Studying the Effect of Robustness Measures in Offline Parameter Tuning for Estimating the Performance of MOEA/D

Miriam Pescador-Rojas
ESCOM, Instituto Politécnico Nacional
Computer Science Department, CINVESTAV-IPN
pescador@computacion.cs.cinvestav.mx

Denis Pallez
Université Côte d'Azur
CNRS, UMI 3175, UMR 7271
denis.pallez@unice.fr

Carlos Ignacio Hernández Castellanos
Technology Department
Mutuo Financiera
carlos@mutuofinanciera.com

Carlos A. Coello Coello
Computer Science Department,
CINVESTAV-IPN
ccoello@cs.cinvestav.mx

***Abstract*—Offline parameter tuning (OPT) of multi-objective evolutionary algorithms (MOEA) has the goal to find an appropriate set of parameters for solving a large number of problems. According to the no free lunch theorem (NFL), no algorithm can have the best performance in all classes of optimization problems. However, it is possible to find an appropriate set of parameters of an algorithm for solving a particular class of problems. For that sake, we need to study how to estimate the aggregate quality function for an algorithmic configuration assessed on a set of optimization problems.**

In this paper, we study robustness measures for dealing with the parameter settings of stochastic algorithms. We focus on decomposition-based MOEAs and propose to tune scalarizing functions for solving some classes of problems based on the Pareto front shapes using up to 7 objective functions. Based on our experimental results, we were able to derive interesting guidelines to evaluate the quality of algorithmic configurations using a combination of descriptive statistics.

***Index Terms*—Offline parameter tuning, robustness measures, multiobjective evolutionary algorithms, scalarizing functions.**

I. Introduction

The challenge for offline parameter tuning (OPT) techniques is to find *the best* parameter configuration of an evolutionary algorithm that successfully solves a large number of multi-objective optimization problems (MOPs). In multiobjective evolutionary algorithms (MOEAs), each parameter has a significant influence on their performance such as its rate of convergence and on the quality of the solutions obtained. OPT is subject to uncertainty according to two aspects: 1) the stochastic procedures involved in MOEAs, 2) the optimal configuration for an MOEA optimizing a set of problem instances at the same time.

The goal of this paper is to study robustness measures used in the area of uncertainty [1], [2] to compute the performance of an algorithmic configuration used in the offline tuning methods. Our main contribution is to provide a set of guidelines on the use of robustness measures for solving multi-objective optimization problems (MOPs). We adopt a particular case study to improve the performance of MOEA based on decomposition (MOEA/D) and we employ an evolutionary OPT tool which has been used in other studies [3]–[5]. We focus on an important component of MOEA/D, the scalarizing function (SF) which is a strategy to transform a MOP into several single objective problems. Some SFs involve additional parameters in $\mathcal{R}$ that require to be tuned according to particular characteristics on the MOP such as the Pareto front shape or the number of objective functions.

The remainder of this paper is organized as follows. Previous works on parameter settings are briefly discussed in section II with a particular focus on MOPs detailed in section IV. Then, uncertainty and robustness measures are introduced in section III. Decomposition-based MOEAs based on scalarizing functions are described in section V. Section VI presents our methodology for studying robustness. Section VII presents our results and their discussion. We finally conclude the paper and present some possible paths for future work in section VIII.

II. Parameter settings

Performance of EAs is strongly related to the definition of appropriate parameter values. Typically, before running an EA, the user empirically determines the proper encoding, selection mechanism, and evolutionary operators, as well as their numerical parameters such as the population size, mutation and crossover rates. There exist two main possibilities for dealing with the parameter setting problem: offline parameter tuning and the online parameter control strategies [6]. The first one refers to selecting a set of parameter values which can be established by hand according to the user's prior experience or applying experimental design methods. In these cases, the same set of parameters is used in all the iterations of an EA. The second strategy includes adaptive mechanisms where the parameter values are modified using information gathered during the evolutionary search process. Adaptive and self-adaptive strategies are feedback-based approaches. The former uses indicators to monitor the performance of the evolutionary process. The second strategy encodes the parameter values in

each individual of the population and solves at the same time the optimization and the configuration problems.

In this work, we focus on the OPT problem which can be formalized as a 6-tuple $T = (A, P, C, O, \psi, b_{max})$ as follows:

- A is a target algorithm to be tuned. In this work, we concentrate on MOEA/D, which is the most popular decomposition-based MOEA;
- P is the set of parameters of A to be tuned. For instance, if A is an EA, then population size, maximum number of generations, crossover rate, etc. belong to P.
- C is the set of values for each parameter of P. An instance $c \in C$ is called a *configuration* of A;
- O is called a *scenario* and is a set of several problem instances (for example, {DTLZ1, DTLZ3, DTLZ5, DTLZ7} from the *Deb-Thiele-Laumanns-Zitzler (DTLZ)* test suite [7]);
- ψ is a scalar fitness function used by T to asses the quality of a configuration e.g., hypervolume (HV) [8], inverted generational distance (IGD) [9] or the $R2$ [10] indicator.
- b_{max} is the maximum budget of function evaluations available to tune A on O;

Here, the goal is to find the 'best' configuration $c \in C$, based on ψ, for tuning A on a scenario O with b_{max} function evaluations.

OPT is computationally expensive, but it is useful for deriving knowledge about the relationships among parameters involved in an EA. This procedure can tackle the problem from two perspectives: *specialization* or *generalization* of algorithms. In the first perspective, given an algorithm's configuration, the aim is to find a subset of optimization problems on which this algorithm obtains good results. In the case of *generalization* of algorithms, the aim is to find an algorithm configuration which solves the major quantity of problems with different features [11]. Both perspectives need to deal with the so-called no free lunch (NFL) theorem which proves that, under certain assumptions, no optimization algorithm is superior to any other on all possible optimization problems [12].

