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

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 treated uniformly and there is no priority among them. Each subproblem is related to an area of the theoretical Pareto Front. It is expected that some 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 have been proposed. In this paper we investigate the priority functions used to decide which subproblems a should receive more resources. We propose that using diversity as the priority criteria has a positive effect on resource allocation. We consider four different priority functions: Diversity on objective space, diversity on decision space, relative improvement (Used on MOEA/D-DRA), and a random priority function for control. WE conducted an experimental analysis on the DTLZ and UF benchmark functions and on the lunar lading real-problem and compared the famous MOEA/D-DE variant with each four priority functions and without it any. The results indicate XXX.

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:
XXX, XXX, and XXX. 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, 12 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.

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

where m is the number of objective functions and $\mathbb{R}^m$ is the objective function space. $x \in \mathbb{R}^D = \{x_1, x_2, ..., x_D\}$ is a D-dimensional vector which represents a candidate solution with D variables, $f : \mathbb{R}^D \to \mathbb{R}^m$ is a vector of objective functions.

These objectives often conflict with each other, as there is no vector $x \in \mathbb{R}^D$ 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 $f(u) \prec f(v)$, if and only if $f_k(u) \leq f_k(v), \forall_k \in \{1, ..., m\}$ 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 $f(y) \succ f(x)$, 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.

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).

$$PS = \{x \in \mathbb{R}^D | \nexists y \in \mathbb{R}^D : f(y) \succ f(x)\}, \tag{2}$$

$$PF = \{f(x) | x \in PS\}. \tag{3}$$

We are interested in analyzing the Multi-objective Evolutionary Algorithm based on Decomposition framework, MOEA/D [12]. 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 iteractions). 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. 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

*Produces the permission block, and copyright information
[†]XXX
[‡]XXX
[§]XXX

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

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

more to the algorithm's search. In the works of Zhang et al. [13] and Zhou et al. [16] 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 considers the integration of an online diversity metric based on a geometrical perspective [5] as a direct way to define the priority function. The second addresses diversity in the decision space using the 2-norm.

Our reason for considering diversity metrics is that we interpret that these priority functions are able monitor the diversity of the solutions of an algorithm and may be used to decide how to distribute the computation resources among subproblems given this important issue, therefore better guiding the search behavior of the algorithm.

The results found supported that idea, since in the real-world Lunar Landing problem these priority functions performed very well (with emphasis on the results of the 2-norm). In the UF benchmark problem only 2-norm performed well, followed closely by the relative improvement. All code is available for reproducibility purpose.

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. 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 [13]; MOEA/D-GRA [16]; MOEA/D-AMS [4]. According to Zhou and Zhang [16], 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

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

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 [7] 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 [13] was used while in MOEA/D-GRA a new replacement strategy was also consider [16]. 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 [7] 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 Pareto Set (diversity on the decision space) and solutions better spread along the Pareto Front (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 4, 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) [17] and the Inverted Generational Distance (IGD) [14] 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 consider the maximum relative diversity loss 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 important 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 u by a priority function.
```

In this study we only use the basic algorithm framework of MOEA/D-GRA, in Algorithm 1. This Algorithm 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 [13]. Likewise, all reproduction procedures and parameters are the same as in MOEA/D-DE [8]. We highlight that the neighborhood is only calculated in the initialization period.

In contrast to Zhou and Zhang in the MOEA/D-GRA, however, we only consider the basic algorithm and no other variant that it was introduced in their work [16], such as the new replacement procedure introduced. 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.

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].

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 $\Delta T = 20$ for artificial benchmarks, as in MOEA/D-GRA [16], while for the real-world problems, we chose

Algorithm 2 2-norm

