$, while line 7 updates the priority function values.

Line 4 deals with the selection of solutions given their priority function values, while the line 7 deals with the calculation of the priority function values. All other procedures and parameters are the same as in MOEA/D-DE [9]. We highlight that the neighborhood is only calculated in the initialization period.

The selection of priority functions provides an important way to control MOEA/D. They allow ways of designing MOEA/D variants that might focus on 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 study diversity on the objective space, diversity in the decision space, 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 methods as well as a random (control) method and we describe how to calculate them in details.

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 of $\Delta T$ iterations. Here $\Delta T = 20$ for artificial benchmarks, as in MOEA/D-GRA [18], while for the real-world problems, we chose $\Delta T = 2$, by trial and error. A sensitivity analysis should be performed for deciding suitable initial values for $u$ and for $\Delta T$.

It should also be noted that if the priority function values results in less than 3 subproblems being updated in one iteration, we reset the priority vector $u = 1$ and all subproblems will be chosen for offspring reproduction at the that iteration.

**Algorithm 3** MRDL

```
Input: old MRDL (initial value is 0); Y^t, objective function values from the incumbent solutions; Y^{t-1}, objective function values from the incumbent solutions of the previous iteraction; N, the population size.
for i=1 to N do
    find index h where (Y_h^{t-1} ≥ Y_i^t) and ||Y_h^{t-1} − Y_i^t|| is minimal.
    if If none is found then
        MRDL[i] = −∞
    else
        d.conv = Y_i^t − Y_h^{t-1}.
        for j=1 to N do
            p′ = Y_j^{t-1} − Y_h^{t-1}
            c′ = Y_j^t − Y_i^t
            proj_{d.conv} * p′ = sum(conv · p′) / (p′ × p′) * p′
            proj_{d.conv} * c′ = sum(conv · c′) / (c′ × c′) * c′
            RDL_j = ||p′ − proj_{d.conv} p′|| / ||c′ − proj_{d.conv} c′||

        MRDL[i] = maximum RDL
u = 1 - scale (MRDL - old MRDL) // between 0 and 1
return u, MRDL
```

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

To consider diversity on the decision space, we propose a priority function based on the norm of the difference between the current solution and its parent. Algorithm 2 gives the details on implementation.

The priority function proposed that considers diversity on the objective space is based on the (2-)Norm of the difference of the current solution to its parent.

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

Then, we scale the values to be between 0 and 1, by using the next equation, based on the values from the entire solution set,

$$\text{Norm}_i = (\text{Norm}_i - \min \text{Norm})/(\max \text{Norm} - \min \text{Norm}) \tag{3}$$

The idea of using the Norm as priority function is that by considering diversity as the priority function more resources are given to incumbent solutions that are similar to their parents, forcing them to update more often and leading to a higher exploration of the decision space.

## 3.2 Priority Function - MRDL

To consider diversity on the objective space, we propose a priority function based on the the Maximum Relative Diversity Loss, MRDL [7].

The diversity on objective space as a priority function is based on the Maximum Relative Diversity Loss, MRDL [7]. The idea of using MRDL is that by measuring diversity on the objective space,

**Algorithm 4** Relative Improvement

```
Input: Y^t, objective function values from the incumbent solutions; Y^{t−ΔT}, objective function values from incumbent solution of iteration t − ΔT, u from the previous ΔT iteration;
for i=1 to N do
    δ[i] = (Y^t[i] − Y^{t−1}[i]) / Y^t[i]
    if δ[i] > 0.001 then
        u[i] = (0.95 + 0.05 · δ[i]/0.001) · u[i]
    else
        u[i] = 1
u / (max(u) + 1.0x10^{−50})
return u
```

more resources are given to incumbent solutions that have similar objective function values between two consecutive iteractions. Therefore, it is expected that this will lead to a higher exploration of the objective space. Algorithm 3 gives the details on implementation.

The calculation of MRDL depends on the concept of weak dominance [20]. A solution $a$ weakly dominates $b$ if in all objectives $a \geq b$ (note that $a \geq a$).