During recent years, several tuning methods have been designed to search automatically the most appropriate configuration in the parameter values of the stochastic search based algorithms. One of the first was the sequential parameter optimization (SPO) [13] which is based on statistical techniques such as design and analysis of computer experiments. In [14]–[16], the so-called ParamILS method presented a steady state algorithm that uses an iterated local search to improve only categorical parameters using an initial configuration. The relevance estimation and value calibration (REVAC) method was proposed in [17]. This approach works with a population of parameter configurations for estimating the distribution of the target algorithm. Here, the relevance of each parameter is determined according to the entropy measurement. Examples of tuning tools that work on categorical and numerical parameters at the same time are the iterated racing for configuration (I-RACE) framework [18] and the evolutionary algorithm called EVOCA [19]. I-RACE samples a set of configurations according to a particular distribution and incorporates an iterative racing procedure for selecting the best configurations. EVOCA is a steady state EA that, at each iteration generates two new individuals via wheel-crossover and a local search procedure. It can find good parameter values without requiring an in-depth knowledge of parameter tuning methods.

III. Uncertainty

The typical goal in optimization is to identify optimal solutions. In the case of parameter settings, we want to optimize the configuration $c \in C$ of an MOEA on a scenario O. As defined in section II, O contains several problems $o_{1..k}$ to be optimized in their turn. However, the optimal configuration for solving one problem o_i can be poor for another problem o_j of one scenario O. Thus, the additional challenge is to have a set of configurations that have a 'good' performance in a set of scenarios. The main issue is how to define this performance. To do so, we propose to use *robustness* measures, some of the concepts of *robust* optimization [20], to solve the OPT problem represented by Equation 1.

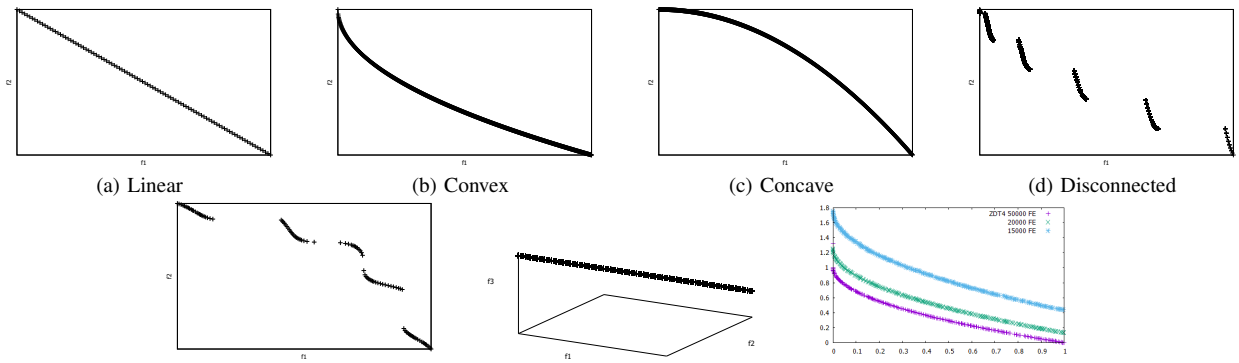
$$\max_{c \in C, o \in O} \psi_A(c, o) \tag{1}$$

In order to do it, we substitute the nominal objective for a robust measure that aggregates on the results from the different problems measured by scalar fitness function ψ. In the following, we present the robust measures that we use in this work.

- Mean: is one of the most used in literature. With this measure, we would be looking for the configuration that works best in the mean of the cases. However, this measure is not indicated when results contain outliers since outlier value has a big impact on the mean value itself.
- Median: is quite useful since it removes outliers. However, removing outliers would mean that we are not interested in all cases but rather in most of them. Thus, using this measure would mean that we are interested in the configuration that works well for most scenarios but it could fail completely in the others.
- Worst case: in this case, the fitness would be represented by the worst result in the given scenario. This would optimize over worst cases and thus ensure that the configuration would work at least with that quality. However, this measure can be over-conservative, since the worst case could never happen in practice.

Each of those measures has some advantages and drawbacks. The decision to use them should include the preference of the decision maker as well as the aim of the algorithm that we are tuning.

Moreover, since the MOEA has uncertainty itself (the same configuration can give different quality), it is also required to treat it in the same form as for the scenario. We first approximate the fitness for the MOEA and afterward for the scenario. Namely, we combine: mean-mean, median-median, best-worst, mean-worst, median-worst and worst-worst. We make an emphasis on the worst-case since we are aiming for one configuration that works in all problem instances.



(a) Linear (b) Convex (c) Concave (d) Disconnected

(e) Mixed (f) Degenerated (g) Multi-frontal

Fig. 1: Different pareto front shapes

IV. Multi-objective optimization

A MOP can be represented as follows:

$$\text{Minimize} \quad \mathbf{f}(\mathbf{x}) := \big(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\big) \tag{2}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{S}, \tag{3}$$

where $\mathcal{S} \subset \mathbb{R}^n$ is the feasible region and each decision variables vector $\mathbf{x} \in \mathcal{S}$ is related to an objective vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$. The solution to a MOP, called the *Pareto optimal set* (POS), is given by:

$$POS := \{\mathbf{x} \in \mathcal{S} \colon \ \nexists \mathbf{y} \in \mathcal{S}, \mathbf{y} \prec \mathbf{x}\}, \tag{4}$$

and its image in $\mathbb{R}^m$, is called the *Pareto optimal front* (POF).