```
Input: NEW new incumbent solutions; OLD, previous iteraction 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
```

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**

    $p = P_j - P_h$

    $c = c_j - c_i$

    $$proj_{d.conv_y} * p\prime = \frac{sum(conv_y * p\prime)}{crossprod(p\prime)} * p\prime$$

    $$proj_{d.conv_y} * c\prime = \frac{sum(conv_y * c\prime)}{crossprod(c\prime)} * c\prime$$

    $$RDL = \frac{||p\prime - proj_{d.conv_y} p\prime||}{||c\prime - proj_{d.conv_y} c\prime||}$$

  MDRL[i] = maximum RDL

u = 1 - scale (MRDL - old MRDL) // between 0 and 1

return u, MRDL

$\Delta T = 2$, 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 2-Norm of the difference of current solutions and their parents

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

$$2 \text{ norm}_i = ||\text{current solution} - \text{parent solution}|| \quad (5)$$

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. Therefore 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 priority function proposed that considers diversity on the objective space is based on the Maximum Relative Diversity Loss, MRDL. The idea of the 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.

MRDL is an online diversity metric that estimates the diversity loss of a solution to the whole population [5]. It is an useful metric since 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. It uses 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.

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, $\Gamma^{p \to c}$.

$$\Gamma_i^{p \to c} = \max_{i=1,\ldots,k} \Gamma_{d.conv_y}^{p \to c}. \tag{6}$$

At every iteraction and each incumbent solution, we need to estimate k convergence directions (shown later) and then 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.

RDL is a diversity measurement 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. 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

$$\Gamma_{d.conv_y}^{p \to c} = \frac{||p\prime - proj_{d.conv_y} p\prime||}{||c\prime - proj_{d.conv_y} c\prime||}. \tag{7}$$

$p\prime$ and $c\prime$ are given by:

$$\begin{aligned} p\prime &= p - p_r, \\ c\prime &= c - c_s, \end{aligned} \tag{8}$$

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.

$$proj_{d.conv_y} p\prime = \frac{d.conv_y \cdot p\prime}{|p\prime|^2} p\prime. \tag{9}$$

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

$$||p\prime - proj_{d.conv_y} p\prime|| = \text{2-norm}(p\prime - proj_{d.conv_y} p\prime) \tag{10}$$

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. [18] study, weak dominance ($A \succeq B$) means that any solution in set B is weakly dominated by a solution in set A. However, this does not rule out equality, because $A \succeq A$ for all approximation sets A.

$$d.conv_y = c_j - p_h \tag{11}$$

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

$$D_c = \{d | \exists c_d \prec p_k, k \in 1, \ldots, N, d \in [1, \ldots, |C|]\} z \tag{12}$$

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.

$$h = \underset{k \in D_p}{argmin} ||p_k - c_j|| \tag{13}$$

$$D_p = \{k | \exists c_j \prec p_k, k \in 1, \ldots, N\} \tag{14}$$

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 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
```

Algorithm 5 Random

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

3.3 Priority Function - Relative Improvement

Here we give a brief description of the Relative Improvement, 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 4. For more information refer to [13] and [16].

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 apply MOEA/-DE with the 4 priority functions to two artificial benchmark problems and the Lunar Landing real-world problem and we compared the performance against the results of the MOEA/-DE.

4.1 Artificial Benchmark Functions

We compared the variants of priority functions using the DTLZ benchmark functions from "Scalable test problems for evolutionary multi-objective optimization" with 100 dimensions and $k = dimensions$ − number of objectives + 1, where the number of objectives is 2 or 3. We also consider the UF benchmark functions from [15], also with 100 dimensions. .

4.2 Parameter Settings - Artificial benchmarks

For every combination, 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 $(1, 1)$. While, 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. For reproducibility the code is made available at XXX.

4.3 Real World Benchmark Function

The Lunar Landing problem is a real-world problem that simulates the selection of landing sites for lunar landers [9]. In lunar exploration plan, finding suitable landing site of the rovers has a very important function. One of the reasons is that these rovers power supply depend on solar power, therefore ensuring sunshine is a critical issue. Also, it is important to landing in a site with the presence of scientifically interesting materials while this site should provide little difficulties to the exploration.

This is a a minimization design problem in which the two decision variables are the longitude and latitude with the objectives being: 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.

Although the number of design variables is small as two (latitude and longitude), it 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

4.4 Parameter Settings - Lunar Landing problem

He highlight only the parameters that differ from the settings for the artificial benchmarks. Since this problems is a severe constrained one, we chose the population size $N = 5050$ and the number of functions evaluations is 30000/60000.

