Resource Allocation and MOEA/D New Priority Functions Based on Diversity*

XXX†
XXX
XXX, XXX
XXX@XXX.XXX

XXX‡
XXX
XXX, XXX
XXX@XXX.XXX

XXX§
XXX
XXX, XXX
XXX@XXX.XXX

ABSTRACT

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

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:
XXX, XXX, and XXX. 2019. Resource Allocation and MOEA/D New Priority Functions Based on Diversity. 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 (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.

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

where m is the number of objective functions and $\mathbb{R}^m$ is the objective function space. $x \in \mathbb{R}^D = \{x_1, x_2, ..., x_D\}$ 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. As in the study by Zitzler et al. [19] 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.

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).

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

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

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

In this study, we are interested in analyzing the Multi-objective Evolutionary Algorithm based on Decomposition framework, MOEA/D [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

*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

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 more to the algorithm's search. In the works of Zhang et al. [13] and Zhou et al. [17] a priority function was proposed aiming to prioritize solutions based on a historical convergence information during different generations. Another approach was implemented in Kang et al. [7], where the priority function was based on the presence of a solution from the main population on a secondary population.

The aim of this study is to explore the relationship between subproblems and priority functions in the context of MOEA/D. Here, we propose a two new functions based to diversity for defining priorities. The first 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 "spectral" norm.

2 RELATED WORKS

2.1 Priority functions

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

Only a few studies have been concerned with resource allocation. We highlight: MOEA/D-DRA [13]; 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

$$u_i = \frac{\text{old function value} - \text{new function value}}{\text{old function value}}. \tag{4}$$

This equation, the relative improvement, 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 was used while in MOEA/D-GRA a new replacement strategy was also consider. MOEA/D-AMS proposes an adaptive mating selection mechanism as dynamically adjusts the mating pools of individuals are included. Finally, both EAG-MOEA/D and MOEA/D-CRA use archive population.

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 relative improvement, equation 4, to further understand how priority functions influence the performance of MOEA/D framework.

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).

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) [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 given by the "spectral" norm. These two new functions are then integrated to the MOEA/D framework and the resource allocation procedure is similar as the one in MOEA/D-GRA. 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.

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.

Algorithm 2 "spectral" norm

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

Algorithm 3 MRDL

Input: old MDRL (default 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 = scale (MDRL - old MDRL) // between 0 and 1
return u, MDRL

3 MOEA/D WITH PRIORITY FUNCTIONS

Algorithm 1 describes MOEA/D with priority functions. Except for line 4, in which a subproblem may not be part of the group that is going to be iterated and for line 7, in which the priority function is calculated, the whole procedure is similar to the MOEA/D-DE [13]. Likewise, all reproduction procedures and parameters are the same as in MOEA/D-DE [8]. It is important to highlight that the neighborhood is only calculated in the initialization period.

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

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

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

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

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 - "Spectral" Norm

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

The idea of the "spectral" 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, $\Gamma^{p \rightarrow c}$.

$$\Gamma_i^{p \to c} = \max_{i=1,\ldots,k} \Gamma_{d.conv_y}^{p \to c}. \tag{5}$$

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

$$\Gamma_{d.conv_y}^{p \to c} = \frac{||p\prime - proj_{d.conv_y} p\prime||}{||c\prime - proj_{d.conv_y} c\prime||}. \tag{6}$$

The numerator in 6 is the closest distance between the parent solution (p) to the convergence direction ($c_r - p_r$). While, the denominator in 6 is the closest distance between the offspring solution (c) to the convergence direction ($c_r - p_r$).

