

---

# Faster Convergence in Multi-Objective Optimization Algorithms Based on Decomposition

**Yuri Lavinas**

lavinas.yuri.xp@alumni.tsukuba.ac.jp

School of Systems and Information Engineering, University of Tsukuba, Japan

**Marcelo Ladeira**

mladeira@unb.br

Department of Computer Science, University of Brasilia, Brazil

**Claus Aranha**

caranha@cs.tsukuba.ac.jp

School of Systems and Information Engineering, University of Tsukuba, Japan

---

## Abstract

The Resource Allocation approach (RA) improves the performance of MOEA/D by maintaining a big population and updating few solutions each generation. However, most of the studies on RA generally focused on the properties of different Resource Allocation metrics. Thus, it is still uncertain what the main factors are that lead to increments in performance of MOEA/D with RA. This study investigates the effects of MOEA/D with the Partial Update Strategy in an extensive set of MOPs to generate insights into correspondences of MOEA/D with the Partial Update and MOEA/D with small population size and big population size. Our work undertakes an in-depth analysis of the populational dynamics behaviour considering their final approximation Pareto sets, anytime hypervolume performance, attained regions and number of unique non-dominated solutions. Our results indicate that MOEA/D with Partial Update progresses with the search as fast as MOEA/D with small population size and explores the search space as MOEA/D with big population size. MOEA/D with Partial Update can mitigate common problems related to population size choice with better convergence speed in most MOPs, as shown by the results of hypervolume and number of unique non-dominated solutions, the anytime performance and Empirical Attainment Function indicates.

## Keywords

MOEA/D, Resource Allocation, population dynamics, partial update strategy.

## 1 Introduction

Multi-objective Optimisation Problems (MOPs) are problems with two or more conflicting objective functions that are optimised simultaneously. The Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) is a popular and efficient algorithm for finding good sets of trade-off solutions for MOPs ([Zhang and Li \(2007\)](#)). The key idea of MOEA/D is to create a linear decomposition of the multi-objective optimisation problem into a set of single-objective sub-problems. These sub-problems are solved in parallel. Each individual of the population is assigned to be the “incumbent” solution of a sub-problem. In the original MOEA/D, all sub-problems are considered to be equally important at any time in the optimisation process. However, more recent research has indicated that the MOEA/D may waste computational effort by improving

specific incumbent solutions that are not very promising (Bezerra et al. (2015)). Treating all sub-problems with the same importance can be a critical issue, especially in problems where the computation cost of the fitness function is high and the evaluation budget is low (Kohira et al. (2018)).

To address this issue, several works have investigated methods to allocate different amounts of computational effort to sub-problems (Zhang et al. (2009); Zhou and Zhang (2016); Lavinas et al. (2019a,b); Wang et al. (2019); Pruvost et al. (2020); Lavinas et al. (2020)). The general approach is to use different indicators (priority functions) to determine how many fitness evaluations are allocated to each sub-problem. At each iteration of the algorithm, only a subset of the population is updated and evaluated, selected according to the priority function used. We call the selected individuals on each iteration the *working population*. These approaches are collectively known as Resource Allocation (RA) and have resulted in consistent performance improvements for the MOEA/D.

Existing RA research placed considerable importance on the properties of different priority functions. However, recent studies from Pruvost et al. (2020), and Lavinas et al. (2020) have found that the improvements in performance obtained from RA could be attributed to the reduction in the size of the working population. These results raise the question: are the improvements from RA due only to the reduced population size, or are there other factors also at work in the improvement observed in the various RA approaches? There is no research so far studying the population dynamics of MOEA/D with RAs.

To fill in this gap, we investigate and quantify the effects of RA on MOEA/D against MOEA/D with different population sizes on continuous problems. To achieve this, we study the correspondences of MOEA/D using RA against (1) no RA on small population size and (2) no RA on larger population size. We perform an experimental investigation and systematically analyse the impact of the proportion parameter  $n$  on the performance of the MOEA/D on 70 different MOPs. This experimental work presented here provides one of the first investigations into how MOEA/D with RA can outperform the original variant, even considering different population sizes.

Moreover, this work aims to understand better the mechanics of MOEA/D regarding the population update. This behaviour suggests using RA techniques to mitigate common problems related to the choice of population size. Overall, our results indicate that MOEA/D with RA shows a hybrid behaviour in our experiments: MOEA/D with RA converges to the Pareto front at about the same speed as MOEA/D with a small population while exploring the search space as MOEA/D with a large population does. These results motivate the use of RA to mitigate common problems related to population size choice, independently of the MOP in question. Together these results provide important insights into the new population dynamics introduced by MOEA/D with RA that accelerates the convergence speed with additional improvements in the number of non-dominated solutions on the final approximation of the Pareto Front.

The remainder of this paper is organised as follows: Section 2 reviews the main concepts related to Resource Allocation in the MOEA/D. Section 3 explains the relationship between population size and Resource Allocation in MOEA/D. Section 4 describes the MOEA/D with partial update strategy (PS). Section 5 describes the experimental settings related to the investigation of the effect of partial updates on the performance of the algorithm, while Section 6 presents the results of such investigation and comparisons between MOEA/D-PS against two different configurations of MOEA/D, one with small population size and another with big population size. Fi-

nally, Section 7 presents our concluding remarks.

## 2 Resource Allocation

The key idea behind MOEA/D is to decompose a MOP into a set of single-objective sub-problems, which are then solved simultaneously. While these sub-problems are usually considered equivalent, a growing body of work indicates that prioritising some sub-problems at specific points of the search can improve the performance of MOEA/D. Techniques that address this issue are collectively called Resource Allocation (RA) techniques.

Priority functions are used in RA to determine preferences between sub-problems. These functions take information about the progress of the search and return priority values that are then used to change the distribution of computational resources among sub-problems (Cai et al. (2015)). It is also possible to design priority functions that target other properties of the optimisation process (Lavinias et al. (2019a)), such as diversity or robustness (Goulart et al. (2017)).

### 2.1 State of the art

Much of the research on Resource Allocation has used the Relative Improvement (RI) as a priority function, with some modifications on other aspects of the algorithm (Zhang et al. (2009); Nasir et al. (2011)). Although Zhou et al. did expand the discussion over Resource Allocation in their work (Zhou and Zhang (2016)), few other works have studied Resource Allocation in depth.

In previous works, we isolated the priority function as a point of investigation (Lavinias et al. (2019a,b)). The goal in these works was to improve the performance of MOEA/D based on the choice of priority function and to understand further the behaviour of MOEA/D under different Resource Allocation approaches. We introduced three new priority functions (Decision-Space Diversity, inverted Decision-Space Diversity and one based on random values). The experimental comparisons of these works revealed this surprising result: the performance of MOEA/D when using a random Resource Allocation method is as competitive as RI and better than not using Resource Allocation at all. This result suggested that MOEA/D may benefit simply from the increased populational inertia (possibly due to slower diversity loss) resulting from holding portions of the population constant during any given iteration.

Later, we investigated this question, using the Partial Update Strategy for the MOEA/D based on priority values sampled from a uniform distribution, leading to a random selection of sub-problems to update at every iteration (Lavinias et al. (2020)). The Partial Update Strategy allows the control of the number of sub-problems modified at any given iteration and regulates the working population dynamics of the MOEA/D. The results reveal two main findings, and the first one is a strong association between more conservative updating of the MOEA/D population and improved performance. The second main finding is that MOEA/D benefits more from having slower population dynamics than from a specific prioritisation of sub-problems. These findings are in agreement with the results observed in the combinatorial domain by Pruvost et al. (2020). Therefore, we understand it is imperative to investigate the correspondences of MOEA/D with the Partial Update Strategy and the populational dynamics of MOEA/D with large and small population sizes.

### 3 The Importance of Population Size

The population size is one of the essential parameters of Evolutionary Algorithms since it influences those algorithms' dynamics during the execution (Glasmachers et al. (2014)). While the right choice can lead to considerable improvements in the algorithm's performance, a wrong choice of population size might have the contrary effect (Glasmachers et al. (2014); Pruvost et al. (2020)). For example, choosing a population size that is too small can cause premature convergence because a small population size might prevent the localization of optimal solutions. However, a bigger population size may cause the algorithm to waste computational resources, a critical issue in computationally costly problems (Kohira et al. (2018)). Furthermore, the proper choice of population size depends on the characteristics of different problems, such as the difficulty of the problem, the presence of multiple local-optima and the shape of the Pareto Front (Črepinšek et al. (2013); Glasmachers et al. (2014); Pruvost et al. (2020)).

In MOEA/D, the population size also defines the number of sub-problems and the regions of the decision space where the sub-problems are established. Therefore, the choice of population size is even more critical for this algorithm. Since different decision space regions might have different characteristics, each sub-problem might display different characteristics in the same MOP. For example, given a MOP and two different sub-problems, while multiple local-optima might influence one sub-problem, another sub-problem might not have to deal with such difficulty.

Considering all of this evidence, it is clear that choosing the right population size, especially in MOEA/D, is a delicate matter. That said, we reason that Resource Allocation can mitigate the burden of choosing the right population size in MOEA/D. That is because MOEA/D with Resource Allocation techniques benefits from maintaining a big population size and updating very few solutions at each iteration (Pruvost et al. (2020); Lavinas et al. (2020)), motivating the use of RA in MOEA/D. In other words, MOEA/D can maintain and improve a substantially large population at a lower cost than the original MOEA/D, the cost of updating those few selected solutions because RA techniques allow MOEA/D to update only a small subset of solutions from the whole larger population. By contrast, without RA techniques, a large population aggravates the computational burden (Li et al. (2015)), slows down the algorithm or introduces many non-useful sub-problems (Jain and Deb (2014)). Consequently, in this work, we aim to verify and understand the mechanics of MOEA/D regarding the population update with Resource Allocation techniques.

### 4 The Partial Update Strategy

To verify and understand the mechanics of MOEA/D regarding the population update and investigate the extent of this effect, we use the Partial Update Strategy (PS), one of the most general Resource Allocation techniques. The PS selects  $n$  individuals randomly to update each iteration, which is equivalent to a priority function sampled from a uniform distribution. In addition to that, the PS strategy always includes the boundary weight vectors' solutions, one for each objective. For example, there are two boundary weight vectors in a two objective MOP; thus, the number of sub-problems selected is  $n + 2$ .