Some MOPs have complicated characteristics that cause difficulties to MOEAs for converging to the POS as a large number of decision variables (large-scale MOPs) or a large number of objective functions (many-objective problems). Furthermore, there are several Pareto front shapes such as linear, convex, concave, mixed, disconnected and degenerated geometries (see Fig. 1). Also, multi-frontal problems can cause a premature convergence for MOEAs, which results in the generation of false Pareto fronts.

There are three main classes of MOEAs: based on Pareto dominance, performance indicators or decomposition. The last class has an efficient search ability to deal with many-objective optimization problems in a low computational time. However, the performance of decomposition-based MOEAs strongly depends on a scalarizing function and a set of target directions. The most popular decomposition-based MOEA is MOEA/D [21], which have been widely used and modified with the aim of improving its performance when solving a wide variety of MOPs.

MOEA/D decomposes a MOP into several single-objective subproblems which are solved in a collaborative manner. Each solution from the population is associated with a search direction (weight vector) to optimize a transformation method called the scalarizing function (SF). Two goals should be accomplished during the search process. The first goal is to minimize the distance between a candidate solution and the reference point (typically, the ideal vector) in order to achieve convergence towards POF. The second goal is the use of different search directions defined by a uniform distribution around all regions in the objective space. It aims to promote diversity for covering the entire POF. MOEA/D establishes neighborhoods at each target direction and applies iteratively a mating selection mechanism and evolutionary operators to enhance individuals. After that, an update process modifies the reference points and the current population according to the SF values. The first version of MOEA/D used the simulated binary crossover (SBX) and polynomial-based mutation (PM).

V. Scalarizing functions

A scalarizing function (SF) is a mathematical programming technique that transforms a MOP into a single-objective one. Its goal is to combine a vector of objective functions $\mathbf{y} = [f_1, \ldots, f_m]^T$ with a reference direction $\boldsymbol{\lambda} = \{\lambda^1, \ldots, \lambda^m\}$ to obtain a scalar value $g(\mathbf{y}) : \mathbb{R}^m \to \mathbb{R}$.

This paper is focused on SFs that can be stated as:

$$\text{minimize} \quad g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda}) \tag{5}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{S}, \tag{6}$$

where $\boldsymbol{\lambda}$ is a weight vector that must satisfy $\sum_{i=1}^{m} \lambda^i = 1$ and $\lambda^i \geq 0$ for all $i \in \{1, \ldots, m\}$. Each component value represents the relative importance assigned to each objective function.

$\mathbf{f}'(\mathbf{x})$ is a transformation function (see equation (7)) to handle negative or incommensurable objective functions.

$$\mathbf{f}'(\mathbf{x}) := \mathbf{f}(\mathbf{x}) - \mathbf{z}^*, \tag{7}$$

where $\mathbf{z}^* = [z_1, \cdots, z_m]^T$ is the ideal point that denotes the optimum value of the ith objective functions of a MOP. An estimation of the ideal point can be obtained by minimizing each of the objective functions individually.

Next, we describe a set of SFs commonly used in decomposition-based MOEAs, and we provide some information that aims to provide a better understanding of the impact of their model parameters for solving MOPs with different Pareto front shapes. Table I shows six SFs compatible with Pareto dominance which can handle convex, linear and concave Pareto front shapes. Both CHE and ASF generate at least weak Pareto optimal solutions and do not require additional parameters. The AASF, PBI, EWC and VADS

TABLE I: Weighted and unconstrained scalarizing functions.

SF	Minimize $g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda}) :=$	Parameter values
Chebyshev (CHE) [22]	$\max_i \{\lambda_i \lvert y_i \rvert\}$	-
Achievement Scalarizing Function (ASF) [23]	$\max\left\{\frac{y_i}{\lambda_i}\right\}$	-
Augmented ASF (AASF) [24]	$\max\left\{\frac{y_i}{\lambda_i}\right\} + \alpha \sum_i \frac{y_i}{\lambda_i}$	small $\alpha > 0$
Penalty Boundary Intersection (PBI) [21]	$d_1 + \theta d_2,$ where $d_1 := \left\lvert \mathbf{y} \cdot \frac{\lambda}{\lVert\lambda\rVert} \right\rvert$ and $d_2 := \left\lVert \mathbf{y} - d_1 \frac{\lambda}{\lVert\lambda\rVert} \right\rVert$	$\theta \in (0, \infty)$ suggested $\theta = 5$
Exponential Weighted Criteria (EWC) [25]	$\sum_i \left(e^{p\,\lambda_i} - 1\right) e^{p\,y_i}$	$p \in [1, \infty)$
Vector Angle Distance Scaling (VADS) [26]	$\frac{\lVert\mathbf{y}\rVert}{\left(\frac{\lambda}{\lVert\lambda\rVert} \cdot \frac{\mathbf{y}}{\lVert\mathbf{y}\rVert}\right)^t}$	$t > 0$ suggested $t = 100$

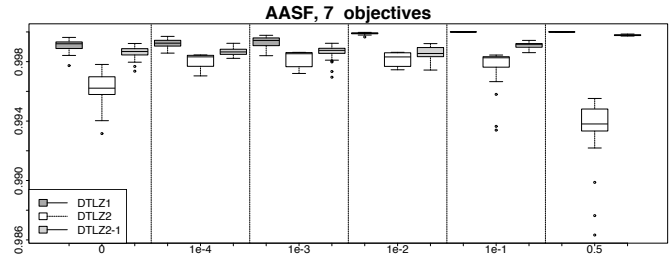
functions can avoid the generation of weak Pareto optimal solutions introducing additional model parameters. However, these parameters are very sensitive to the Pareto front shape and the MOP's dimensionality.