We perform statistical tests on the hypervolume (HV) metric values for measuring the quality of a set of obtained non-dominated solutions found by the algorithms on the Lunar Landing problem.The reference point for the HV calculation was set to $(1, 0, 1)$. Higher values of the HV indicate better approximations while lower values. 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

It is in our understand that using priority functions in both real-world and artificial benchmarks improve the results of MOEA/D-DE. For all group of functions the performance of MOEA/D-DE, in terms of HV or IGD median values, was always suppressed by at least one variation using priority functions.

5.1 UF benchmark functions

Figure 1 shows box-plot that exemplify the results found in the UF benchmark functions in terms of the HV values while Figure 3 does the same but in terms of the IGD values. In them we can see that MRDL as a priority function is slightly better than MOEA/D-DE, with the difference of the median being statistical different in the UF5 and UF8, considering both HV and IGD values. 2-Norm, R.I. and Random perform the best, in both terms of median and standard deviation and that

Turning to the results of the Table 1, we discuss the results of every priority function. The 2-norm as priority function lead to several good results in median of the HV values while had very good results in the median of the IGD values. for HV values it had the best median in the UF3 and UF9 cases. Although for the UF3 problem, there is no statistical significant difference for the R.I. while for the UF9, there is no statistical significant difference for the Random. For the UF4 and UF10 its median value is not the best, but this difference is not statistical significant. The relative improvement function was first introduced in the context of the unconstrained MOEA competition in the CEC 2009 [13], being the winner of that competition [15]. This competition introduced the UF benchmark functions, so it came to us with no surprising the good results from the relative improvement priority function. In terms of HV values, in five functions it had the higher median. 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 2-Norm) and UF6 functions(with no statistical difference to R.I.) with its median being always higher that the one from MRDL.

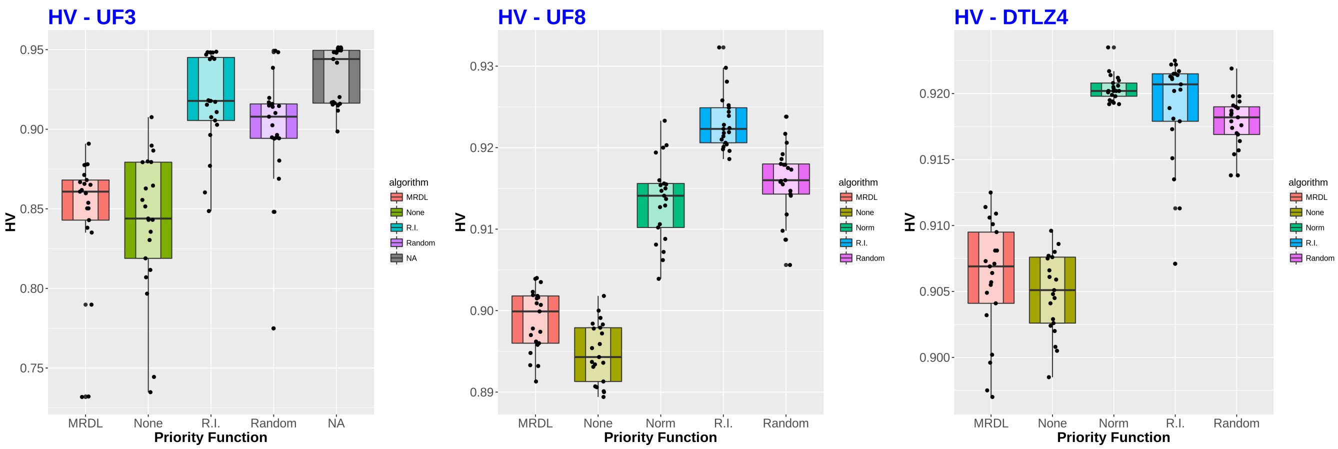
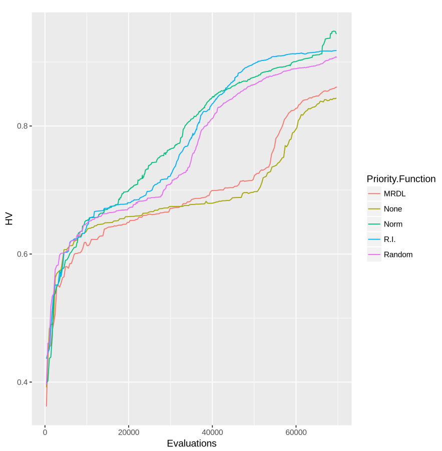
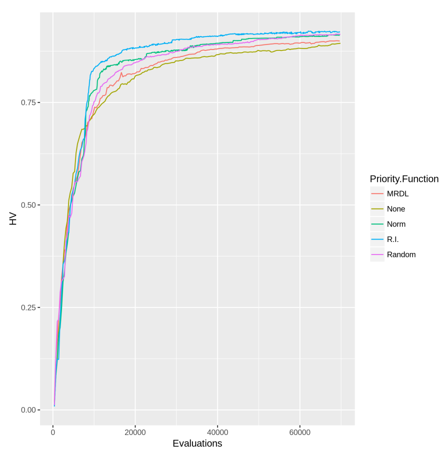