The reason for always selecting the boundaries is because they have an impact on the coordinates of the reference point  $z^*$  used by the scalarizing function (Wang et al. (2019)). Algorithm 1 details the pseudocode of the MOEA/D-DE (Zhang et al. (2009)), using the Partial Update Strategy (MOEA/D-PS). Moreover, in a preliminary experi-

ment, we saw a small (non-statistically significant) difference in performance favouring the case where the boundaries are always selected over the case where all vectors are selected randomly. Finally, we did not consider the case where the same set of the weight vectors is always used in all iterations, because [Zhou and Zhang \(2016\)](#) showed an in-depth experiment that indicates that pre-specified vectors lead to worse performance to vectors specified on the run.

---

**Algorithm 1** MOEA/D-PS (MOEA/D-DE with the Partial Update Strategy)
 

---

```

1: Input: number of subproblems to select  $n$ , termination criteria, number of objectives  $m$ .
2: Initialize MOEA/D-DE variables (e.g. weight vectors, set of solutions, etc.)
3: while Termination criteria are not meet do
4:   for  $i = 1$  to the number of subproblems do
5:      $u_i \leftarrow \text{rand}()$  ▷ Vector of random values
6:   end for
7:   Select  $n + m$  subproblems:  $n$  with the highest  $u_i$  value and  $m$  number of boundary subproblems.
8:   for  $j = 1$  to  $n$  do ▷ Number of subproblems selected
9:     Generate new candidate  $y$  for subproblem  $j$ .
10:    Evaluate these new candidate solutions.
11:   end for
12:   Update the whole set of solutions with the  $n$  new solutions generated in steps 8-11.
13: end while

```

---

Notice that the standard MOEA/D and other variants such as MOEA/D-DE can be instantiated from Algorithm 1 by setting  $n = \text{total number of sub-problems}$ . We highlight that this is only possible because we are using the generational version of MOEA/D. The only difference that the Partial Update Strategy introduces in the base algorithm is that only a few sub-problems are updated at any given iteration, regulated by the value of  $n$ .

Other Resource Allocation techniques could also be expressed with this structure by modifying the priority value attribution function in Line 5 of the algorithm. It is relevant to observe that sub-problems that are not selected by the Partial Update Strategy at a given iteration may still have their incumbent solutions updated. In other words, updating only small parts of the population using Resource Allocation in MOEA/D-PS affects only the variation step, not the replacement one. Thus, sub-problems not selected for variation may receive new candidate solutions, e.g., generated for a neighbouring sub-problem.

#### 4.1 Computational Complexity of MOEA/D-PS

Here, we describe the additional computational complexity for allocating the computing resources in one iteration. On Line 5 of Algorithm 1, the priority value attribution has a complexity of  $O(N)$ , where  $N$  is the population size. Similarly, the complexity for selecting sub-problems, on Line 7, is also  $O(N)$ . Thus, the additional complexity of one iteration caused by the Partial Update Strategy is  $O(N)$ , which is much less than the complexity of MOEA/D itself. While the reduction in the number of updated solutions increases the total number of generations, this reduction also decreases the number of evaluations per iteration. Thus, the total number of evaluations is kept exactly the

same.

According to this analysis, we can conclude that the additional computational complexity of MOEA/D-PS is minimal and that any increment in the performance of MOEA/D when using the Partial Update Strategy comes at a low cost.

## 5 Population Dynamics Behaviour Experiment

We carry out the following experimental study to investigate the relationship between Resource Allocation, represented by MOEA/D-PS, and population size in practice. We compare the final approximation sets' quality based on (a) the final approximation hypervolume and proportion non-dominated of the final approximation sets, (b) the hypervolume anytime performance, (c) the empirical attainment function and (d) the statistical analysis based on the Wilcoxon Rank Sum Tests. This experiment aims to show the hybrid behaviour of MOEA/D-PS, with shared characteristics of both MOEA/D configurations studied here.

### 5.1 Benchmark Problems

Four different benchmark sets were used: (1) the scalable DTLZ set (Deb et al. (2005)), with 2 objectives, (2) the inverted scalable DTLZ benchmark ( $DTLZ^{-1}$ ) set (Ishibuchi et al. (2016)), with 2 objectives, (3) the UF set (Zhang et al. (2008)), with 2 and 3 objectives, and finally, (4) the Black-Box Optimization Bi-Objective Benchmark test functions (Tusar et al. (2016)). For all functions in all benchmark sets, we set the number of dimensions to  $D = 40$ <sup>1</sup>. The implementation of the test problems available from the *smoof* package (Bossek (2017)) was used in all experiments<sup>2</sup>.

1. The four DTLZ Benchmark functions we use have the following features (Huband et al. (2006)):
  - DTLZ1: Linear Pareto Front - unimodal;
  - DTLZ2: Concave Pareto Front - unimodal;
  - DTLZ3: Concave Pareto Front - multimodal;
  - DTLZ4: Concave Pareto Front - unimodal.
2. The features of the four inverted DTLZ benchmark ( $DTLZ^{-1}$ ) set (Ishibuchi et al. (2016)), are:
  - $DTLZ1^{-1}$ : Linear Pareto Front;
  - $DTLZ2^{-1}$ : Convex Pareto Front;
  - $DTLZ3^{-1}$ : Convex Pareto Front;
  - $DTLZ4^{-1}$ : Convex Pareto Front.
3. The UF Benchmark set is composed of ten unconstrained test problems with Pareto sets designed to be challenging to existing algorithms (Li et al. (2019)). Problems UF1-UF7 are two-objective MOPs, while UF8-UF10 are three-objective problems (Zhang et al. (2008)).

<sup>1</sup>This is the maximum value one can use in the Black-Box Optimization Bi-Objective Benchmark test. For consistency, we keep this value fixed in all problems.

<sup>2</sup>For the inverted DTLZ benchmark ( $DTLZ^{-1}$ ) we modify the DTLZ implementation from the *smoof* package in order to invert the problem set.



- UF1: Convex Pareto Front - multimodal;
  - UF2: Convex Pareto Front - multimodal;
  - UF3: Convex Pareto Front - multimodal;
  - UF4: Concave Pareto Front - multimodal;
  - UF5: Linear Pareto Front - multimodal;
  - UF6: Linear Pareto Front - multimodal;
  - UF7: Linear Pareto Front - multimodal;
  - UF8: Concave Pareto Front - multimodal;
  - UF9: Linear and discontinuous Pareto Front - multimodal;
  - UF10: Concave Pareto Front - multimodal.
4. Finally, we list the problems features of the Black-Box Optimization Bi-Objective Benchmark and then we describe the meaning of the features of these problems. We select all 55 two-objectives problems from this test suit.
- F1, F2, F11: separable - separable;
  - F3, F4, F12, F13: separable - moderate;
  - F5, F6, F14, F15: separable - ill-conditioned;
  - F7, F8, F16, F17: separable - multi-modal;
  - F9, F10, F18, F19: separable - weakly-structured;
  - F20, F21, F28: moderate - moderate;
  - F22, F23, F29, F30: moderate - ill-conditioned;
  - F24, F25, F31, F32: moderate - multi-modal;
  - F26, F27, F33, F34: moderate - weakly-structured;
  - F35, F36, F41: ill-conditioned - ill-conditioned;
  - F37, F38, F42, F43: ill-conditioned - multi-modal;
  - F39, F40, F44, F45: ill-conditioned - weakly-structured;
  - F46, F47, F50: multi-modal - multi-modal;
  - F48, F49, F51, F52: multi-modal - weakly structured;
  - F53, F54, F55: weakly-structured - weakly-structured.

Here, we give a simple explanation of the features in the problems above. The first feature is separability. A separable function does not show any dependencies between the variables. The next features are the moderate-conditioned and ill-conditioned, and they indicate how much changing a solution impacts the objective value ([Hansen et al. \(2011\)](#)). If the impact is high, we have an ill-conditioned function; otherwise, we have a moderate-conditioned function. A multi-modal function has at least two local optima, and finally, a weakly-structured function is a function that with a very unclear general structure ([Finck et al. \(2010\)](#)).

## 5.2 Experimental Parameters

We used the MOEA/D-DE parameters as they were introduced in the work of Li and Zhang (Li and Zhang (2009)) in all tests. All objectives were linearly scaled at every iteration to the interval  $[0, 1]$ , and the Sobol decomposition (Zapotecas-Martínez et al. (2015)) function was used. Table (1) summarizes the experimental parameters. Details of these parameters can be found in the documentation of package MOEADr and the original MOEA/D-DE reference (Zhang et al. (2009); Campelo and Aranha (2018); Campelo et al. (2020)).

We highlight the choice of the  $n$  parameter value of the Partial Update Strategy. This choice is based on our previous work (Lavinas et al. (2020)), where the influence of different  $n$  values on the performance of the MOEA/D-PS is investigated. The results demonstrated that low values are associated with (anytime) performance improvements, especially when  $n$  is close to 10% of the population size. Therefore, in this work, we set  $n$  equal to 10% of the population size, meaning a value of  $n = 50$ , since the population size is set to 500<sup>3</sup>.

Recently, it was pointed out that during the search, multi-objective algorithms might discard non-dominated solutions in favour of dominated ones (Li and Yao (2019)). In consequence, the final population of EMO algorithms might maintain a subset of solutions that are dominated by other solutions already discarded. In another work (Ishibuchi et al. (2020)), it was shown that a better final solution set, in comparison with the final population, can be obtained if this final solution set is constructed considering solutions from many, if not all, iterations. Therefore, we understand that such good solutions found by MOEA/D during the run might be excluded from the final population, a factor that is even more noticeable when MOEA/D maintains a small population size. To alleviate such problem, we use a bounded external archive with the same size as the population in use.

## 5.3 Experimental Evaluation

As indicated previously, we use the following criteria to compare the results of the different strategies: (a) final approximation hypervolume (HV) and proportion non-dominated solutions, (b) anytime performance, and (c) empirical attainment performance (EAP). For the calculation of the hypervolume, the objective function approximation values were scaled to the  $(0, 1)$  interval, with the reference point set to the nadir point:  $(1, 1)$  for two objective problems and  $(1, 1, 1)$  for three objective problems. We scale the approximation values using the range of objective values found by all algorithms in the comparisons. There is some concern that using a reference point equal to the nadir point may cause problems when calculating the hypervolume. However, because all problems have two or three objectives and the population size is large, this is not a serious issue for this experiment.

The hypervolume can be defined as the volume of the  $n$ -dimensional polygon formed by some reference point,  $\vec{x}_{\text{ref}} = (x_1, x_2, \dots, x_m)$ , and a finite set  $S = (x_1, x_2, \dots, x_m)$  of solutions in the positive orthant,  $\mathbb{R}_{\geq 0}^n$ , with  $m$  being the number of objectives (Beume et al. (2009)). That is:

$$HV(S) = \text{Hypercube calculate between } S \text{ and } \vec{x}_{\text{ref}}. \quad (1)$$

<sup>3</sup>The number of selected sub-problems is equal to  $50 + m$ ,  $m$  being the number of objectives of the MOP being solved.