The parameter α in AASF should take small values. In [27], it was recommended to set $\alpha \approx 10^{-4}$. MOEAs that adopt this scalarizing function are [27], [28]. The EWC function requires a large value of p to achieve Pareto optimality, but this can lead to numerical overflow [29]. In [30], EWC was used to solve a problem related to a voltage distribution network. For the VADS function, $t = 100$ was recommended by its authors. The PBI function can produce uniformly distributed solutions in objective space by setting appropriate values for θ. There are several studies [31]–[33] that provided a sensitivity analysis of PBI, varying $\theta \in [0.1, 100]$.

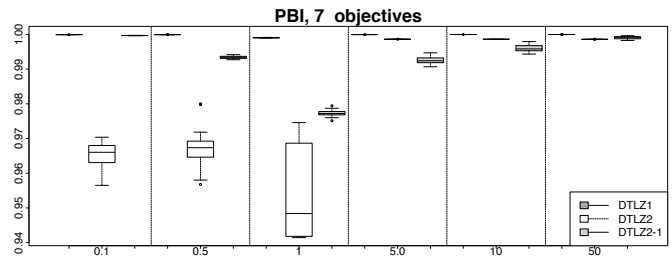
Figure 2 illustrates some boxplots that represent the effect of the parameter sensitivity in the AASF and PBI functions. Here, we use the MOEA/D framework with the same parameter setting and only the SF varied. We compute the normalized hypervolume on 30 independent runs for solving the DTLZ1, DTLZ2 and DTLZ2^{-1} test functions. We can see that one parameter value can be appropriate for a particular Pareto front shape and can work poorly for others. This effect occurs in the same way if we vary the MOP's dimensionality. Such sensitivity to parameter setting justifies the need to find and optimize the 'best' configuration when dealing with several MOPs and same MOEA configuration.

VI. Experimental methodology

The goal of our experiments is to study the effect of robustness measures cause in the OPT methods. To achieve this goal, we use the descriptive statistics mentioned in section III and the tuning tool called evolutionary calibrator (EVOCA) to find the suitable scalarizing function (SF) and its model parameters to reach the maximum performance into MOEA/D framework. Our two particular goals are: 1) to verify which measures are the most useful in three scenarios characterized according to the Pareto front geometry, 2) to identify the most robust configurations when scaling up the number of objective functions in a MOP.



(a) The AASF function with $\alpha = \{0.0, 0.0001, 0.001, 0.01, 0.1, 0.5\}$



(b) The PBI function with $\theta = \{0.1, 0.5, 1.0, 5.0, 10.0, 50.0\}$

Fig. 2: Examples of parameter sensitivity for the AASF and PBI functions. Figures a) and b) show the normalized hypervolume indicator for linear (DTLZ1), concave (DTLZ3) and concave (DTLZ3^{-1}) Pareto fronts.

According to the tuple presented in section II, we use the next elements in our experimental OPT:

- A : The multi-objective evolutionary algorithm based on decomposition (MOEA/D).
- $P := \{CHE, ASF, AASF, EWC, VADS, PBI\}$ with their corresponding model parameter values defined in the ranges $\alpha \in [0, 10.0]$, $p \in [0.1, 10.0]$, $t \in [1, 100]$, and $\theta \in [0.1, 50.0]$.
- $O :=$ DTLZ1, DTLZ3, DTLZ3^{-1} from the *Deb-Thiele-Laumanns-Zitzler (DTLZ)* test suite [7] and the Lamé superspheres (LS) test problems [34] varying the parameter $\gamma = \{0.3, 0.5, 1.0, 2.0, 4.0\}$ with the goal of achieving diverse Pareto front shapes. We classify our test MOPs in three different scenarios according to their Pareto front geometry as follows.
  1) $convex = \{$DTLZ3^{-1}, LS ($\gamma = 0.3$), LS ($\gamma = 0.5$)$\}$
  2) $linear = \{$DTLZ1, LS ($\gamma = 1.0$)$\}$
  3) $concave = \{$DTLZ3, LS ($\gamma = 2.0$), LS ($\gamma = 4.0$)$\}$

  We tested this approach for $m = 3, 5, 7$ objectives. The number of decision variables was set to $n = m + 4$.
- $\psi :=$ the *normalized hypervolume* (NHV), defined as:

$$NHV(A) := \frac{HV}{\prod_i r_i}\,, \tag{8}$$

where $HV(\mathcal{A}, \mathbf{r}) = \mathcal{L}\big(\cup_{i=1}^{\mu} [\mathcal{A}^{(i)}, \mathbf{r}]\big)$ is the hypervolume indicator [35]. $\mathcal{L}$ measures convergence towards the PF and maximum spread through the union of hypercubes formed by all non-dominated elements in $\mathcal{A}$ and a refer-

ence point $\mathbf{r} := (r_1, \ldots, r_m)^T$. It is set to $(2, \ldots, 2)^T$ in all our test problems. A high NHV value is better.

- $b_{max} :=$ The maximum budget established in the EVOCA algorithm is $10,000$ function evaluations and ten seeds per test problem at each generation.

Table II shows the parameter values used by MOEA/D, where m is the number of objectives, n is the number of decision variables, H is a parameter used by the simplex lattice design (SLD) method to generate uniform weight vectors. $popsize$ and NFE are the population size and number of function evaluations used. B is the neighborhood size per weight vector.

TABLE II: The parameter setting used by MOEA/D. The mark p in column H means that the original set of weight vectors generated by SLD is pruned in order to obtain the desirable population size.

m	n	H	$popsize$	NFE	B
3	7	15	136	60,000	27
5	9	6p	180	80,000	36
7	11	4	210	90,000	42

Algorithm 1 shows the general structure of EVOCA while Algorithm 2 shows how we introduce the different robustness measures. Mean, median or worst are aggregation functions that could be used in place of σ or ω.