Let $N$ be the number of incumbent solutions and the objective values of iteraction $t$ be $Y^t$ and the objectives values of iteraction $t-1$ be $Y^{t-1}$. For each incumbent solution $i$, find index $h \in Y^{T-1}$. This index is the index of a parent that weak dominates the solution $i$. If $h$ is not found (no parent weak dominates the solution) the MRDL value for this solution is set to $-\infty$. Given $i$ and $h$, for each subproblem, the value of Relative Diversity Loss (RDL) is given by

$$RDL = \frac{||p\prime - proj_{d.conv}p\prime||}{||c\prime - proj_{d.conv}c\prime||}. \tag{4}$$

RDL is a diversity measurement quantity that indicates the amount of diversity loss of an individual solution between two consecutive iterations. High values of RDL imply a reduction of the solution spread, since the further an objective vector of a solution is from the convergence direction, the more it contributes in terms of diversity in the objective space [7]. The maximum value of RDL is the MRDL of the solution $i$.

### 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 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 iterations. Algorithm 4 gives the details on implementation of the equation 1.

We highlight that R.I. was first introduced in the context of the unconstrained MOEA competition in the CEC 2009 [15], being the winner of that competition [17]. Also, in this competition the UF benchmark functions were introduced.

### 3.4 Priority Function - Random

The random priority function is used as basis for comparison. Given no information besides the size of the population, we define the vector of priority $u$ from a uniform distribution. Algorithm 5 gives the details on implementation.

**Algorithm 5** Random

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

## 4 EXPERIMENTAL ANALYSIS

To examine the effects of Resource Allocation under different priority functions on the MOEA/D, we perform a comparative experiment on benchmark functions and an optimization problem based on real world data. In this experiment, we use the MOEA/D-DE implemented by the MOEADr package [2], modified to include Resource Allocation as described in the previous section. We compare five different RA strategies: No Resource Allocation, and RA using the following priority functions: MRDL, Norm, Relative Improvement and Random. In the following figures and tables, these strategies are referred, respectively, as: None, MRDL, Norm, R.I. and Random.

### 4.1 Target Problems

Two benchmark problem sets are used. The first one is the DTLZ function set [5], with 100 dimensions and $k$ = dimensions - number of objectives +1, where the number of objectives is 2. The second one is the UF function set [17], with 100 dimensions.

The Lunar Landing problem is an optimization problem about the selection of landing sites [10]. In lunar exploration it is critical to find suitable landing sites for the rovers. A good landing site must provide enough sunshine for power supply, availability of nearby scientifically interesting materials, little communication interference, and low terrain inclination, among other considerations. The optimization problem is characterized by two decision variables: longitude and latitude; three minimization objectives: total continuous shade days, length of communication window (inverted), and inclination angle; and two constraints: maximum continuous shade days and maximum inclination. This problem is considered to be severely constrained, due to the presence of two craters in the landing area.

### 4.2 Experimental Parameters

We use the conventional MOEA/D-DE parameters [9] for each Resource Allocation strategy: update size $nr = 2$, 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 $p_m = 0.03333333$. The decomposition function is Simple-Lattice Design (SLD), the scalar aggregation function is Weighted Sum (WS), the update strategy is the Restricted Update Strategy and we performed a simple linear scaling of the objectives to [0, 1].

For every strategy/function pair we perform 21 repetitions with 70000 function evaluations and population size $N = 350$. Because the Lunar Landing problem is severely constrained, we used a much higher population size $N = 5050$, and a slightly lower number of function evaluations (60000), following the winner of a recent competition using this problem [6].

### 4.3 Experimental Evaluation

We compare the results of the different strategies based on their Hypervolume (HV) and Inverted Generational Distance [1] (IGD) metrics. Higher values of the HV indicate better approximations of the Pareto Front, while lower values of the IGD indicate better approximations. We also evaluate the proportion of non-dominated solutions and the number of feasible solutions.

For the calculation of HV, the objective function was scaled to the $0, 1$ interval, and the reference point was set to $(1, 1)$ for the 2-objective benchmark problems, $(1, 1, 1)$ for the 3-objective benchmark problems, and $(1, 0, 1)$ for the Lunar Landing Problem [6]).