Table 1: Experimental parameter settings.

Parameters	Value
DE mutation parameter	$F = 0.25$
Polynomial mutation parameters	$\eta_m = 20$ $p_m = 0.01$
Restricted Update parameter	$nr = 2$
Locality parameter	$\delta_p = 0.9$
Neighborhood size	$T = 20\%$ of the pop. size
Bounded External Archive	TRUE
Decomposition method	Value
Sobol <a href="#">Zapotecas-Martínez et al. (2015)</a>	50 or 500 weights
Population size	Value
MOEA/D-PS	500
MOEA/D with big pop.	500
MOEA/D with small pop.	50
Resource allocation parameter	Value
$n$	50 + the boundary weight vectors
Experiment Parameters	Value
Repeated runs	10
Computational budget	100000 evals.

### Final Approximation: Hypervolume and Proportion Non-dominated Solutions

We compare the final approximation hypervolume results of the algorithms in terms of hypervolume and the proportion non-dominated solutions (for both, higher is better). The proportion non-dominated solutions provides insights into how diverse is a set of solutions. The higher number of non-dominated solutions in a given set suggests a more diverse set of solutions in the objective space.

For a fair comparison, in special for the hypervolume results, the algorithms are evaluated based on an archive of  $size = 500$ . For MOEA/D-PS and MOEA/D with the big population, the archive is just the final population, but for MOEA/D with the small population, the archive accumulates solutions of different iterations. In specific, given that the small population is of size 50, the archive of this variant combines solutions from 10 iterations, lead to an archive of  $size = 50 * 10 = 500$ .

**Anytime Hypervolume Performance** Analysing MOP solvers using the final approximation provides limited information related to these algorithms' performance since any MOP should return a suitable set of solutions at any time during the search. Thus, it is imperative to extend the performance analysis because good solutions should be found as earlier as possible independently of the termination criterion or if the execution is interrupted during the search ([Zilberstein \(1996\)](#); [Radulescu et al. \(2013\)](#); [Dubois-Lacoste et al. \(2015\)](#); [Tanabe et al. \(2017\)](#)). Here, we analyse the anytime performance effects in terms of HV values to investigate the impact of different population sizes in MOEA/D against MOEA/D-PS. As for the final approximation analysis, we use an archive of size 500 for all algorithms.

**Empirical Attainment Performance** The empirical attainment function (EAF) allows the examination of the solution many sets of different runs of an algorithm, and it can illustrate where and by how much the outcomes of two algorithms differ in the objective space ([López-Ibáñez et al. \(2010\)](#)). The EAF is based on the attainment surface and rep-

resents the probability that an arbitrary objective region in the objective space is attained (that is, dominated or equal) by an algorithm and probability can be estimated using data collected from several independent runs of such algorithm. The attained surface separates the objective space into two regions: (1) where the objective space is dominated (attained) by solutions of many sets and (2) where the objective space is not dominated by those solutions (Fonseca and Fleming (1996); Da Fonseca et al. (2001)). For example, the median attainment surface shows regions dominated by at least half of the runs (López-Ibáñez et al. (2010)).

**Statistical Analysis** We also evaluate the HV performance statistically in terms of the final (archive) population. The techniques' differences are analysed using Wilcoxon Rank Sum Tests, with a significance level of  $\alpha = 0.05$  and Hommel adjustment for multiple comparisons.

#### 5.4 Reproducibility

For reproducibility purposes, all the code and experimental scripts are available online at <https://github.com/yurilavinas/MOEA/Dr/tree/ECJ>. Furthermore, the data for all experiments is available at <https://zenodo.org/record/5551197#.YV6EE6CRVhf>.

## 6 Results

The following Section of this paper moves on to describe in greater detail the experimental results in terms of (a) the final approximation hypervolume and the proportion non-dominated of the final approximation sets, (b) the hypervolume anytime performance, (c) the empirical attainment function and (d) the statistical analysis based on the Wilcoxon Rank Sum Tests. These analytical procedures and the results obtained from them are described in the previous Section 5.

### 6.1 Final Approximation Results

Table 2 shows, on the left side, the mean results obtained by the MOEA/D-PS with  $n = 50$  and MOEA/D without Resource Allocation with (1) small population size and (2) big population size for all test problems. We recall that for fair of comparison, we use an archive population of  $size = 500$  for all algorithms, see Section 5.3.

Furthermore, looking at the proportion of non-dominated solutions in the right side of Table 2, we can see that updating a subset of solutions from a larger population at each iteration resulted in the highest proportion of non-dominated solutions on all MOPs. The main exception is on the UF benchmark set, where MOEA/D with a big population has a higher proportion non-dominated solutions (but not the highest proportion and duplicated non-dominated solutions). This higher overall number of non-dominated solutions indicates that the approximation set is much more diverse compared to the approximation sets of the traditional MOEA/D, independently of the population configuration. It is clear from both the hypervolume and the number of non-dominated solutions results that MOEA/D-PS that has a similar performance with a more diverse set of solutions in comparison with MOEA/D with big population size and better than MOEA/D with small population size.<sup>4</sup>

<sup>4</sup>From results (available in the GitHub repository) obtained from our experiments we saw that MOEA/D-PS tends to group some solutions in the same region of the search space. Yet, in most MOPs studied here, this MOEA/D variant is still able to find more unique solutions than MOEA/D with a big population. We also noted that the number of unique solutions tends to decrease with the difficulty of the problem in question.

Table 2: Means (standard errors) for HV and proportion of non-dominated solutions, for each problem. Best results are highlighted in bold.