Algorithm 1: EVOCA(A,P)

Input : Metaheuristic A, Parameters P
Output: Population

```
Pop = Generate Initial Population;
Evaluate each configuration ∈ Pop using Algorithm 2;
while not termination_criterion met do
    New-Pop = Pop;
    Child = Wheel-crossover(Pop);
    Evaluate Child using Algorithm 2 ;
    Replace worst individual by Child in New-Pop;
    Mut-child = Mutation(Child);
    Evaluate Mut-child using Algorithm 2;
    if Mut-Child better than child then
        Replace 2^nd worst individual by Mut-child in
        New-Pop
    Pop=New-Pop;
```

In EVOCA, the chromosome is represented by a string where each element corresponds to a parameter, and its value. Thus, the string length is the number of parameters to be tuned. In our experiments, it is 6 SFs and 4 model parameters. The population size is computed considering the number of parameters to be tuned and their initial domain sizes. The population size adopted in EVOCA is 25.

EVOCA uses two operators: a wheel-crossover operator that constructs one child from the whole population. It uses a roulette wheel procedure [36] to select the value of the gene of each offspring. The child generated replaces the worst individual on the current population. The crossover procedure

Algorithm 2: EvaluateConf(c, O, S, ψ, σ, ω, A) $\in \mathbb{R}$

Input : Configuration c, Scenario O, Seeds S
Fitness func. ψ, Aggregation func. σ, ω,
Algorithm A
Output: evaluate quality of one configuration

```
begin
    Σ ← ∅
    foreach o ∈ O do
        Ω ← ∅
        foreach s ∈ S do
            apf ← A(c, o, s) // MOEA execution
            Ω ← Ω ∪ ψ(apf)
        Σ ← Σ ∪ ω(Ω)
    return σ(Σ)
```

is performed at each iteration. The mutation operator is a hill climbing first improvement procedure, which takes a copy of the child generated by the crossover operator and tries to improve it by modifying one of its parameter values. The mutation operator is always applied. When a numerical parameter is selected, it tries to randomly take a new value within a continuous range that represents the parameter domain. The mutated child generated replaces the second worst individual on the current population, in case of producing a better performance.

VII. Results

Table III shows the scalarizing function obtained by EVOCA for each of the approaches in the different scenarios related to the Pareto front shapes. It is interesting to see that the approaches select not only different parameters for the scalarizing function but also different functions in several cases. This is an interesting result since it shows the impact of changing the robustness measure that is adopted.

For all scenarios, the experimental results are shown in Table IV. In gray, we show the best result among the different robustness measures. An arrow upwards indicates that the approach is outperformed in a significant way by the baseline algorithm (EVOCA with mean-mean). An arrow downwards means that the baseline algorithm outperforms the robustness measure. Furthermore, Figures 3, 4 and 5 show the box plots for the different scenarios using the hypervolume indicator without applying a normalization process. Marks a, b, c, d, e, f correspond to each robustness measures presented in Table IV. We can see in Figure 3 that EVOCA's recommendation for convex scenario outperforms the baseline version only in Lamé supersphere with $\alpha = 0.3$ and $\alpha = 0.5$ using *median-median* and *best-worst* measures. But in multifrontal MOPs *median-median* fails and other measures obtain similar results. This is because the median statistics discard the outlier results. Figure 4 shows the linear scenario, where there is an evident tendency: the *best-worst* measure outperforms the baseline version in MOPs with more than 3 objectives. Finally, we obtain similar results in Figure 5, where the EVOCA's rec-

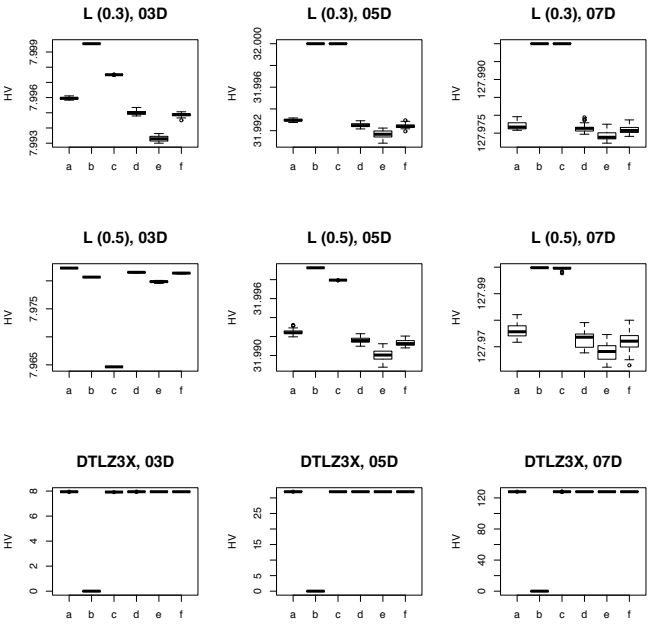


Fig. 3: Scenario with convex geometry and 3, 5, 7 objectives

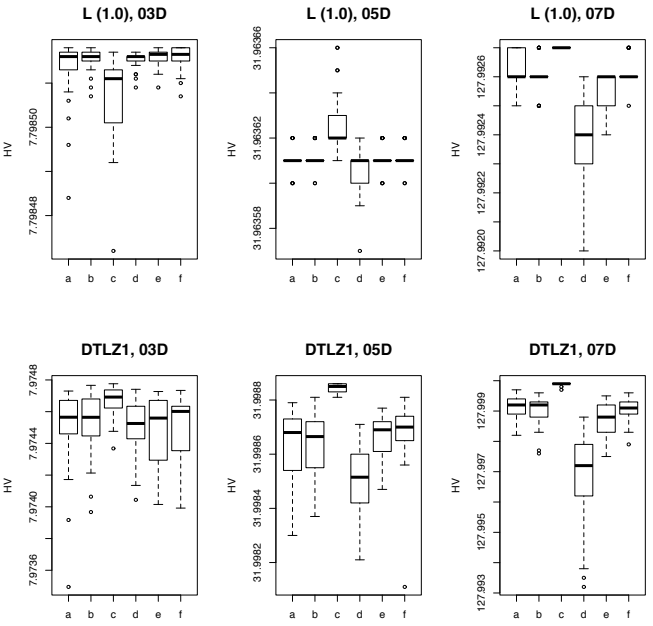


Fig. 4: Scenario with linear geometry and 3, 5, 7 objectives

ommendation improve the baseline version only in unimodal MOPs.