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

$$\begin{aligned} p\prime &= p - p_r, \\ c\prime &= c - c_s, \end{aligned} \tag{7}$$

with p_r and c_s being the parent, and offspring objective vectors used to calculate the convergence direction in equation 10. 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.

$$proj_{d.conv_y} p\prime = \frac{d.conv_y \cdot p\prime}{|p\prime|^2} p\prime. \tag{8}$$

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

$$||p\prime - proj_{d.conv_y} p\prime|| = \text{"spectral" norm}(p\prime - proj_{d.conv_y} p\prime) \tag{9}$$

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

$$d.conv_y = c_j - p_h \tag{10}$$

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

$$D_c = \{d | \exists c_d \prec p_k, k \in 1, \ldots, N, d \in [1, \ldots, |C|]\} z \tag{11}$$

N is the parent population size, $|C|$ is the size of the offspring population C. In equation 11, 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.

$$h = \underset{k \in D_p}{argmin} ||p_k - c_j|| \tag{12}$$

$$D_p = \{k | \exists c_j \prec p_k, k \in 1, \ldots, N\} \tag{13}$$

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

D_p in equation 13 denotes the index set of parent solutions which are weakly dominated by c_j (j index comes from equation 11).

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 [17].

3.4 Priority Function - Random

Finally, we describe the last priority function studied in this work. The random priority function is used a 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 "spectral" norm, MRDL, relative improvement and random. To answer that question we apply MOEA/-DE with the 5 priority functions to the two artificial benchmark problems and one real-world problem.

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 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.

4.3 Parameter Settings

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.

5 RESULTS

First we give the overall results, and then we discuss in more specific details given the group pf benchmark problems.

5.1 Overall 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.

For all groups of functions, "spectral" norm had the best results given the median of HV or IGD, with except for the UF-1 and Uf-6 functions. This priority function had outstanding results in the Lunar Landing Problem. MRDL as a priority function had not improved much the performance of MOEA/D-DE, in terms of median values of HV or IGD, by the clear exception of the Lunar Landing Problem, although with no statistical difference. The results of the MRDL priority function only lead to a statistical improvement in one function, UF-5, for both metrics values while in the UF-4, for the HV value, it showed the best value (the same as the others priority functions). Relative improvement performed the best when comparing with MOEA/D-DE with statistical difference from MOEA/D-DE in all but the Lunar Landing problem. In most cases, given both metrics values, it had the best median results.

5.2 UF benchmark functions

Figure **??** (b) and (c) show box-plot that exemplify the results found in the UF benchmark functions in terms of the HV values while Figure 3 (a) and (b) does the same but in terms of the IGD values. In it we can see that MRDL as a priority function is slightly better than MOEA/D-DE in terms of median and standard deviation. Figures 2 (b) and (c) and 4 (a) and (b) show how the values of the HV and IGD evolve over the iteractions of the algorithms. For the UF-3 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 "spectral" 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 "spectral norm".

Turning to the results of the Tables 1 and 2, we discuss the results of every priority function. The "spectral" 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. We highlight that the result of the "spectral" norm in the function UF-5 were clearly better than the other results. 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 all but in the UF-3 benchmark function it had the best result, being significantly different from the MOEA/D-DE in every single case. The results of the relative improvement in terms of median values of the IGD values was not as remarkable, but still very good.

From the both the IGD- HV- evolution graph it has a strange regressive peak, while it seems like none of the variants had converged.