To verify any statistical differences in the results for the different strategies, we use the Pairwise Wilcoxon Rank Sum Tests with confidence interval $\alpha = 0.05$ and with the Hommel adjustment method for multiple comparisons. For reproducibility purposes, all the code and data used in these experiments are available at [ANONYMIZED].

## 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. Figure 2 does the same but in terms of the IGD values, but only to the artificial benchmarks. Finally, Figure 3 illustrates the PF approximation for the DTLZ4 found by all priority functions and without it. These Figures are representation of the results for other functions. The (box-plot) Figures for all functions are available in the supplementary materials.

In all these Figures, we can see that, in general, using resource allocation performs better than not using Resource Allocation. The Tables 1 and 2 reinforces these results. The priority functions R.I., Norm, and Random achieve similar results in terms of HV values. These results are indicated by the box-plots 1, the Table 1, the approximation to the PF 3, and the Pairwise Wilcoxon Rank Sum Tests 3. We ask ourselves if the fact that Random performed as well as Norm and RI in HV indicates that there is still space for finding more appropriate priority functions. For IGD values, the same trend is confirmed, however, there is statistically significance difference between the results of Norm and R.I 3. On the Lunar Landing Problem, all strategies found similar Hypervolume results (Table 1).

Now we move to the results in the Table 1. It shows the results for every priority function measured by HV and IGD. In the UF functions and considering the results of the HV metric, Norm as priority function had few good results. with the best median in UF3 and UF9. The R.I. had the higher median in five functions. Surprisingly, the Random priority function got the higher median in the UF2, UF4 and UF6. When considering IGD the Norm had the best median in UF3, UF4, UF5, UF7, UF8 and UF9 functions. The R.I. had the highest median in the 4 functions. Again, Random surprised us, being the best in UF2 and UF4. In both metrics, MRDL had slightly better results than MOEA/D-DE.

[1]IGD could not be calculated for the Lunar Landing problem, which has no Ideal Reference Pareto Front.

Now, for the DTLZ set and first considering HV values, Norm as priority function lead to several good results in median in DTLZ2, DTLZ6 and DTLZ7 functions. R.I. performed as the best algorithm in terms of median of HV values also in 3 functions. Reinforcing our surprise, the Random priority function got the higher median in the DTLZ1, DTLZ2, DTLZ3 and DTLZ5 functions. For the values of the IGD, Norm had the best medians in the functions: DTLZ2, DTLZ5 and DTLZ6, again with the same number of best results as R.I. Here, Only in the DTLZ1 Random had the highest values.

On the Lunar Landing Problem, the best priority function in terms of median HV values is the MRDL. However, as commented above, the results are all similar.

### 5.1 Proportion of Non-dominated and Feasible Solutions

Another difference that we see among the Resource Allocation strategies is found in the proportion of non dominated and feasible solutions. The results on the Table 2 indicate that the Norm strategy leads to a very high rate of non-dominated solutions in the final solution set. The MRDL priority function improved a little the rate of non-dominated solutions. The same behavior (better rate of non-dominated solutions) is found in the R.I. and Random results.

On the Lunar Landing problem, Norm had the highest rate of feasible solutions, but had the second best rate of non-dominated solutions, surpassed by R.I.

### 5.2 Resource Allocation

Figure 4 illustrates the amount resource allocated by Norm, R.I. and MRDL to every subproblem on UF3 and DTLZ4 problems. We show images from the UF9 and DTLZ4 functions due to space limitations. 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 uniformly noisy.

It is clear from this Figure 4 that, during the execution of the algorithm, the Resource Allocation to each subproblem is different. This behavior is also different for each 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 different MOPs.

We highlight that 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, assigning similar priorities to the same subproblems. The distribution of resources is less abrupt in the case of Norm, however R.I. had better results, in these problems. MRDL influences weakly the distribution of Resource Allocation, which might indicate its poor performance. In both MRDL cases shown the distribution got closer to 200 resources per subproblem, similar to not using any priority function.