From the results, it is possible to observe that the best-worst approach, is capable of outperforming the base algorithm in most cases when the problems are unimodal. A similar case occurs with the median. However, for the multi-modal problems we can observe a deterioration of the quality of the results. This suggests that such problems should be in a different scenario and have their own configuration to find good solutions.

Also, we can notice that the configurations found work well while increasing the number of objectives. Further, we can observe that the different robustness measures outperform the baseline algorithm in most of the problems (aside from the multi-modal problems). This suggests that alternative measures to treat uncertainty can have a positive impact while searching for configurations for a set of scenarios.

VIII. Conclusions and Future Work

In this work, we have presented a comparative study to analyze the effect of different robustness measures in the offline parameter tuning procedure applied to MOEA/D. We found that AASF showed the best results for the convex case, VADS for the concave case and PBI for the linear case. This shows evidence that the shape of the Pareto front has an effect on the scalarizing function to be used. Furthermore, the values for the scalarizing functions are different among the robustness measures. We have observed that the median and best-worst cases have interesting results for the unimodal problems. Thus, they are worth studying a bit further.

As part of our future work, we are interested in scaling this study with the number of objectives in order to find robust

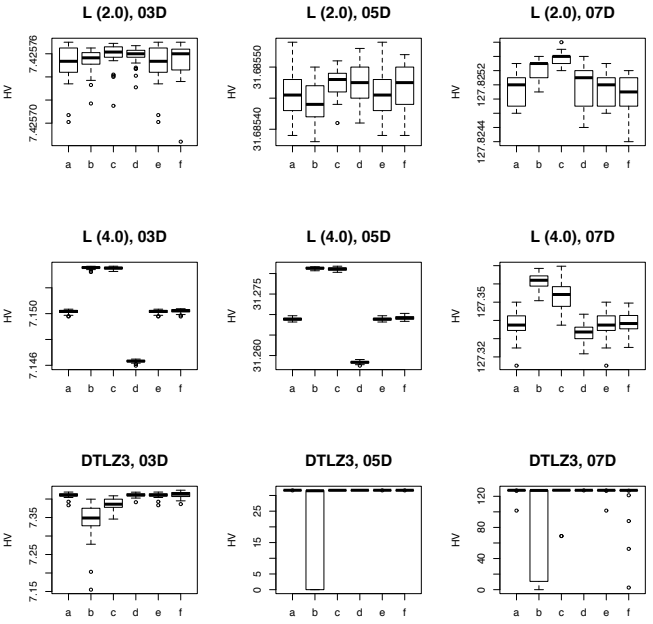


Fig. 5: Scenario with concave geometry and 3, 5, 7 objectives

TABLE III: EVOCA's recommendation for different scenarios using 3,5,7 objectives

Geo	mean-mean	median-median	best-worst	worst-worst	mean-worst	median-worst
Convex	AASF ($\alpha = 5.3727$)	EWC ($p = 7.2$)	CHE	AASF ($\alpha = 1.5305$)	AASF ($\alpha = 0.6977$)	AASF ($\alpha = 1.4065$)
Concave	VADS ($p = 11.9$)	PBI ($\theta = 8.2$)	PBI ($\theta = 2.6$)	VADS ($p = 6.3$)	VADS ($p = 11.9$)	VADS ($p = 12.1$)
Linear	PBI ($\theta = 15.9$)	PBI ($\theta = 10.4$)	AASF ($\alpha = 0.0469$)	PBI ($\theta = 4.2$)	PBI ($\theta = 8.4$)	PBI ($\theta = 11.3$)

TABLE IV: Results reported by EVOCA with different robust measures for different scenarios