5.3 DTLZ benchmark functions

5.4 Lunar Landing Problem

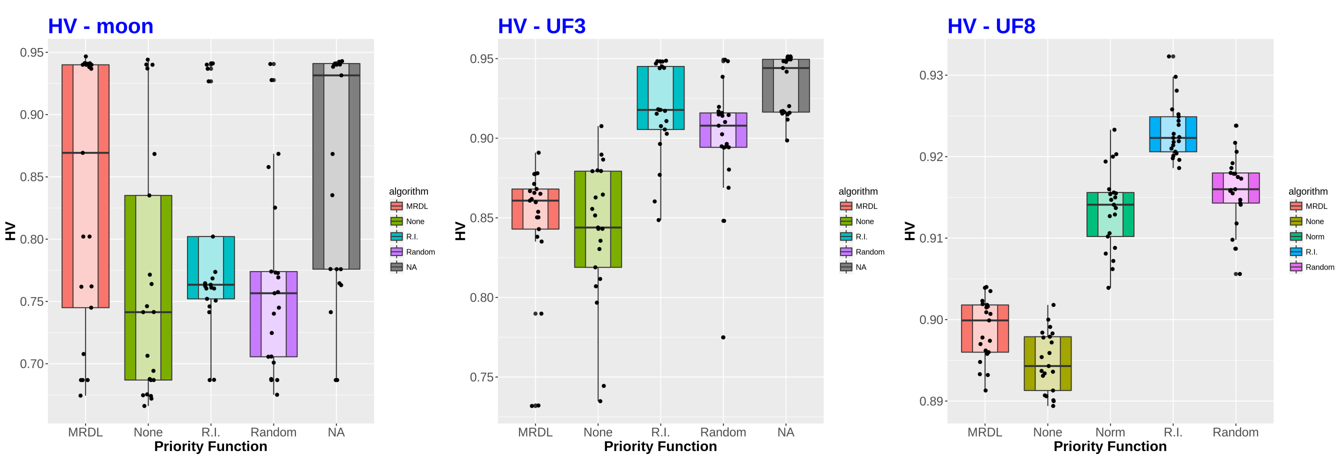
Figure **??** (a) show box-plot the results found in the Lunar Landing benchmark problem in terms of the HV values. Both priority functions related to diversity, MRDL and "spectral" norm, improved a lot the performance of the MOEA/D-DE. On the other hand, the relative improvement lead to a diminish of the standard deviation, with just a slight improvement on the performance. Figure 2 (a) 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), using around half of the total number evaluations (70000) to converge.

PF:	None	MRDL	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	0.86 (0.02)	0.86 (0.01)	0.83 (0.02)	**0.88 (0.01)**∗	**0.88 (0.01)**∗
UF2	0.75 (0.01)	0.75 (0.01)	0.76 (0.01)∗	**0.77 (0.08)**∗	**0.77 (0.01)**∗
UF3	0.84 (0.04)	0.86 (0.04)	**0.94 (0.02)**∗	0.92 (0.03)∗	0.91 (0.04)∗
UF4	0.36 (0.01)	**0.37 (0.01)**	**0.37 (0.01)**∗	**0.37 (0.01)**∗	**0.37 (0.01)**∗
UF5	0.63 (0.02)	0.66 (0.02)∗	0.75 (0.03)∗	**0.81 (0.01)**∗	**0.81 (0.02)**∗
UF6	0.66 (0.02)	0.66 (0.01)	0.66 (0.02)	**0.69 (0.01)**∗	**0.69 (0.01)**∗
UF7	0.80 (0.01)	0.80 (0.01)	0.82 (0.01)∗	**0.84 (0.01)**∗	0.83 (0.01)∗
UF8					
UF9					
UF10					

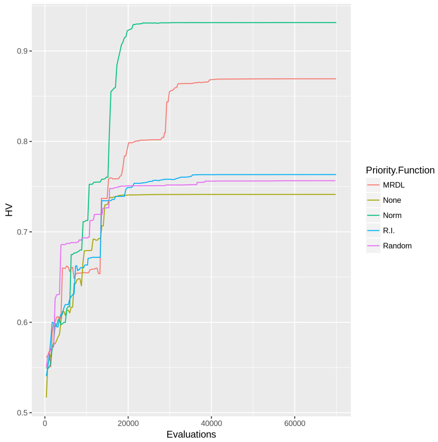
Table 1: HV Results - Lunar problem: The best algorithm is the one that uses Norm. It is the only one that is stats different from None, although by median all priority functions are better. Highlighted are the best values found.Star ∗ means stat diff from MOEA/D-DE without priority function. SD Values smaller than 0.01 were truncated to that
for UF the results are