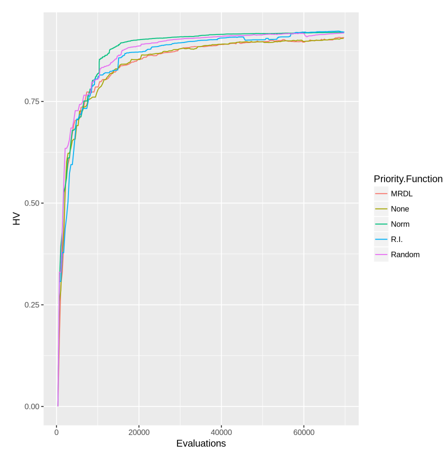
Figure 1: HV values of the last iteraction on Artificial Benchmark Problems



(a) Evolution - UF3



(b) Evolution - UF8



(c) Evolution - DTLZ4

Figure 2: Evolution of HV values on Artificial Benchmark Problems

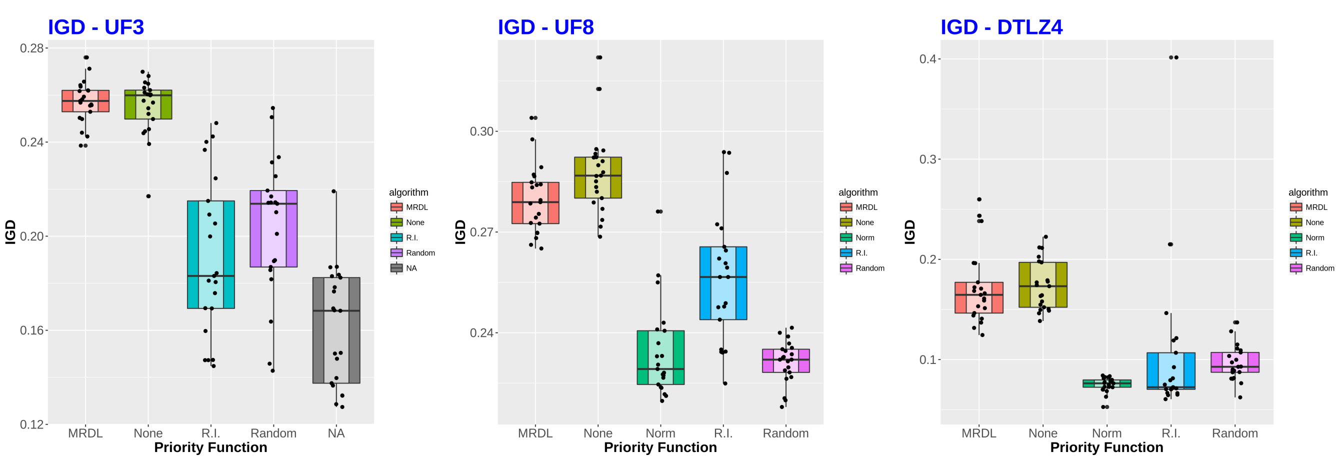
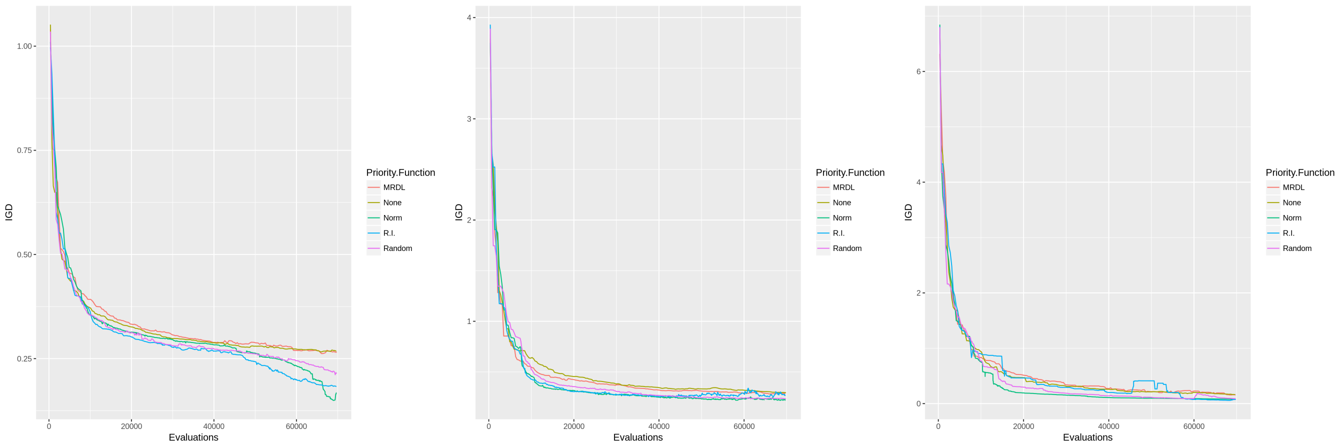