Sc	Geo	Dim	mean-mean	median-median	best-worst	worst-worst	mean-worst	med-worst
Convex	L $\gamma = 0.3$	03D	0.99949(7.6989e-06)	↑ 0.99994(2.4944e-08)	↑0.9968(7.6713e-07)	↓0.99937(1.6849e-05)	↓0.99916(2.2555e-05)	↓0.99935(1.4162e-05)
		05D	0.99978(3.6113e-06)	↑ 1.0000(4.0000e-08)	↑0.99999(4.4408e-16)	↓0.99976(5.9568e-06)	↓0.99973(1.1476e-05)	↓0.99976(6.4822e-06)
		07D	0.99982(8.5306e-06)	↑ 1.0000(1.7950e-08)	↑ 1.0000(0.0000e+00)	↓0.99981(8.6050e-06)	↓0.99979(1.0696e-05)	↓0.99981(9.7633e-06)
	L $\gamma = 0.5$	03D	0.99778(3.6635e-06)	↓0.99758(4.4021e-07)	↓0.99557(3.0983e-06)	↓0.99768(9.3282e-06)	↓0.99748(1.5878e-05)	↓0.99767(8.3159e-06)
		05D	0.99976(9.1288e-06)	↑ 0.99997(5.1207e-08)	↑0.99993(2.5377e-07)	↓0.99973(1.0707e-05)	↓0.99968(1.9492e-05)	↓0.99973(1.0752e-05)
		07D	0.99981(2.0659e-05)	↑ 0.99999(3.3306e-16)	↑0.99999(4.4457e-06)	↓0.99978(2.2967e-05)	↓0.99975(2.4902e-05)	↓0.99977(2.8233e-05)
	DTLZ3^{-1}	03D	0.99335(2.0098e-04)	↓0.0000(0.0000e+00)	↓0.99010(3.0459e-04)	0.99334(1.6041e-04)	0.99340(1.3607e-04)	0.99342(7.1454e-05)
		05D	0.99991(1.4226e-05)	↓0.0000(0.0000e+00)	↓0.99978(1.0693e-05)	↓0.99989(2.1022e-05)	↓0.99983(4.3120e-05)	↓0.99989(2.0928e-05)
		07D	0.99996(1.4068e-05)	↓0.0000(0.0000e+00)	↓0.99983(2.1933e-04)	↓0.99995(2.5509e-05)	↓0.99994(2.3587e-05)	↓0.99995(2.0053e-05)
Linear	L $\gamma = 1.0$	03D	0.97481(9.1966e-07)	0.97481(3.1155e-07)	↓0.97481(1.1706e-06)	0.97481(2.3795e-07)	0.97481(2.6042e-07)	0.97481(3.3757e-07)
		05D	0.99886(1.4922e-07)	0.99886(1.4083e-07)	↑ 0.99886(4.1831e-07)	0.99886(2.8511e-07)	0.99886(1.5205e-07)	0.99886(1.3743e-07)
		07D	0.99994(2.8481e-07)	0.99994(4.0573e-07)	↑ 0.99994(3.3306e-16)	↓0.99994(9.7262e-07)	↓0.99994(4.9379e-07)	↑0.99994(2.5289e-07)
	DTLZ1	03D	0.99681(3.2364e-05)	0.99681(2.5263e-05)	↑ 0.99683(1.1418e-05)	0.99681(2.3043e-05)	0.99680(2.7473e-05)	0.99681(2.3141e-05)
		05D	0.99995(3.9159e-06)	0.99995(3.7555e-06)	↑ 0.99996(5.0619e-07)	↓0.99995(3.9932e-06)	0.99995(2.3360e-06)	↑0.99995(3.7648e-06)
		07D	0.99999(2.5451e-06)	0.99999(3.7277e-06)	↑ 0.99999(4.3969e-07)	↓0.99997(1.1753e-05)	↓0.99998(4.1686e-06)	0.99999(2.9046e-06)
Concave	L $\gamma = 2.0$	03D	0.92821(2.0129e-06)	0.92821(1.2526e-06)	↑ 0.92821(1.3764e-06)	↑ 0.92821(1.0095e-06)	↑0.92821(2.0129e-06)	0.92821(2.0137e-06)
		05D	0.99017(1.2045e-06)	0.99017(9.6148e-07)	↑ 0.99017(6.5248e-07)	0.99017(1.0589e-06)	↑0.99017(1.2045e-06)	0.99017(1.1618e-06)
		07D	0.99863(1.5886e-06)	↑0.99863(1.0217e-06)	↑ 0.99863(7.1644e-07)	0.99863(2.2111e-06)	↑0.99863(1.5886e-06)	0.99863(1.9602e-06)
	L $\gamma = 4.0$	03D	0.89376(1.8281e-05)	↑ 0.89419(1.3943e-05)	↑0.89419(1.2420e-05)	↓0.89328(1.2611e-05)	↑0.89376(1.8281e-05)	0.89377(1.8035e-05)
		05D	0.97715(1.1496e-05)	↑ 0.97754(8.2635e-06)	↑0.97753(1.0646e-05)	↓0.97682(1.0701e-05)	↑0.97715(1.1496e-05)	↑0.97716(1.2953e-05)
		07D	0.99482(5.2873e-05)	↑ 0.99500(3.4305e-05)	↑0.99494(6.3347e-05)	↓0.99479(4.2571e-05)	↑0.99482(5.2873e-05)	0.99483(4.2165e-05)
	DTLZ3	03D	0.92623(8.8163e-04)	↓0.91765(6.6721e-03)	↓0.92330(1.9209e-03)	0.92632(6.7448e-04)	↑0.92623(8.8163e-04)	0.92643(1.0158e-03)
		05D	0.98925(3.7990e-04)	↓0.66321(4.5688e-01)	↓0.98887(4.2865e-04)	0.98921(3.2035e-04)	↑ 0.98925(3.7990e-04)	0.98907(4.3676e-04)
		07D	0.99051(3.6548e-02)	↓0.67700(4.3885e-01)	↓0.96755(1.1470e-01)	↑ 0.99781(3.7914e-04)	↑0.99051(3.6548e-02)	0.93313(2.0597e-01)

configurations that scale with the number of objectives. Furthermore, we intend to increase the number of test problems and their characteristics. Our preliminary results suggest that multi-modal problems deserve a special scenario. Finally, we plan to use more sophisticated robustness measures and to use them as a starting point for the design of self-adaptive models.

Acknowledgment

The first author acknowledges support from IPN project no. SIP 20181450, CONACyT and CINVESTAV-IPN to pursue graduate studies in Computer Science. The last author gratefully acknowledges support from CONACyT project no. 221551.

References

[1] H.-G. Beyer and B. Sendhoff, "Robust optimization a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 3334, pp. 3190 – 3218, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782507001259

[2] A. Gaspar-Cunha and J. A. Covas, "Robustness in multi-objective optimization using evolutionary algorithms," *Computational Optimization and Applications*, vol. 39, no. 1, pp. 75–96, Jan 2008. [Online]. Available: https://doi.org/10.1007/s10589-007-9053-9

[3] E. Montero and M.-C. Riff, *Towards a Method for Automatic Algorithm Configuration: A Design Evaluation Using Tuners*. Cham: Springer International Publishing, 2014, pp. 90–99.