PF:	None	MRDL	Norm	R.I.	Random
UF1	0.14 (0.01)	0.13 (0.01)	0.10 (0.02)∗	**0.09 (0.01)**∗	**0.09 (0.01)**∗
UF2	0.08 (0.01)	0.08 (0.01)	**0.06 (0.01)**∗	**0.06 (0.01)**∗	**0.06 (0.01)**∗
UF3	0.26 (0.01	0.26 (0.01)	**0.17 (0.02)**∗	0.18 (0.03)∗	0.21 (0.03)∗
UF4	0.10 (0.01)	0.10 (0.01)	**0.09 (0.01)**∗	**0.09 (0.01)**∗	**0.09 (0.01)**∗
UF5	1.75 (0.08)	1.65 (0.09)∗	**0.97 (0.06)**∗	1.05 (0.06)∗	1.08(0.07)∗
UF6	0.12 (0.03)	0.12 (0.02)	0.10 (0.07)∗	**0.08 (0.01)**∗	**0.08 (0.02)**∗
UF7	0.12 (0.02)	0.13 (0.01)	**0.06 (0.01)**∗	0.07 (0.01)∗	0.07 (0.01)∗
UF8					
UF9					
UF10					

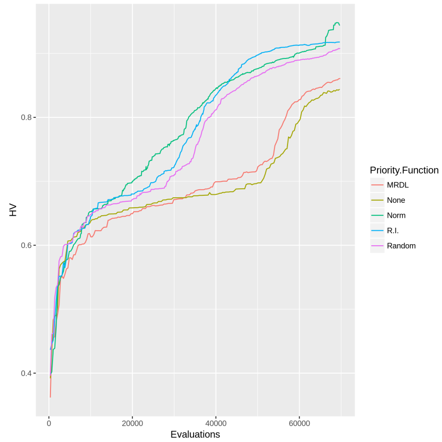
Table 2: Highlighted are the best values found.Star ∗ means stat diff from MOEA/D-DE without priority function. SD Values smaller than 0.01 were truncated to that
for UF the results are



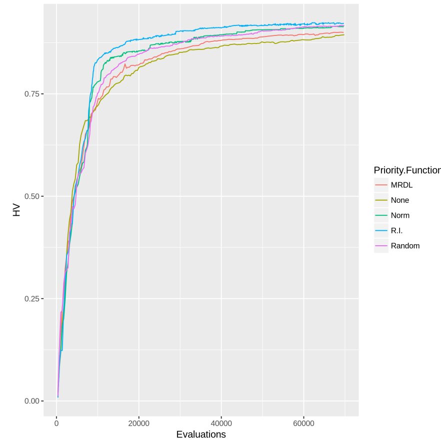
(a) HV values of the last iteraction on the Lunar Landing Problem **(b) HV values of the last iteraction on the UF-3 function Problem** **(c) HV values of the last iteraction on the UF-8 function Problem**