Figure 3: IGD values of the last iteraction on Artificial Benchmark Problems



(a) Evolution - UF3 **(b) Evolution - UF8** **(c) Evolution - DTLZ4**

Figure 4: Evolution of HV values on Artificial Benchmark Problems

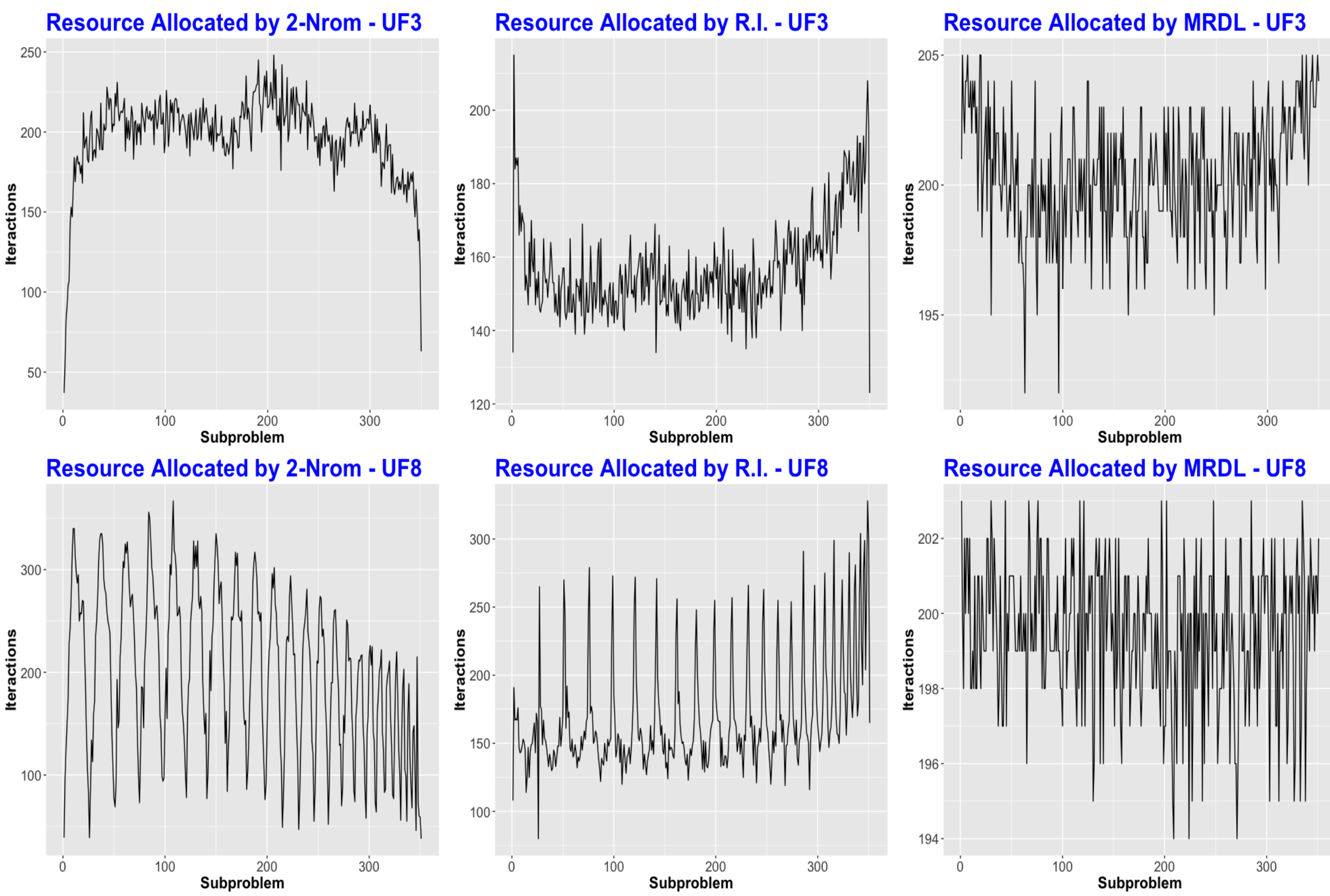
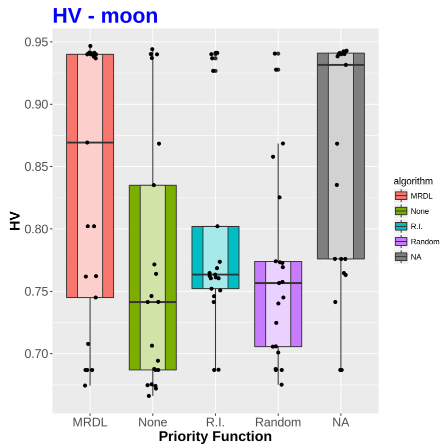
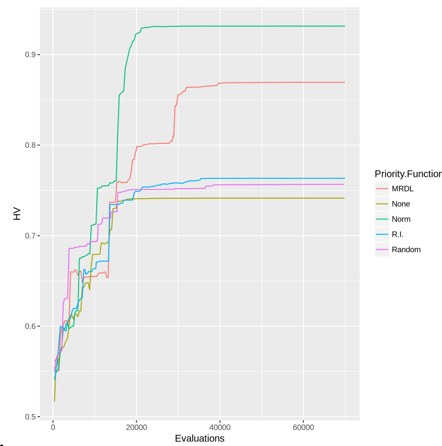


Figure 5: Resource allocation by subproblem on Artificial Benchmark Problems

Moving to the IGD results, in the Table 2 we verify that the 2-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 2-Norm are not statistically different from the R.I. or the Random. In UF8 the results of the 2-Norm not statistically different from the Random. The R.I. had the highest



(a) HV values of the last iteration on the Lunar Landing Problem



(b) Evolution of the HV on the Lunar Landing

Figure 6: SBX crossover - (λ, λ) scheme.

Priority Function:	None	MRDL	2-Norm	R.I.	Random
Lunar	0.74 (0.10)	0.87 (0.11)	**0.93 (0.10)**∗	0.76 (0.08)	0.76 (0.08)
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)**∗	**0.920 (0.004)**∗	0.918 (0.002)∗

Table 1: Results of HV - Median and in parenthesis, the standard deviation. Highlighted in bold are the best values found by a priority function for that function. The priority function that has a Star ∗ is statistically different from MOEA/D-DE without priority function.

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 2-Norm.

Figures 2 and 4 show how the values of the HV and IGD evolve over the iteractions of the algorithms. For the UF3 function the results caught our attention. First, it seems that more evaluations are need to a convergence of the values. Second, for the HV values the priority functions relative improvement and 2-norm made a "jump" over 30000 and 60000 evaluations, improving fast the HV metric values. Also at the end fo the evaluations, there is a strange regressive peak of the HV and IGD values of the 2-Norm. That said, in most cases the values of HV and IGD converged by the end of the number evaluations (70000).