## 6 DISCUSSION

The aim of the present research was to investigate how priority functions relate to MOEA/D. We proposed two new priority functions (related to diversity) for estimating difficulty and for calculating priorities among subproblems for better Resource Allocation. We

| HV | 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) | **0.88 (0.013)** | 0.874 (0.015) |
| UF2 | 0.750 (0.009) | 0.750 (0.005) | 0.762 (0.010) | 0.82 (0.008) | **0.83 (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 | 0860 (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) |

| IGD | 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 (0.200) | 3.456 (0.229) | 2.380 (0.241) | **2.364 (0.272)** | 2.639 (0.253) |
| DTLZ1 | 381.5 (125.1) | 337.5 (164.9) | 231.0 (086.4) | 222.5 (105.7) | **205.8 (093.8)** |
| DTLZ2 | 0.158 (0.013) | 0.143 (0.010) | **0.072 (0.007)** | 0.095 (0.013) | 0.085 (0.010) |
| DTLZ3 | 1248 (300.2) | 1047 (405.6) | 572.2 (312.9) | **495.2 (267.6)** | 557.2 (234.3) |
| DTLZ4 | 0.173 (0.024) | 0.165 (0.037) | 0.076 (0.007) | **0.072 (0.08)** | 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 | 15.97 (2.148) | 14.89 (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 1: HV and IGD medians and standard deviations, in parenthesis for every function/priority function. The best values found by a priority function are in bold. Standard deviation was used as tie breaker.**

| Rates | 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 2: 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.**

*isolated* the priority functions in MOEA/D as the only variant. This allowed us to examine their effect on the performance of MOEA/D.

These two new priority functions focus on different aspects of diversity. The first, MRDL, addresses diversity on the objective space