[4] A. Menchaca-Mendez, E. Montero, M.-C. Riff, and C. A. C. Coello, *A More Efficient Selection Scheme in iSMS-EMOA*. Cham: Springer International Publishing, 2014, pp. 371–380.

[5] M. Pescador-Rojas, R. Hernández Gómez, E. Montero, N. Rojas-Morales, M.-C. Riff, and C. A. Coello Coello, *An Overview of Weighted and Unconstrained Scalarizing Functions*. Cham: Springer International Publishing, 2017, pp. 499–513.

[6] A. Eiben and S. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 19 – 31, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650211000022

[7] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, October 2006.

[8] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study," in *Parallel Problem Solving from Nature V*, A. E. Eiben, Ed. Amsterdam: Springer-Verlag, September 1998, pp. 292–301.

[9] C. A. Coello Coello and N. Cruz Cortés, "Solving Multiobjective Optimization Problems using an Artificial Immune System," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, June 2005.

[10] M. P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations to the non-dominated set," Technical University of Denmark, Tech. Rep. IMM-REP-1998-7, March 1998.

[11] S. K. Smit and A. E. Eiben, "Parameter tuning of evolutionary algorithms: Generalist vs. specialist," in *Applications of Evolutionary Computation*, C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. I. Esparcia-Alcazar, C.-K. Goh, J. J. Merelo, F. Neri, M. Preuß, J. Togelius, and G. N. Yannakakis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 542–551.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr 1997.

[13] B.-B. T., L. C. W. G., and P. M., "Sequential parameter optimization," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 773–780.

[14] H. Frank, H. H. H., and S. Thomas, "Automatic algorithm configuration based on local search," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'07. AAAI Press, 2007, pp. 1152–1157.

[15] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, "Paramils: An automatic algorithm configuration framework," *J. Artif. Int. Res.*, vol. 36, no. 1, pp. 267–306, Sep. 2009.

[16] H. H. Hoos, *Automated Algorithm Configuration and Parameter Tuning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–71.

[17] V. Nannen and A. E. Eiben, "Relevance estimation and value calibration of evolutionary algorithm parameters," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 975–980.

[18] M. López-Ibáez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Sttzle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43 – 58, 2016.

[19] M. C. Riff and E. Montero, "A New Algorithm for Reducing Metaheuristic Design Effort," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 3283–3290.

[20] B. L. Gorissen, hsan Yankolu, and D. den Hertog, "A practical guide to robust optimization," *Omega*, vol. 53, pp. 124 – 137, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048314001698

[21] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.

[22] V. B. Jr, "On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives," in *Multiple Criteria Decision Making: Proceedings of a Conference Jouy-en-Josas, France May 21–23, 1975*, H. Thiriez and S. Zionts, Eds. Berlin, Germany: Springer, 1973, vol. 130, pp. 76–86.

[23] A. P. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization," in *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter*, G. Fandel and T. Gal, Eds. Berlin, Germany: Springer, August 20-24 1980, pp. 468–486.

[24] K. Miettinen, *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science, 1998, vol. 12.

[25] T. W. Athan and P. Y. Papalambros, "A Note on Weighted Criteria Methods for Compromise Solutions in Multi-objective Optimization," *Engineering Optimization*, vol. 27, no. 2, pp. 155–176, 1996.

[26] E. J. Hughes, "Multiple Single Objective Pareto Sampling," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 4. Canberra, Australia: IEEE Press, December 2003, pp. 2678–2684.

[27] C. C. Tutum and K. Deb, "A Multimodal Approach for Evolutionary Multi-objective Optimization (MEMO): Proof-of-Principle Results," in *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, A. Gaspar-Cunha, C. H. Antunes, and C. Coello Coello, Eds. Guimarães, Portugal: Springer. Lecture Notes in Computer Science Vol. 9018, March 29 - April 1 2015, pp. 3–18.

[28] A. B. Ruiz, R. Saborido, and M. Luque, "A Preference-based Evolutionary Algorithm for Multiobjective Optimization: The Weighting Achievement Scalarizing Function Genetic Algorithm," *J. of Global Optimization*, vol. 62, no. 1, pp. 101–129, may 2015.

[29] R. Marler and J. Arora, "Survey of Multi-objective Optimization Methods for Engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[30] G. Carpinelli, P. Caramia, F. Mottola, and D. Proto, "Exponential Weighted Method and a Compromise Programming Method for Multi-objective Operation of plug-in Vehicle Aggregators in Microgrids," *International Journal of Electrical Power & Energy Systems*, vol. 56, pp. 374 – 384, 2014.

[31] H. Sato, "Inverted PBI in MOEA/D and its Impact on the Search Performance on Multi and Many-Objective Optimization," in *2014 Genetic and Evolutionary Computation Conference (GECCO 2014)*. Vancouver, Canada: ACM Press, July 12-16 2014, pp. 645–652, iSBN 978-1-4503-2662-9.

[32] ——, "Analysis of inverted pbi and comparison with other scalarizing functions in decomposition based moeas," *Journal of Heuristics*, vol. 21, no. 6, pp. 819–849, 2015.

[33] H. Ishibuchi, K. Doi, H. Masuda, and Y. Nojima, "Relation between weight vectors and solutions in moea/d," in *2015 IEEE Symposium Series on Computational Intelligence*, Dec 2015, pp. 861–868.

[34] M. T. Emmerich and A. H. Deutz, "Test Problems Based on Lamé Superspheres," in *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4033. Matshushima, Japan: LectNotesCS, March 2007, pp. 922–936.

[35] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph.D. dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.