5.2 DTLZ benchmark functions

Figure 1 shows box-plot that exemplify the results found in the DTLZ benchmark functions in terms of the HV values while Figure 3 does the same but in terms of the IGD values. In them t we can see

Priority Function:	None	MRDL	2-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)∗

Table 2: Results of IGD - Median and in parenthesis, the standard deviation. Highlighted in bold are the best values found by a priority function for that function. The priority function that has a Star ∗ is statistically different from MOEA/D-DE without priority function.

that MRDL as a priority function is slightly better than MOEA/D-DE in terms of median. With R.I., 2-Norm and Random performing the best.

Figures 2 and 4 show how the values of the HV and IGD evolve over the iteractions of the algorithms. In most cases the values of HV and IGD converged by the end of the number evaluations (70000).

5.3 Lunar Landing Problem

Figure 6 (a) show box-plot the results found in the Lunar Landing benchmark problem in terms of the HV values. Both priority functions related to diversity, MRDL and 2-norm, improved a lot the performance of the MOEA/D-DE. On the other hand, the relative improvement lead to a diminish of the standard deviation, with just a slight improvement on the performance. Figure 6 (b) show how the values of the HV evolve over the iteractions of the algorithms. All of the variants had converged around 20000 evaluations, with the exception of the MRDL (it converged with 40000). All used around half or less of the total number evaluations (70000) to converge.

Changing to the results of the Table 1, we discuss the results of medians and standard deviation the priority functions. We highlight the outstanding result of 2-norm since it was clearly better than the other priority function results in terms of median of HV values as it is statistically different from MOEA/D-DE, yet its standard deviation was very high. The results of MRDL were not as good as the previous one, however, it is still had impressive median HV values results, but without statistically significant difference from MOEA/D-DE. The relative improvement improved the results of MOEA/D-DE but not as much as in the case of the UF benchmark functions, where it had frequently the best results. Here, on the other hand, the main improvement was in terms of diminishing the standard deviation. More results can be found at the supplementary materials.

5.4 Resource Allocation

Figure 5 illustrates the amount resource allocated by Norm, R.I. and MRDL to every subproblem on UF 3 and 8 problems. We exclude the visuals from MOEAD/D-DE since it give the same amount of resource to every problem (200) and of the Random, since it is completely noise.

It is clear from Figures 5 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 called our attention the results form the priority functions Norm and R.I., since it appears to be that they prioritized subproblems in an opposite way. It is also important to highlight that each priority function influences the search differently given the context of the MOP. At first, it is not clear how MRDL influences the resource allocation in a general way, however when looking at the results of MRDL, we cogitate that if a solution is given more resources its neighbor solutions and vice-versa

5.5 Non-Dominated Solutions and Execution Time

The results on the Table 3 indicate that the 2-Norm leads to the best rate of non-dominated solutions in any benchmark function. Not only that but it is always the fastest. The MRDL priority function improved a little the rate of non-dominated solutions, at the cost of a longer execution time, which may be alleviated by more costly problems, such as real-world problems. The same behavior is found in the R.I. and Random results. More results can be found at the supplementary materials.