MOP	HV			NNDOM		
	MOEA/D-PS	Big population	Small population	MOEA/D-PS	Big population	Small population
DTLZ1	<b>1.00 (0.01)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	0.13 (0.08)	0.66 (0.25)
DTLZ2	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.02)</b>	0.95 (0.02)	0.92 (0.05)
DTLZ3	<b>1.00 (0.01)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.11 (0.13)	0.80 (0.20)
DTLZ4	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.02)</b>	0.88 (0.05)	0.91 (0.05)
DTLZ1 <sup>-1</sup>	0.44 (0.02)	0.35 (0.01)	<b>0.46 (0.01)</b>	<b>1.00 (0.00)</b>	0.29 (0.04)	0.98 (0.02)
DTLZ2 <sup>-1</sup>	<b>0.78 (0.00)</b>	<b>0.78 (0.00)</b>	0.77 (0.00)	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>
DTLZ3 <sup>-1</sup>	<b>0.75 (0.01)</b>	0.60 (0.03)	<b>0.75 (0.01)</b>	<b>1.00 (0.00)</b>	0.17 (0.03)	0.96 (0.03)
DTLZ4 <sup>-1</sup>	<b>0.78 (0.00)</b>	<b>0.78 (0.00)</b>	0.77 (0.00)	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	0.99 (0.01)
UF1	0.94 (0.02)	<b>0.98 (0.00)</b>	0.95 (0.02)	<b>0.96 (0.04)</b>	0.37 (0.06)	0.54 (0.07)
UF2	0.92 (0.01)	<b>0.94 (0.01)</b>	<b>0.94 (0.01)</b>	<b>0.98 (0.03)</b>	0.38 (0.05)	0.54 (0.04)
UF3	0.91 (0.03)	<b>0.98 (0.00)</b>	0.96 (0.03)	<b>0.97 (0.04)</b>	0.24 (0.03)	0.53 (0.09)
UF4	0.48 (0.00)	<b>0.50 (0.00)</b>	0.47 (0.01)	<b>0.96 (0.04)</b>	0.56 (0.05)	0.73 (0.07)
UF5	0.90 (0.04)	<b>0.96 (0.01)</b>	0.90 (0.03)	<b>0.78 (0.20)</b>	0.20 (0.03)	0.44 (0.10)
UF6	0.92 (0.04)	<b>0.97 (0.00)</b>	0.93 (0.05)	<b>0.96 (0.04)</b>	0.31 (0.05)	0.50 (0.06)
UF7	0.92 (0.08)	<b>0.97 (0.00)</b>	0.94 (0.06)	<b>0.93 (0.21)</b>	0.48 (0.08)	0.61 (0.11)
UF8	0.99 (0.00)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.99 (0.01)</b>	0.64 (0.09)	0.79 (0.05)
UF9	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.01)</b>	0.60 (0.12)	0.70 (0.10)
UF10	<b>0.98 (0.01)</b>	<b>0.98 (0.00)</b>	0.97 (0.01)	<b>0.93 (0.08)</b>	0.46 (0.04)	0.73 (0.08)
F1	<b>0.96 (0.00)</b>	<b>0.96 (0.01)</b>	<b>0.96 (0.00)</b>	<b>0.99 (0.01)</b>	0.32 (0.06)	0.50 (0.04)
F2	<b>1.00 (0.01)</b>	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.29 (0.06)	0.54 (0.08)
F3	<b>0.98 (0.00)</b>	<b>0.98 (0.00)</b>	<b>0.98 (0.00)</b>	<b>1.00 (0.01)</b>	0.36 (0.05)	0.58 (0.04)
F4	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.25 (0.05)	0.47 (0.07)
F5	0.91 (0.01)	0.90 (0.01)	<b>0.93 (0.00)</b>	<b>0.99 (0.01)</b>	0.33 (0.03)	0.49 (0.06)
F6	0.99 (0.00)	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.25 (0.05)	0.51 (0.06)
F7	0.99 (0.00)	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.18 (0.02)	0.37 (0.07)
F8	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.99 (0.00)</b>	<b>0.89 (0.11)</b>	0.16 (0.01)	0.42 (0.11)
F9	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.31 (0.06)	0.54 (0.09)
F10	0.79 (0.03)	<b>0.80 (0.00)</b>	0.77 (0.05)	<b>0.99 (0.01)</b>	0.45 (0.07)	0.55 (0.05)
F11	<b>0.98 (0.00)</b>	<b>0.98 (0.00)</b>	<b>0.98 (0.00)</b>	<b>0.99 (0.01)</b>	0.31 (0.06)	0.55 (0.06)
F12	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.96 (0.04)</b>	0.29 (0.05)	0.62 (0.06)
F13	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.27 (0.05)	0.53 (0.07)
F14	0.95 (0.01)	0.95 (0.01)	<b>0.98 (0.01)</b>	<b>0.99 (0.02)</b>	0.30 (0.05)	0.48 (0.05)
F15	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.28 (0.04)	0.49 (0.05)
F16	0.99 (0.00)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.99 (0.01)</b>	0.17 (0.01)	0.41 (0.14)
F17	0.99 (0.01)	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.94 (0.05)</b>	0.15 (0.02)	0.39 (0.10)
F18	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.31 (0.04)	0.54 (0.09)
F19	0.93 (0.07)	<b>0.96 (0.02)</b>	0.93 (0.04)	<b>0.99 (0.02)</b>	0.40 (0.07)	0.62 (0.08)
F20	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.98 (0.02)</b>	0.43 (0.12)	0.66 (0.07)
F21	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.02)</b>	0.28 (0.06)	0.56 (0.07)
F22	0.92 (0.03)	<b>0.96 (0.01)</b>	<b>0.96 (0.02)</b>	<b>0.99 (0.02)</b>	0.33 (0.06)	0.55 (0.06)
F23	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.31 (0.04)	0.54 (0.04)
F24	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.96 (0.04)</b>	0.16 (0.02)	0.39 (0.07)
F25	0.99 (0.00)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.92 (0.10)</b>	0.14 (0.02)	0.32 (0.07)
F26	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.98 (0.02)</b>	0.37 (0.10)	0.70 (0.13)
F27	0.90 (0.07)	<b>0.94 (0.02)</b>	0.90 (0.06)	<b>0.99 (0.01)</b>	0.37 (0.05)	0.62 (0.06)
F28	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.02)</b>	0.24 (0.06)	0.47 (0.08)
F29	0.96 (0.01)	0.96 (0.01)	<b>0.98 (0.00)</b>	<b>0.99 (0.02)</b>	0.24 (0.05)	0.44 (0.05)
F30	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.02)</b>	0.27 (0.05)	0.45 (0.06)
F31	0.98 (0.01)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.96 (0.07)</b>	0.16 (0.03)	0.42 (0.12)
F32	0.99 (0.00)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.95 (0.06)</b>	0.14 (0.02)	0.34 (0.06)
F33	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.02)</b>	0.25 (0.03)	0.56 (0.12)
F34	0.94 (0.02)	0.94 (0.01)	<b>0.95 (0.02)</b>	<b>0.99 (0.02)</b>	0.35 (0.11)	0.50 (0.05)
F35	0.78 (0.02)	0.81 (0.01)	<b>0.82 (0.01)</b>	<b>0.99 (0.01)</b>	0.40 (0.06)	0.55 (0.04)
F36	0.91 (0.03)	<b>0.97 (0.01)</b>	0.98 (0.02)	<b>0.99 (0.01)</b>	0.30 (0.04)	0.52 (0.05)
F37	0.89 (0.02)	0.96 (0.01)	<b>0.97 (0.01)</b>	<b>0.98 (0.02)</b>	0.21 (0.03)	0.42 (0.03)
F38	0.91 (0.03)	0.96 (0.02)	<b>0.97 (0.02)</b>	<b>0.91 (0.12)</b>	0.18 (0.05)	0.39 (0.06)
F39	0.94 (0.01)	0.95 (0.01)	<b>0.97 (0.01)</b>	<b>0.99 (0.01)</b>	0.33 (0.05)	0.53 (0.07)
F40	0.61 (0.05)	<b>0.67 (0.01)</b>	0.64 (0.04)	<b>0.99 (0.01)</b>	0.52 (0.09)	0.67 (0.05)
F41	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.26 (0.05)	0.50 (0.07)
F42	0.98 (0.01)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.98 (0.02)</b>	0.16 (0.03)	0.37 (0.05)
F43	0.99 (0.00)	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>	<b>0.95 (0.06)</b>	0.15 (0.02)	0.35 (0.03)
F44	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>0.99 (0.01)</b>	0.28 (0.05)	0.51 (0.10)
F45	0.93 (0.06)	<b>0.98 (0.02)</b>	0.95 (0.05)	<b>0.99 (0.01)</b>	0.36 (0.07)	0.58 (0.08)
F46	0.97 (0.01)	<b>1.00 (0.00)</b>	0.98 (0.01)	<b>0.99 (0.01)</b>	0.16 (0.03)	0.58 (0.15)
F47	0.98 (0.00)	<b>1.00 (0.00)</b>	0.98 (0.00)	<b>0.99 (0.01)</b>	0.15 (0.03)	0.59 (0.15)
F48	0.98 (0.00)	<b>1.00 (0.00)</b>	0.98 (0.01)	<b>0.93 (0.13)</b>	0.15 (0.02)	0.39 (0.12)
F49	0.91 (0.05)	<b>0.97 (0.00)</b>	0.95 (0.03)	<b>0.98 (0.02)</b>	0.21 (0.03)	0.43 (0.10)
F50	0.99 (0.00)	<b>1.00 (0.00)</b>	0.99 (0.00)	<b>0.99 (0.01)</b>	0.13 (0.02)	0.62 (0.10)
F51	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>	<b>0.97 (0.03)</b>	0.14 (0.02)	0.36 (0.08)
F52	0.90 (0.05)	<b>0.98 (0.01)</b>	0.97 (0.03)	<b>0.98 (0.02)</b>	0.16 (0.03)	0.44 (0.09)
F53	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>	0.37 (0.08)	0.61 (0.07)
F54	0.94 (0.06)	<b>0.97 (0.02)</b>	0.95 (0.03)	<b>1.00 (0.01)</b>	0.35 (0.05)	0.60 (0.06)
F55	0.22 (0.03)	<b>0.32 (0.04)</b>	0.22 (0.07)	<b>0.99 (0.01)</b>	0.87 (0.05)	0.95 (0.03)
Best/total	31/73	57/73	43/73	73/73	2/73	1/73

## 6.2 Anytime Performance

Figures 1, 2, 3, 4 illustrate the anytime performance of the three different MOEA/D variants hypervolume (higher is better) on the UF10, DTLZ3<sup>-1</sup>, DTLZ2 and F9, respectively. We selected those Figures because they represent well the trend behaviour of the algorithms studied in this work. These results indicate that MOEA/D-PS shows an improvement over MOEA/D with a big population, achieving higher HV values in earlier stages of the search. Surprisingly, MOEA/D-PS converges as fast as MOEA/D with a small population. This similar convergence behaviour is consistent when considering only the working population since the working population dynamics in MOEA/D-PS has the size to the working population dynamics of MOEA/D with small population.

The anytime performance Figures 1, 2, 3, 4 together with Table 2 reveal the most striking observation to emerge from our conceptual comparison. What stands out is that MOEA/D-PS improves the convergence speed substantially when compared to MOEA/D using the same population size (MOEA/D-PS vs MOEA/D with big population). At the same time, it achieves an impressive higher number of non-dominated solutions (Table 2). Moreover, this observation is not limited to a specific group of MOPs but present in most MOPs studied here, suggesting the stability of MOEA/D with the Partial Update Strategy. This combination of findings provides an explanation of the relationship between MOEA/D-PS with the consistent improvements in performance over the MOEA/D framework.

## 6.3 Empirical Attainment Performance

Figures 5, 6, 7, 8, 9 and 10 depict the differences between the EAFs of MOEA/D-PS and the EAF of the MOEA/D with different population sizes, on F22 and DTLZ1<sup>-1</sup>, respectively, in shades of red. *The shades of the red show the amount of the differences of the probabilistic distribution of the outcomes obtained by the algorithms: shades closer to red indicate higher differences between the probability distributions, shades closer to orange indicate little difference, and shades closer to white indicate no difference.* We chose these problems because they represent well the behaviour of the algorithms we study here. The lower line shows the global best set of solutions attained overall runs of all algorithms (grand best attainment surface) in all Figures. In contrast, the upper line shows solutions that are always dominated (grand worst attainment surface). For all Figures, the left panel show regions where the EAF of MOEA/D-PS performs better or worse when compared to the other MOEA/D variants. For Figures 5, 7 and 9, the right panels show differences in support of MOEA/D with small population size and the right panels on Figures 6, 8 and 10 show the differences in support of MOEA/D with big population size..

Figures 5 and 6 show the EAF for the F22 MOP. We can see that MOEA/D-PS performs better than the other two variations on the centre side of the Figure, attaining this region at least 60% of the runs, indicated by the shades of red. This result is a rather surprising one. That is because a closer inspection of the Figures shows that MOEA/D-PS has a strong tendency to focus on well-balanced solutions, i.e., solutions that have a good trade-off among the objectives. Figures 7 and 8 show the same behaviour for the UF6 MOP. However, this focus on central regions only is not as strong for the MOP with a linear-shaped Pareto Front (DTLZ1<sup>-1</sup>, discussed next).

As above, Figure 9 and 10 show, in shades of red, the attainment regions but now for the DTLZ1<sup>-1</sup>. Looking at both Figures, we can see that MOEA/D-PS performs much better than MOEA/D with both small and big population. These big differences suggest that MOEA/D-PS can perform much better in MOPs with a linear-shaped Pareto Front.