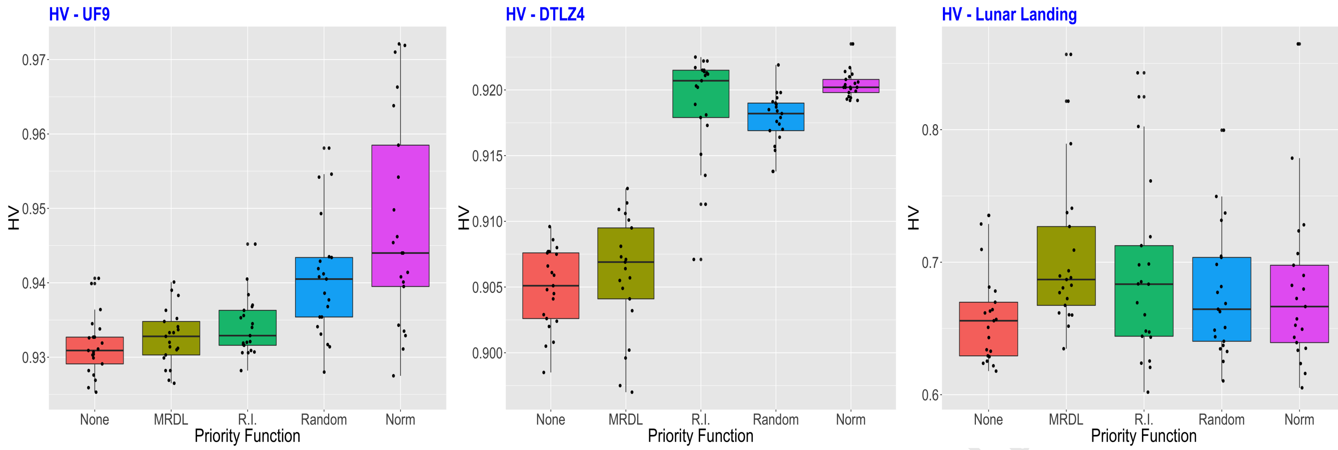


**Figure 1: Box plot of HV values on UF9, DTLZ4 and Lunar Landing.**

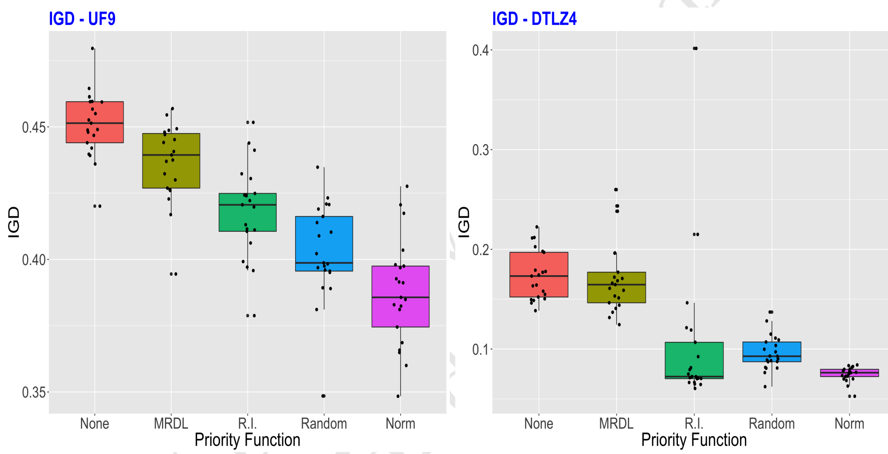


**Figure 2: Box plot of IGD values on UF9 and DTLZ4.**

| HV | MRDL | None | Norm | R.I. |
|---|---|---|---|---|
| None | 0.82 | - | - | - |
| Norm | 2.9e-06 | 6.2e-07 | - | - |
| R.I. | 8.6e-08 | 1.3e-08 | 0.82 | - |
| Random | 2.5e-06 | 3.8e-07 | 0.82 | 0.82 |

| IGD | MRDL | None | Norm | R.I. |
|---|---|---|---|---|
| None | 0.478 | - | - | - |
| Norm | <2e-16 | <2.e-16 | - | - |
| R.I. | 1.6e-13 | 7.6e-14 | 0.109 | - |
| Random | 7.7e-12 | 5.0e-12 | 0.002 | 0.337 |

**Table 3: Statistical Analysis of the HV and IGD results based on the Pairwise Wilcoxon Rank Sum Test. No significant difference between MRDL and None, while other priority functions are statistically different to None.**