(a) Evolution of the HV on the Lunar Landing

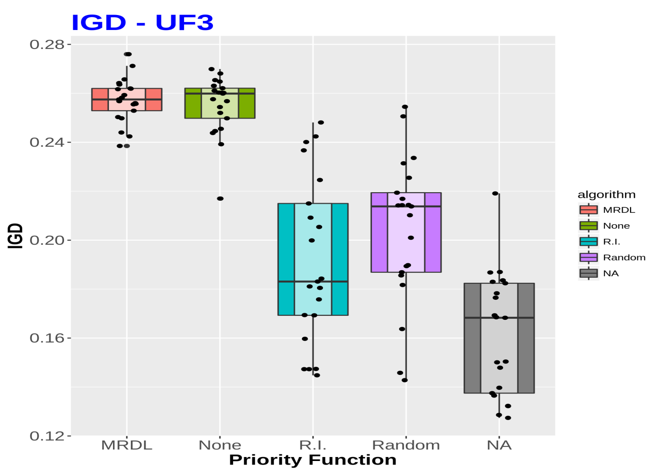


(b) Evolution of the HV on the UF3

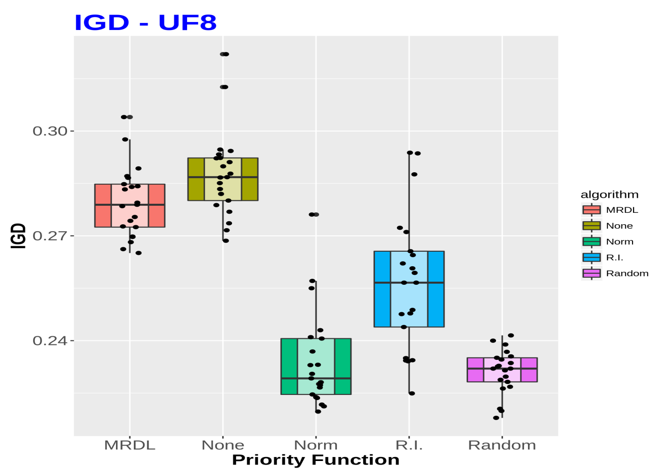


(c) Evolution of the HV on the UF8

Figure 2: SBX crossover - (λ, λ) scheme.



(a) IGD values of the last iteraction on the UF-3 function



(b) IGD values of the last iteraction on the UF-3 function

Figure 3: SBX crossover - (λ, λ) scheme.

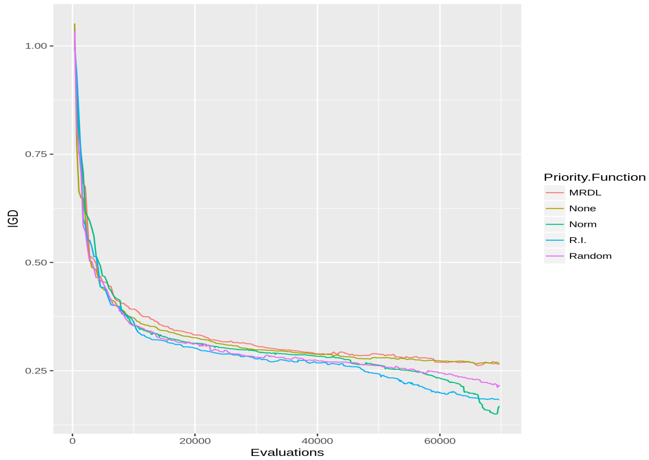
Changing to the results of the Table 1, we discuss the results of the priority functions. We highlight that the result of "spectral" norm since it was clearly better than the other priority function results in terms of median of HV values as it is statistically different from MOEA/D-DE. Although the results of MRDL were not as good as the previous one, it is still has 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.

6 CONCLUSION

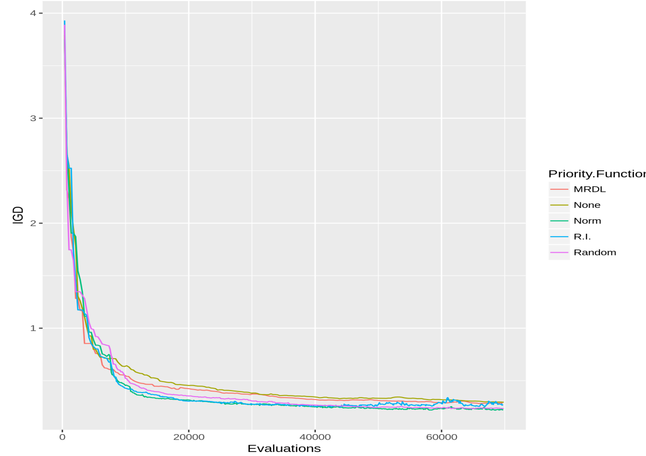
Using priority functions do help

Using norm helps more

We expected that MRDL would help a lot, but it barely helped. Maybe more iteractions are needed.



(a) Evolution of the IGD on the UF3



(b) Evolution of the IGD on the UF8

Figure 4: SBX crossover - (λ, λ) scheme.

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, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1 (2011), 32–49.

[17] Aimin Zhou and Qingfu Zhang. 2016. Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 52–64.

[18] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*. Springer, 292–301.

[19] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.