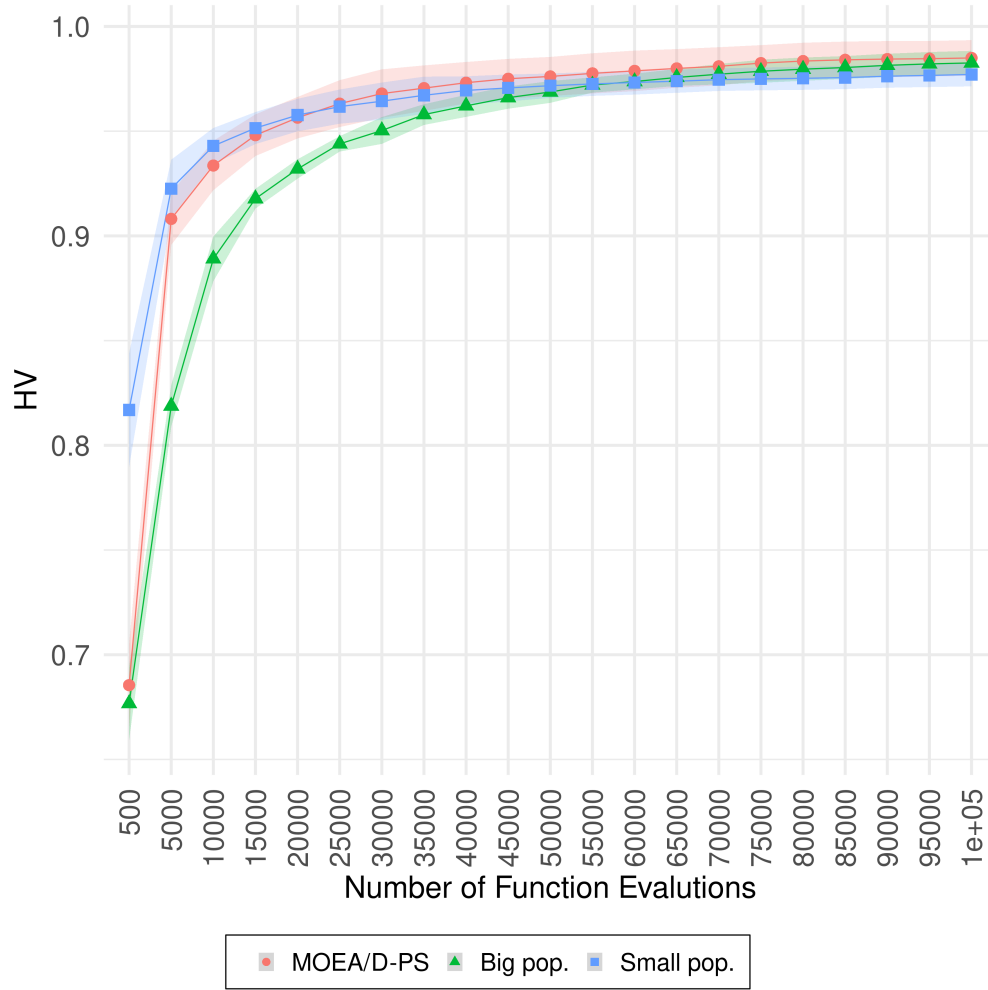


Figure 1: Anytime HV (higher is better, shaded areas indicate the standard deviation) on UF10. The performance of MOEA/D-PS is shown as the red circles, MOEA/D with population size 500 is shown as the green triangles, and MOEA/D with population size equals to 50 is shown as the blue squares, on UF10. The anytime performance of MOEA/D-PS is similar to the anytime performance of MOEA/D with a small population. We can see that the three variants have almost the same performance at the end of the search.

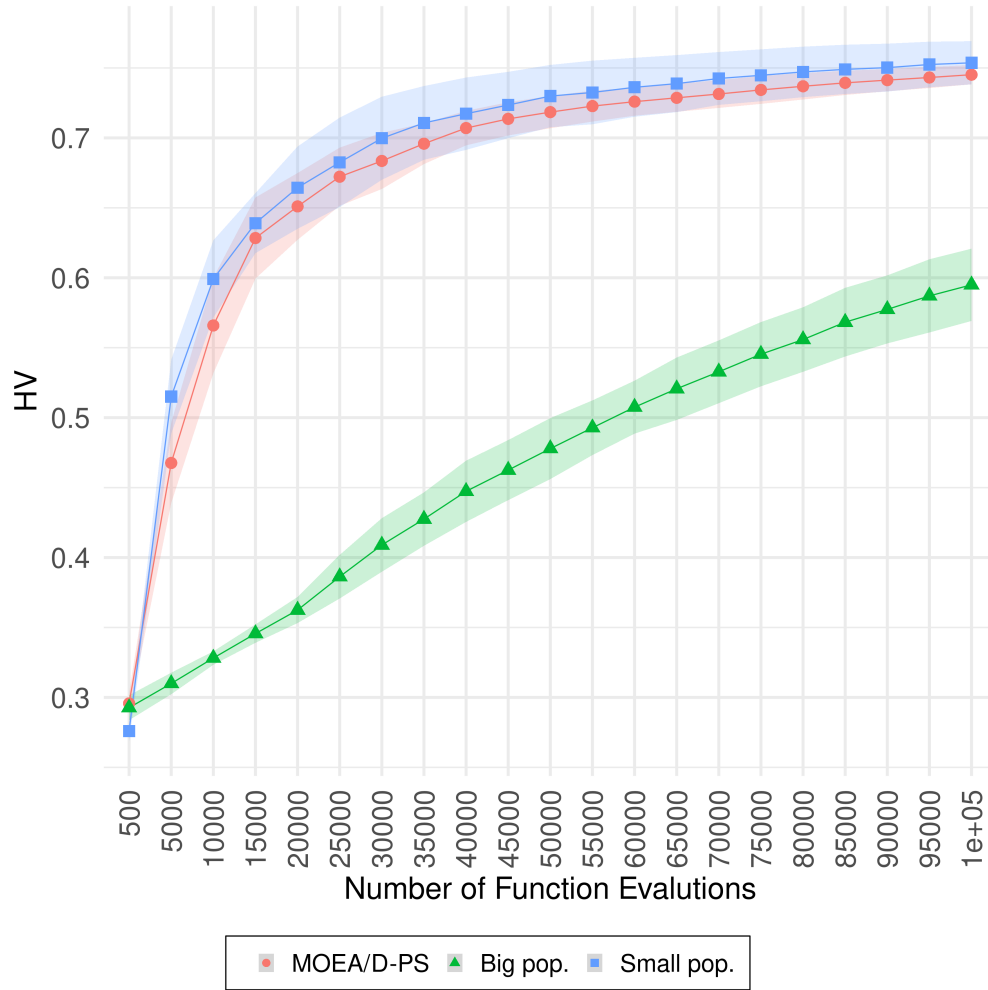


Figure 2: Anytime HV (higher is better, shaded areas indicate the standard deviation) on  $DTLZ3^{-1}$ . The performance of MOEA/D-PS is shown as the red circles, MOEA/D with population size 500 is shown as the green triangles, and MOEA/D with population size equals to 50 is shown as the blue squares, on  $DTLZ3^{-1}$ . The anytime performance of MOEA/D-PS is similar to the anytime performance of MOEA/D with a small population and much faster than MOEA/D with big population.



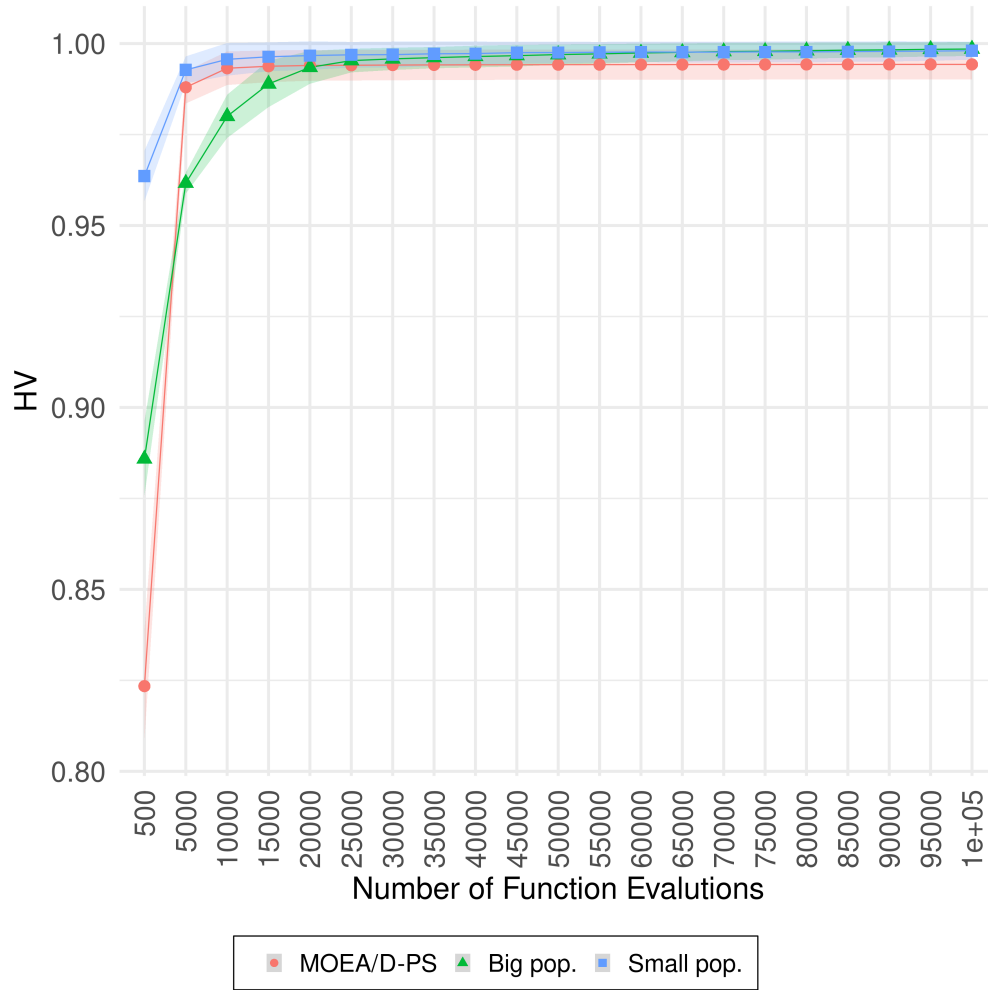


Figure 3: Anytime HV (higher is better, shaded areas indicate the standard deviation) on DTLZ1. The performance of MOEA/D-PS is shown as the red circles, MOEA/D with population size 500 is shown as the green triangles, and MOEA/D with population size equals to 50 is shown as the blue squares, on DTLZ1. The anytime performance of MOEA/D-PS is similar to the anytime performance of MOEA/D with a small population.