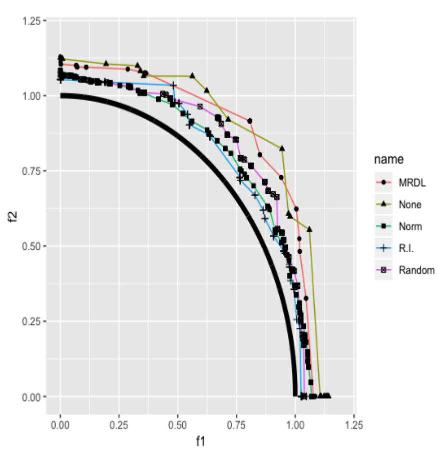


**Figure 3: Pareto Front approximations of all priority functions and None on DTLZ4.**

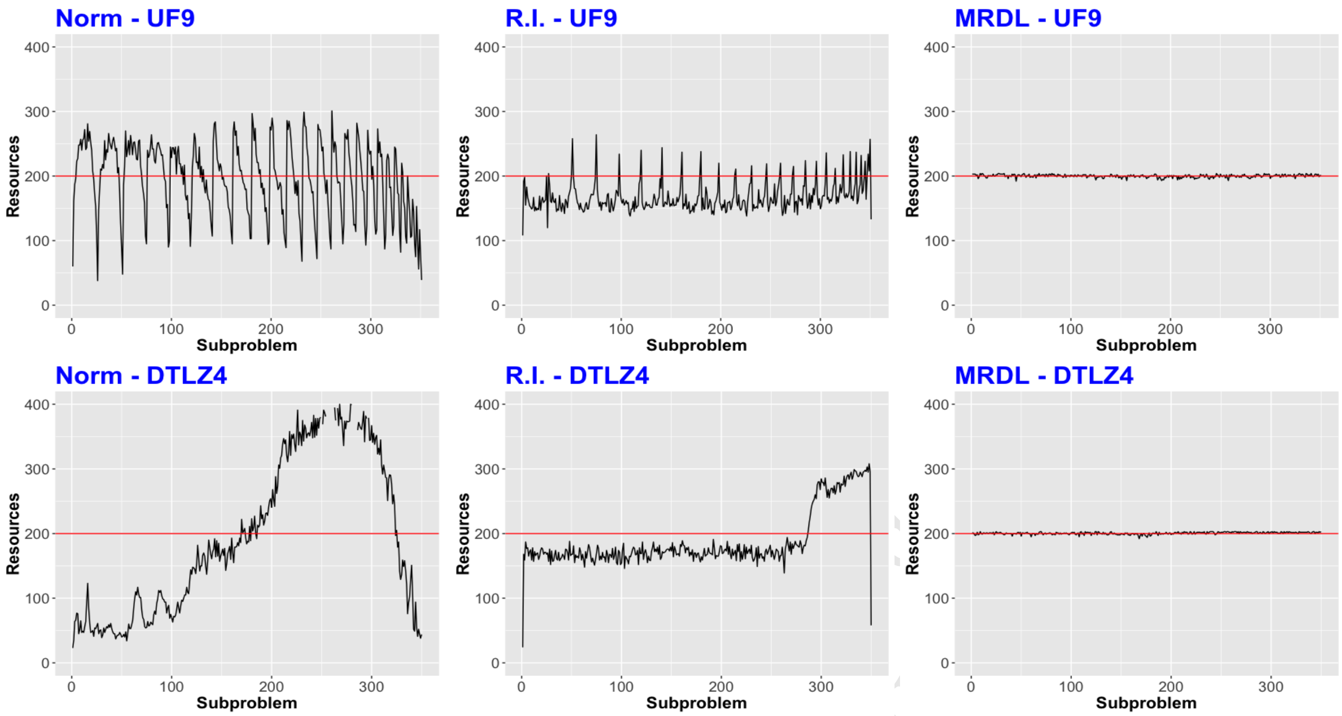


**Figure 4: Resource Allocation by subproblem - The red line indicates the default amount of resource for each problem, i.e., with no priority function.**

while the second, the Norm, addresses diversity on the decision space. We then compared these new priority functions with the most popular approach, the Relative Improvement, and the standard MOEA/D.

This study has shown that using Norm as priority function effectively improves the performance of MOEA/D, since it achieved high HV and IGD values and excellent rates of non-dominated solutions on the benchmark problems. It also lead to the highest rate of feasible solutions among all priority functions in the severely constrained lunar exploration. Some of these results were superior than the results of the R.I. (specially the rate of non-dominated solutions). These results indicate that Norm indeed leads to more diversity of the final solution set, demonstrating the effectiveness of it as a priority function and as a direct way to increase diversity in MOEA/D. This suggests that it really there is a role for diversity in promoting better performance in HV and IGD metrics as well as higher rates of non-dominated solutions. In contrast, it seems MRDL performed just slightly better than MOEA/D and it did not generate enough difference to serve as an effective priority function. However, given the surprising results of Random, we infer that there is still space for finding more appropriate priority functions.

Overall, the findings of this work strengthens the idea that exploring priority function focusing on critical issues (such as diversity and proportion of non-dominated solutions) is worth of attention. This suggests the choice of priority functions is a critical component of a Resource Allocation system. Our results recommend R.I. or Norm as reasonable choices for Resource Allocation depending on the MOP being addressed.

In this work, we do not yet consider archive based Resource Allocation and archive based priority functions, such as MOEA/D-CRA [8]. We will address this issue in a continuation to this study. There are many components and variants of MOEA/D and is interesting to combine the Norm priority function with the them. Then, we can further explore the relationship of priority functions based on diversity with others components and variants of the MOEA/D framework. How to define more efficient and effective utility functions for different problems is also worth further investigation (such as priority function that also consider constraints) as well as to verify the results of using priority function in other real-world problems.

# 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 and Claus Aranha. 2018. MOEADr: Component-Wise MOEA/D Implementation. URL https://cran.R-project.org/package=MOEADr. (2018). R package version 1.2.0.

[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] Japanese Society for Evolutionary Competition. 2018. Evolutionary Competition 2018. (2018). http://www.jpnsec.org/files/competition2018/EC-Symposium-2018-Competition.html in Japanese.

[7] 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.

[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] MD Nasir, Arnab Kumar Mondal, Soumyadip Sengupta, Swagatam Das, and Ajith Abraham. 2011. An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 765–772.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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).

[18] 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.

[19] 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.

[20] 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.