Priority Function:	None	MRDL	2-Norm	R.I.	Random
Lunar (Non-dominated (%))	0.69 (0.23)	0.87 (0.18)	0.83 (0.18)	0.69 (0.16)	0.44 (0.16)
Lunar (Time (seconds))	32 (0.92)	39 (11.02)	46 (23.34)	2 (7.75)	52 (0.86)
UF1 (Non-dominated (%))	0.28 (0.03)	0.29 (0.05)	0.94 (0.10)	0.47 (0.09)	0.68 (0.06)
UF1 (Time (seconds))	28 (2.36)	46 (2.16)	1 (0.60)	37 (3.05)	34 (2.87)
UF2 (Non-dominated (%))	0.34 (0.04)	0.34 (0.05)	0.96 (0.05)	0.61 (0.16)	0.81 (0.06)
UF2 (Time (seconds))	24 (0)	41 (0.59)	1 (0.00)	34 (1.47)	31 (0.00)
UF3 (Non-dominated (%))	0.22 (0.03)	0.24 (0.03)	0.78 (0.13)	0.44 (0.15)	0.59 (0.07)
UF3 (Time (seconds))	25 (0.22)	43 (1.55)	1 (0.60)	55 (2.09)	31 (0.00)
UF4 (Non-dominated (%))	0.58 (0.05)	0.60 (0.04)	0.90 (0.08)	0.75 (0.06)	0.82 (0.04)
UF4 (Time (seconds))	24 (0.00)	41 (0.00)	1 (0.00)	34 (1.56)	31 (0.22)
UF5 (Non-dominated (%))	0.21 (0.03)	0.26 (0.04)	0.96 (0.03)	0.69 (0.13)	0.71 (0.08)
UF5 (Time (seconds))	24 (0.22)	41 (0.60)	2 (0.44)	33 (0.54)	31 (0.22)
UF6 (Non-dominated (%))	0.27 (0.03)	0.27 (0.03)	0.95 (0.08)	0.45 (0.07)	0.61 (0.07)
UF6 (Time (seconds))	24 (0.36)	41 (0.79)	1 (0.00)	33 (0.51)	31 (0.00)
UF7 (Non-dominated (%))	0.38 (0.06)	0.37 (0.05)	0.96 (0.03)	0.57 (0.11)	0.80 (0.04)
UF7 (Time (seconds))	24 (0.30)	42 (0.38)	1 (0.46)	34 (0.90)	31 (0.30)
UF8 (Non-dominated (%))	0.40 (0.03)	0.39 (0.03)	0.70 (0.06)	0.67 (0.07)	0.63 (0.03)
UF8 (Time (seconds))	24 (0.00)	39 (0.58)	1 (0.00)	51 (2.23)	31 (0.30)
UF9 (Non-dominated (%))	0.32 (0.03)	0.33 (0.03)	0.52 (0.03)	0.49 (0.04)	0.48 (0.03)
UF9 (Time (seconds))	25 (0.50)	23 (4.59)	1 (27.84)	47 (2.75)	32 (0.48)
UF10 (Non-dominated (%))	0.43 (0.04)	0.46 (0.05)	0.75 (0.05)	0.68 (0.08)	0.73 (0.06)
UF10 (Time (seconds))	25 (0.51)	38 (1.83)	1 (0.00)	47 (3.51)	32 (0.30)
DTLZ1 (Non-dominated (%))	0.05 (0.01)	0.07 (0.01)	0.98 (0.06)	0.62 (0.17)	0.68 (0.13)
DTLZ1 (Time (seconds))	25 (0.90)	41 (1.16)	2 (0.00)	36 (10.19)	32 (0.00)
DTLZ2 (Non-dominated (%))	0.18 (0.03)	0.25 (0.03)	0.96 (0.05)	0.73 (2.93)	0.84 (0.07)
DTLZ2 (Time (seconds))	25 (0.30)	41 (0.54)	1 (12.66)	40 (2.93)	32 (0.60)
DTLZ3 (Non-dominated (%))	0.04 (0.02)	0.04 (0.02)	0.99 (0.05)	0.32 (0.17)	0.44 (0.18)
DTLZ3 (Time (seconds))	25 (0.22)	40 (0.78)	2 (0.51)	32 (3.09)	32 (0.22)
DTLZ4 (Non-dominated (%))	0.11 (0.03)	0.13 (0.03)	0.96 (0.03)	0.7 (0.24)	0.63 (0.13)
DTLZ4 (Time (seconds))	26 (0.36)	43 (1.01)	1 (0.00)	47 (5.94)	33 (0.68)

Table 3: Percentage of the median values and the standard deviation in parenthesis of non-dominated solutions and of the execution time.

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 2-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 2-Norm gave very good results, even better than the R.I. a common priority function from the literature. Therefore, we suggest that this is a direction that should be further explore, for example in real-world problems. We expected that MRDL would give the best results in terms of improvement in performance, but to our surprise it barely helped. 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, However since only the results of the 2-Norm are substantial, we 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. Finally, we add that using priority function could help to prioritize desired characteristics, as diversity (2-Norm) or convergence towards the PF (R.I) and maybe some other features.

There are many other components and variants of MOEA/D and is interesting to combine them with the 2 norm priority function to them. Then we can better understand the relationship of priority functions based on diversity with the 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 as well as verify the results of using priority function in others 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, 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] 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] 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*.

[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] 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.

[13] 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.

[14] 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.

[15] 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).

[16] 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.

[17] 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.

[18] 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.