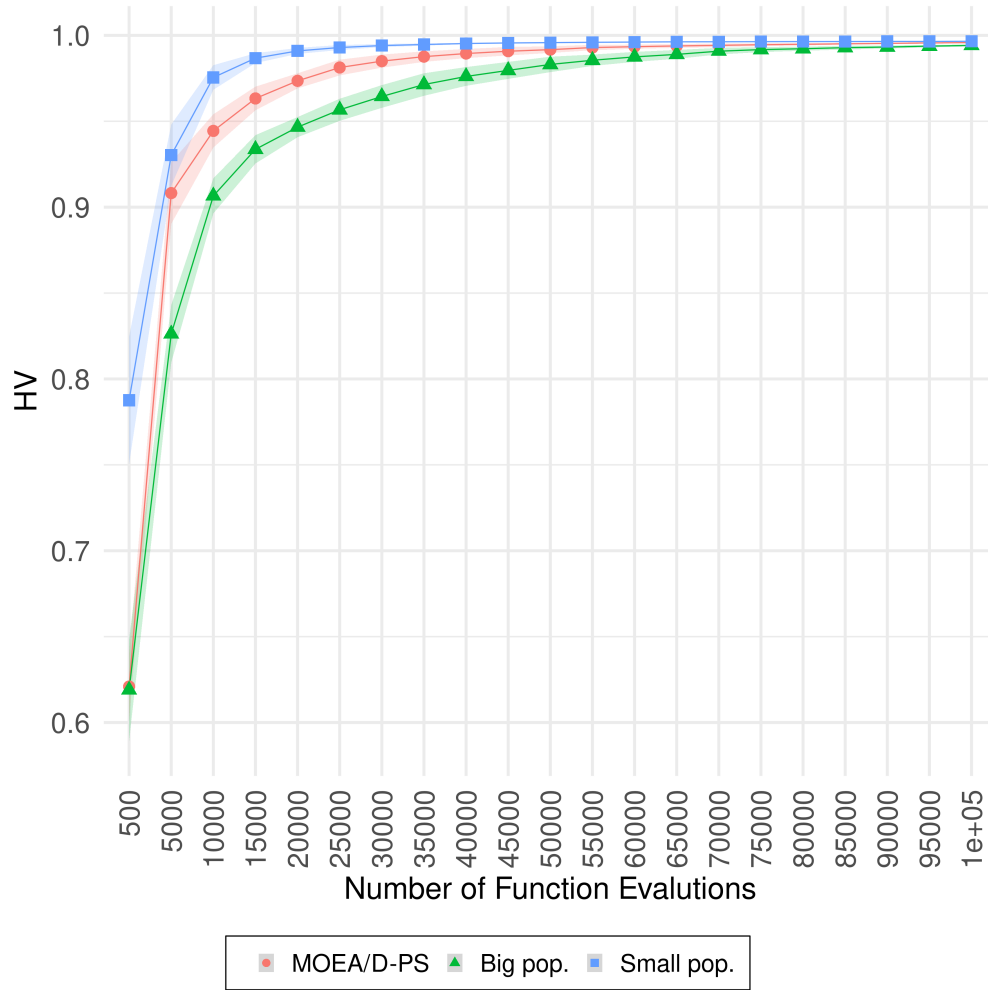


Figure 4: Anytime HV (higher is better, shaded areas indicate the standard deviation) on F9. The performance of MOEA/D-PS is shown as the red circles, MOEA/D with population size 500 is shown as the green triangles, and MOEA/D with population size equals to 50 is shown as the blue squares, on F9. The anytime performance of MOEA/D-PS shows that this algorithm converges as fast as MOEA/D with a small population and that MOEA/D is faster than MOEA/D with big population.

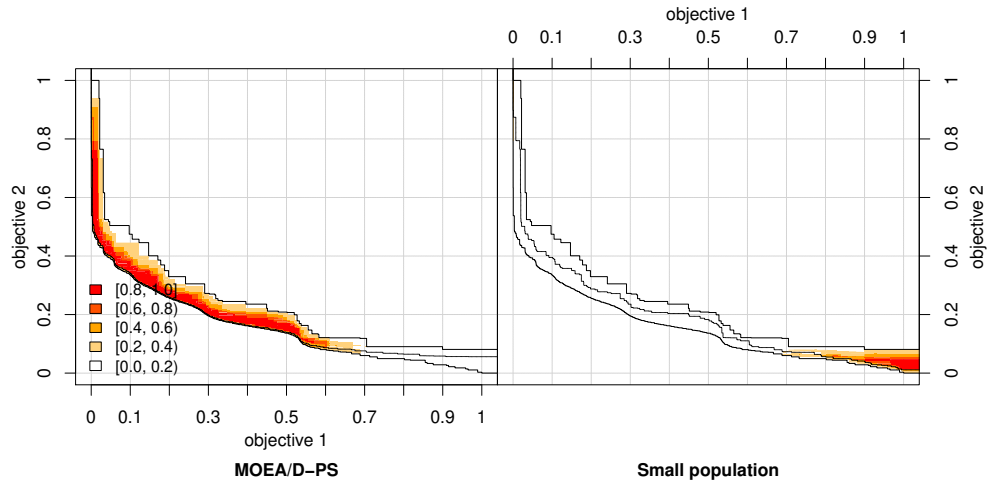


Figure 5: Differences between the EAFs of MOEA/D-PS with MOEA/D with small population. The red shade encodes the magnitude of the observed difference on F22. The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs better in almost all regions of the objective space, except for the region closer to objective 1 (low values on the x-axis). This region is where MOEA/D with small population (right) performs better.

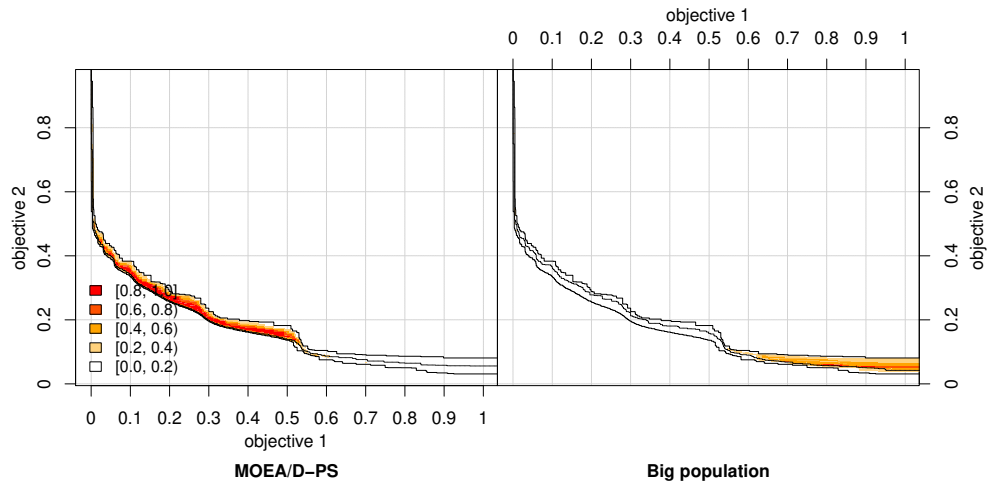


Figure 6: Differences between the EAFs of MOEA/D-PS with MOEA/D with big population. The red shade encodes the magnitude of the observed difference on F22. The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs better at the central regions of the objective space, with minimal advantage on the region closer to objective 2 (low values on the y-axis). MOEA/D with big population (right) performs better in the region closer to objective 1 (low values on the x-axis).

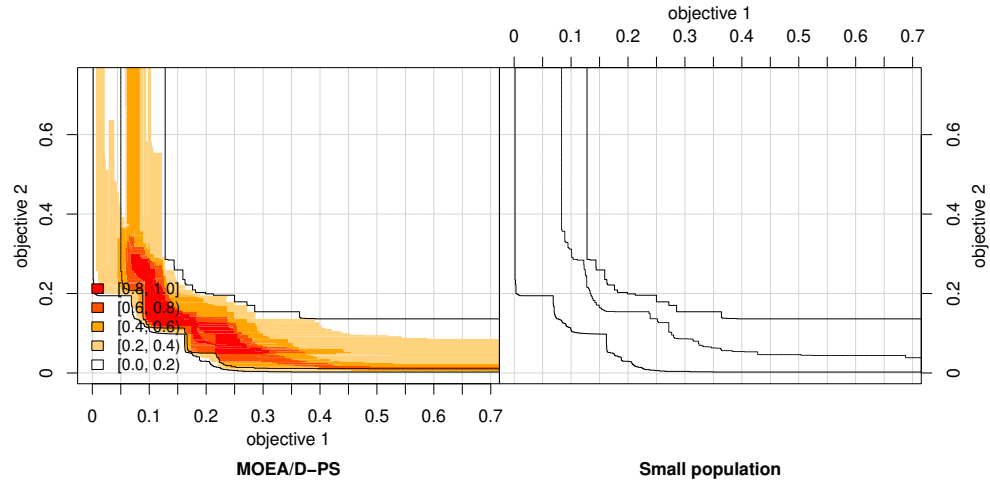


Figure 7: Differences between the EAFs of MOEA/D-PS with MOEA/D with small population. The red shade encodes the magnitude of the observed difference on UF6. The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs well towards all regions of the objective space.

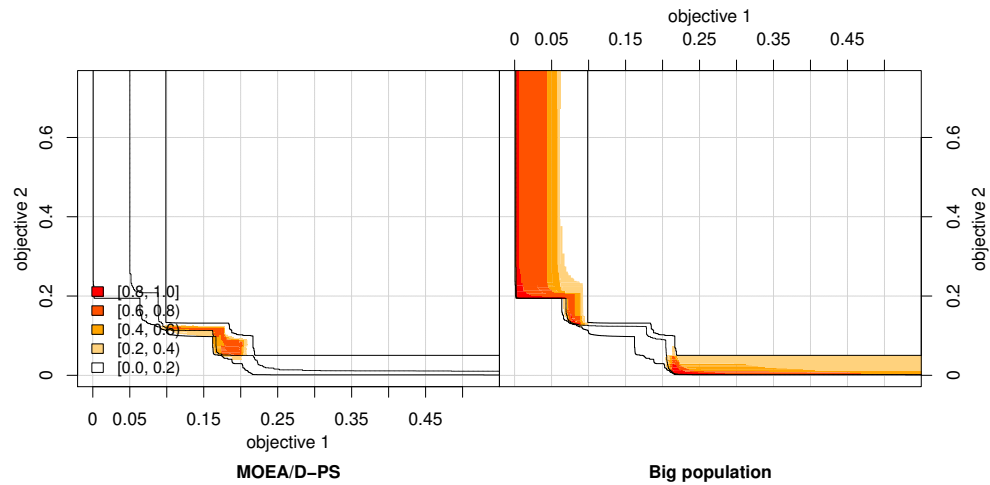


Figure 8: Differences between the EAFs of MOEA/D-PS with MOEA/D with big population. The red shade encodes the magnitude of the observed difference on UF6. The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs better at central regions of the objective space. In contrast, MOEA/D with big population (right) performs better in the regions closer to objectives 1 and 2 (low values on the x-axis and y-axis).

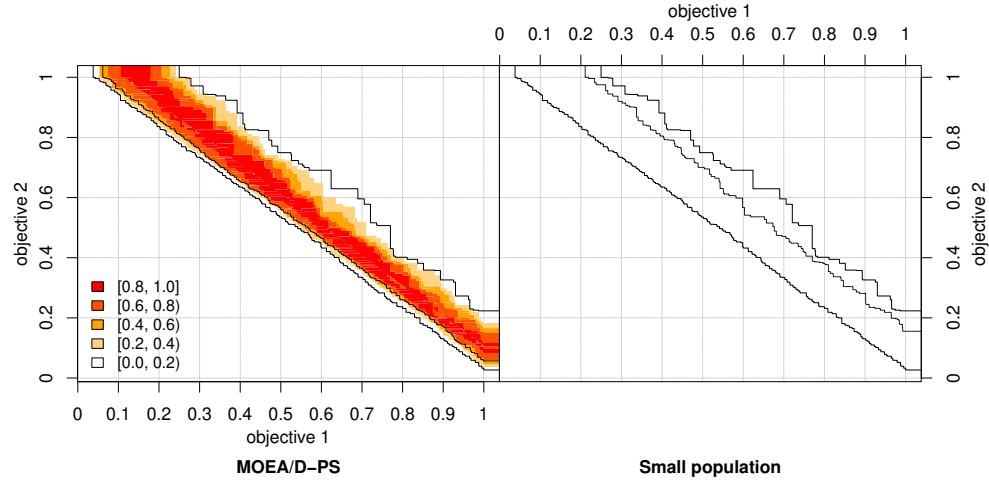


Figure 9: Differences between the EAFs of MOEA/D-PS with MOEA/D with small population. The red shade encodes the magnitude of the observed difference on  $DTLZ1^{-1}$ . The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs well towards all regions of the objective space.

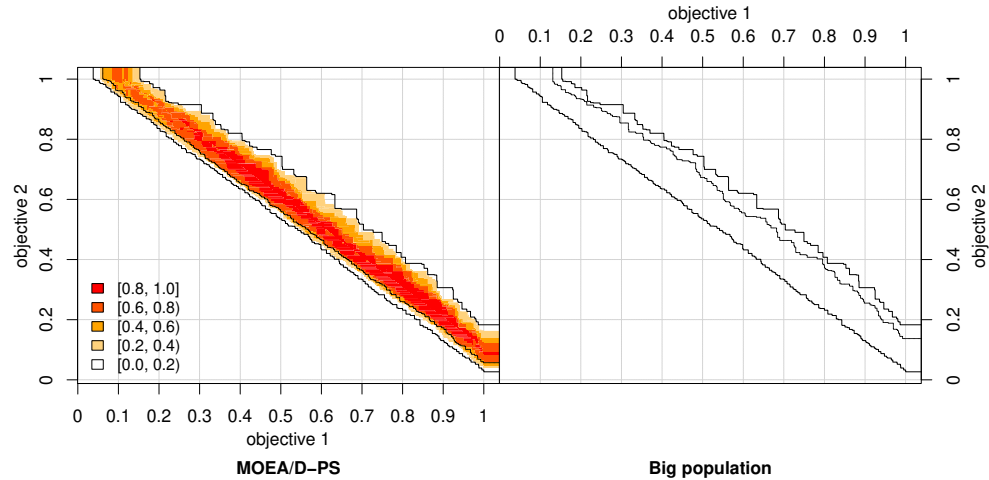


Figure 10: Differences between the EAFs of MOEA/D-PS with MOEA/D with big population. The red shade encodes the magnitude of the observed difference on  $DTLZ1^{-1}$ . The darker the red shade, the better performance of the algorithm. MOEA/D-PS (left) performs better at all regions of the objective space.

#### 6.4 Statistical Analysis

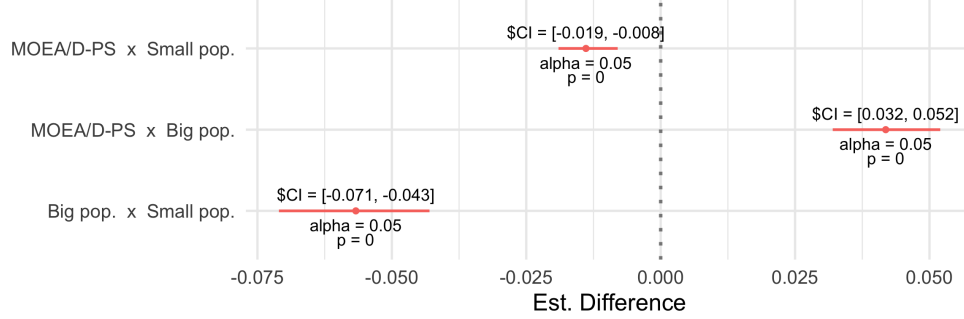


Figure 11: Statistical comparisons between the methods tested using 95% confidence intervals for the median of paired differences of hypervolume values at 5000 function evaluations between MOEA/D-PS and MOEA/D with big population and MOEA/D with small population. Red intervals with a diamond indicate statistically significant results. MOEA/D-PS has a statistical advantage over MOEA/D with big population and a disadvantage over MOEA/D with small population. This result confirms that MOEA/D-PS increases the convergence speed over MOEA/D with big population, when having a small working population may help MOEA/D-PS find increments in hypervolume convergence speed.

Table 3: Anytime Statistical significance of differences in median HV, for the three algorithms tested in this section, at 5000 function evaluations. Values are Hommel-adjusted p-values of Wilcoxon Rank-sum tests. “↑” indicates superiority of the column method and “←” indicates superiority of the row method (95% confidence level).

5000 function evaluations		
	MOEA/D-PS	Big pop.
Big pop.	2.890526e-11 ↑	
Small pop.	1.088418e-10 ←	9.243532e-13 ←

The current study found that MOEA/D-PS improves the convergence speed substantially when compared to MOEA/D using the same population size (MOEA/D-PS vs MOEA/D with big population, Subsection 6.2). Here we want to verify the convergence improvements of MOEA/D-PS over MOEA/D in terms of higher HV values in earlier stages of the search since all methods achieve approximately the same hypervolume result. For that, we perform the Wilcoxon Rank Sum Tests over 5000 and 15000 evaluations. We also consider the final result for completeness, when the computational budget is met, at 100000 evaluations.

Figure 11 and Table 3 present the statistical comparison between MOEA/D-PS, MOEA/D with small population and MOEA/D with big population at 5000 function evaluations. As we can see, MOEA/D-PS has a statistical advantage over MOEA/D with big population and a disadvantage over MOEA/D with small population, confirming our previous analysis that MOEA/D-PS increases the convergence speed over MOEA/D with big population, lower bounded by MOEA/D with small population. The same trend continues until around 15000 function evaluations are met, as shown



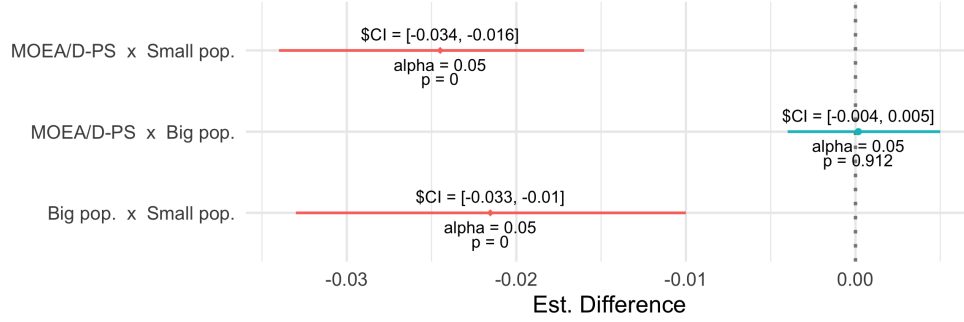


Figure 12: Statistical comparisons between the methods tested using 95% confidence intervals for the median of paired differences of hypervolume values at 15000 function evaluations between MOEA/D-PS and MOEA/D with big population and MOEA/D with small population. Red intervals with a diamond indicate statistically significant results, and blue intervals with a circle indicate no statistically significant results. MOEA/D-PS has no statistically significant differences to MOEA/D with big population and a disadvantage over MOEA/D with small population. We consider this to be the point in search where MOEA/D with big population start to recover from its slower convergence speed.

Table 4: Anytime Statistical significance of differences in median HV, for the three algorithms tested in this section, at 15000 function evaluations. Values are Hommel-adjusted p-values of Wilcoxon Rank-sum tests. “↑” indicates superiority of the column method, “←” indicates superiority of the row method and “≈” indicates differences not statistically significant (95% confidence level).

15000 function evaluations		
	MOEA/D-PS	Big pop.
Big pop.	9.124481e-01 ≈	
Small pop.	1.169062e-11 ←	4.004438e-07 ←

Table 5: Anytime Statistical significance of differences in median HV, for the three algorithms tested in this section, at 100000 function evaluations. Values are Hommel-adjusted p-values of Wilcoxon Rank-sum tests. “↑” indicates superiority of the column method, “←” indicates superiority of the row method and “≈” indicates differences not statistically significant (95% confidence level).

100000 function evaluations		
	MOEA/D-PS	Big pop.
Big pop.	1.99 e-04 ↑	
Small pop.	7.26 e-05 ↑	0.14 ≈

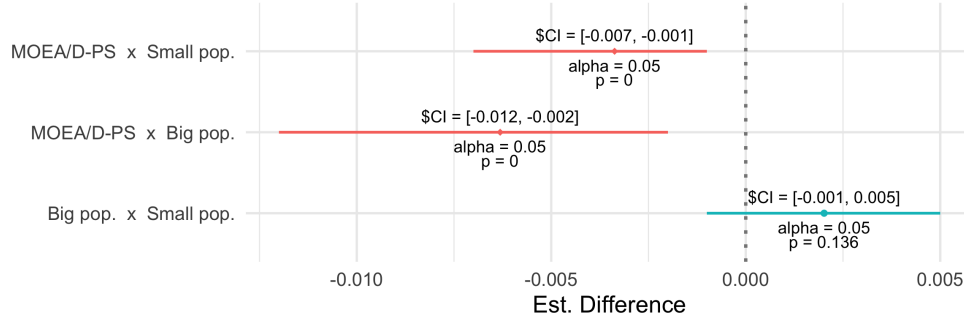


Figure 13: Statistical comparisons between the methods tested using 95% confidence intervals on the paired differences between the algorithms, given hypervolume values at 100000 function evaluations. Red intervals with a diamond indicate statistically significant results, and blue intervals with a circle indicate no statistically significant results. MOEA/D-PS has no statistically significant differences to MOEA/D with big population and a disadvantage over MOEA/D with small population. Although all algorithms achieve high hypervolume results in most MOP, we found statistical differences among them at the end of the search. A big population may help MOEA/D find increments in hypervolume.

in Figure 12 and Table 4) when the statistical results show no difference between MOEA/D-PS and MOEA/D with big population. Then, at 100000 (the maximum number of evaluations), we can see at Figure 13 and Table 5 that the statistical tests indicate some disadvantage of MOEA/D-PS against the other methods. In summary, we found a significant difference between all methods over the first initial stages of the search, corroborating the results observed in the anytime performance, Subsection 6.2). There are statistical differences in the hypervolume values of the final approximation sets, although all algorithms achieve good results in most of the functions.

### 6.5 Influence of the Population Size

Another reason for the lower performance of MOEA/D with a small population size might be that the limits imposed by the size of the working population and consequently on the quality of the external archive. We reason that such a small population size negatively influences the external archive because good solutions found by the algorithm during the run might be excluded from the final population.

On the other hand, MOEA/D-PS might be able to use the larger population as an extra archive population for a small working population. It might be that, for problems where a large population size is needed to search for better-performing solutions, using the PS strategy is useful because the large population size of MOEA/D-PS is playing the role of an archive population for a small working population size. However, more work is needed to clarify the relationship between the archive population and the large population in MOEA/D-PS and in more general, the large population of MOEA/D with RA.

## 7 Conclusion

In this work, we investigate the relationship of partially updating the population in MOEA/D using Resource Allocation, represented by MOEA/D-PS, and population

size in practice. This study is motivated by the realisation that the Partial Update Strategy uses its control parameter,  $n$ , to regulate the proportion of the population selected for variation at any iteration, the working population. This working population usually has a small size, resembling MOEA/D variants with small population size. On the other hand, the whole population, which combines both the working population and the population not selected by the Partial Update Strategy, has a considerable size. This size resembles MOEA/D variants with big population size. We extensively study the conceptual correspondences of MOEA/D with the Partial Update and MOEA/D with small population size and big population size on more than 70 MOPs.

We found strong evidence that MOEA/D-PS performance is good independently of the MOP in question since MOEA/D with the Partial Update of the population (MOEA/D-PS) always performs as one of the best algorithms in the MOPs studied here. The anytime analysis of the algorithms shows that MOEA/D-PS has a similar convergence behavior as MOEA/D with a small population. That is, MOEA/D-PS quickly finds its best results while maintaining the benefit of a broader exploration of the search space, a common characteristic of evolutionary algorithms with a big population. The reason relies on the relationship between different populations and their advantages in solving MOPs. That is, a small population can approach the Pareto Front quickly but might not be able to explore and cover this Pareto Front given its limited size; however, a larger population will likely be better at this task, at a higher cost (Glasmachers et al. (2014)).

The anytime results found in Section 6 show that MOEA/D-PS provides a simple yet efficient approach to mitigate common problems related to population size choice. Examples of such problems are the likely waste of computation resources induced by the large size of a population or the premature stagnation caused by the small size of the working population (Lin et al. (2019)). Consequently, the results of this work confirm that MOEA/D-PS is an effective and stable improvement over the MOEA/D framework. Moreover, the anytime analysis complements the results of those earlier studies, such as maintaining a big population size while updating a small subset of solutions as in Pruvost et al. (2020); Lavinias et al. (2020). Besides, this study demonstrates the partial update strategy ability to increase the convergence speed in most of the more than 70 MOPs explored in this work, as confirmed by the overall good anytime performance. More work is needed to determine the best strategy for deciding which solutions are selected to be updated.

In addition to the anytime performance, we further analysed the performance of MOEA/D with the Partial Update Strategy using the Empirical Attainment Function (EAF). The EAF results illustrate the hybrid behaviour of MOEA/D-PS, as we can observe that it can find better EAF regions in all problems tested compared to both MOEA/D with a small or a big population. This analysis also suggests that MOEA/D-PS tends to focus on well-balanced solutions, i.e., solutions with a well-balanced trade-off among objectives.

Moreover, our findings have significant implications for the understanding of how MOEA/D with Resource Allocation works. Together, these findings support that using the partial update of the population as one improvement to the MOEA/D framework shall be preferred. As future works, we want to verify whether MOEA/D-PS would benefit from adapting the  $n$  value throughout the search and from studying the algorithm behaviour on constrained MOPs such as in the one proposed by Andrew Ning et al. (2014); Nishiyama et al. (2015); Kohira et al. (2018).

## References

- Andrew Ning, S., Damiani, R., and Moriarty, P. J. (2014). Objectives and constraints for wind turbine optimization. *Journal of Solar Energy Engineering*, 136(4).
- Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., and Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082.
- Bezerra, L. C., López-Ibáñez, M., and Stützle, T. (2015). Comparing decomposition-based and automatically component-wise designed multi-objective evolutionary algorithms. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 396–410. Springer.
- Bossek, J. (2017). smooft: Single- and multi-objective optimization test functions. *The R Journal*.
- Cai, X., Li, Y., Fan, Z., and Zhang, Q. (2015). An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):508–523.
- Campelo, F. and Aranha, C. (2018). MOEADr: Component-wise MOEA/D implementation. URL <https://cran.R-project.org/package=MOEADr>. R package version 1.2.0.
- Campelo, F., Batista, L., and Aranha, C. (2020). The MOEADr package: A component-based framework for multiobjective evolutionary algorithms based on decomposition. *Journal of Statistical Software*. In press. Available from: <https://arxiv.org/abs/1807.06731>.
- Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 45(3):1–33.
- Da Fonseca, V. G., Fonseca, C. M., and Hall, A. O. (2001). Inferential performance assessment of stochastic optimisers and the attainment function. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 213–225. Springer.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer.
- Dubois-Lacoste, J., López-Ibáñez, M., and Stützle, T. (2015). Anytime pareto local search. *European Journal of Operational Research*, 243(2):369 – 385.
- Finck, S., Hansen, N., Ros, R., and Auger, A. (2010). Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical report, Citeseer.
- Fonseca, C. M. and Fleming, P. J. (1996). On the performance assessment and comparison of stochastic multiobjective optimizers. In *International Conference on Parallel Problem Solving from Nature*, pages 584–593. Springer.
- Glasmachers, T., Naujoks, B., and Rudolph, G. (2014). Start small, grow big? Saving multi-objective function evaluations. In *International Conference on Parallel Problem Solving from Nature*, pages 579–588. Springer.
- Goulart, F., Borges, S. T., Takahashi, F. C., and Campelo, F. (2017). Robust multiobjective optimization using regression models and linear subproblems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 569–576. ACM.
- Hansen, N., Ros, R., Mauny, N., Schoenauer, M., and Auger, A. (2011). Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769.
- Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.

- Ishibuchi, H., Pang, L. M., and Shang, K. (2020). A new framework of evolutionary multi-objective algorithms with an unbounded external archive.
- Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. (2016). Performance of decomposition-based many-objective algorithms strongly depends on Pareto Front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190.
- Jain, H. and Deb, K. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622.
- Kohira, T., Kemmotsu, H., Akira, O., and Tatsukawa, T. (2018). Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 183–184. ACM.
- Lavinas, Y., Aranha, C., and Ladeira, M. (2019a). Improving resource allocation in MOEA/D with decision-space diversity metrics. In *Theory and Practice of Natural Computing*, pages 134–146, Cham. Springer International Publishing.
- Lavinas, Y., Aranha, C., Ladeira, M., and Campelo, F. (2020). MOEA/D with random partial update strategy. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- Lavinas, Y., Aranha, C., and Sakurai, T. (2019b). Using diversity as a priority function for resource allocation on MOEA/D. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 215–216. ACM.
- Li, H., Deb, K., Zhang, Q., Suganthan, P., and Chen, L. (2019). Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties. *Swarm and Evolutionary Computation*, 46:104–117.
- Li, H. and Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.
- Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.
- Li, M. and Yao, X. (2019). An empirical investigation of the optimality and monotonicity properties of multiobjective archiving methods. In Deb, K., Goodman, E., Coello Coello, C. A., Klamroth, K., Miettinen, K., Mostaghim, S., and Reed, P., editors, *Evolutionary Multi-Criterion Optimization*, pages 15–26, Cham. Springer International Publishing.
- Lin, Q., Zhu, Q., Wang, N., Huang, P., Wang, W., Chen, J., and Ming, Z. (2019). A multi-objective immune algorithm with dynamic population strategy. *Swarm and Evolutionary Computation*, 50:100477.
- López-Ibáñez, M., Paquete, L., and Stützle, T. (2010). Exploratory analysis of stochastic local search algorithms in biobjective optimization. In *Experimental Methods for the Analysis of Optimization Algorithms*, pages 209–222. Springer.
- Nasir, M., Mondal, A. K., Sengupta, S., Das, S., and Abraham, A. (2011). An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 765–772. IEEE.
- Nishiyama, M., Otake, H., Hoshino, T., Hashimoto, T., Watanabe, T. W., Tatsuaki, T., and Oyama, A. (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*.
- Pruvost, G., Derbel, B., Liefooghe, A., Li, K., and Zhang, Q. (2020). On the combined impact of population size and sub-problem selection in MOEA/D. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, pages 131–147. Springer.

- Radulescu, A., López-Ibáñez, M., and Stützle, T. (2013). Automatically improving the anytime behaviour of multiobjective evolutionary algorithms. In Purshouse, R. C., Fleming, P. J., Fonseca, C. M., Greco, S., and Shaw, J., editors, *Evolutionary Multi-Criterion Optimization*, pages 825–840, Berlin, Heidelberg, Springer Berlin Heidelberg.
- Tanabe, R., Ishibuchi, H., and Oyama, A. (2017). Benchmarking multi-and many-objective evolutionary algorithms under two optimization scenarios. *IEEE Access*, 5:19597–19619.
- Tusar, T., Brockhoff, D., Hansen, N., and Auger, A. (2016). COCO: the bi-objective black box optimization benchmarking (bbob-biobj) test suite. *arXiv preprint arXiv:1604.00359*.
- Wang, P., Zhu, W., Liu, H., Liao, B., Cai, L., Wei, X., Ren, S., and Yang, J. (2019). A new resource allocation strategy based on the relationship between subproblems for MOEA/D. *Information Sciences*, 501:337–362.
- Zapotecas-Martínez, S., Aguirre, H., Tanaka, K., and Coello, C. (2015). On the low-discrepancy sequences and their use in MOEA/D for high-dimensional objective spaces. In *Congress on Evol. Computation (CEC 2015)*, pages 2835–2842.
- Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.
- Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of MOEA/D on CEC’09 unconstrained MOP test instances. In *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pages 203–208. IEEE.
- Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (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.
- Zhou, A. and Zhang, Q. (2016). Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20(1):52–64.
- Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73–73.