Martí, L., García, J., Berlanga, A., Molina, J.M. (2016). A stopping criterion for multi-objective optimization evolutionary algorithms. *Information Sciences*, 367-368, pp. 700-718.

DOI: 10.1016/j.ins.2016.07.025

A Stopping Criterion for Multi-Objective Optimization Evolutionary Algorithms

Luis Martí[c,*], Jesús García[d], Antonio Berlanga[d], José M. Molina[d]

[a]*Institute of Computing,*
Univesidade Federal Fluminense
Av. Gal. Milton Tavares de Souza. Niterói 24210-346 Rio de Janeiro, Brazil.
[b]*Group of Applied Artificial Intelligence,*
Department of Informatics, Universidad Carlos III de Madrid.
Av. de la Universidad Carlos III, 22. Colmenarejo 28270 Madrid, Spain.

[*] Corresponding author.

Email addresses: lmarti@ic.uff.br (Luis Martí), jgherrer@inf.uc3m.es (Jesús García), aberlan@ia.uc3m.es (Antonio Berlanga), molina@ia.uc3m.es (José M. Molina)

URL: http://lmarti.com (Luis Martí)

Preprint submitted to Information Sciences *December 14, 2015*

A Stopping Criterion for Multi-Objective Optimization Evolutionary Algorithms

Luis Martí[c,*], Jesús García[d], Antonio Berlanga[d], José M. Molina[d]

[c]*Institute of Computing,*
Univesidade Federal Fluminense
Av. Gal. Milton Tavares de Souza. Niterói 24210-346 Rio de Janeiro, Brazil.
[d]*Group of Applied Artificial Intelligence,*
Department of Informatics, Universidad Carlos III de Madrid.
Av. de la Universidad Carlos III, 22. Colmenarejo 28270 Madrid, Spain.

Abstract

This paper puts forward a comprehensive study of the design of global stopping criteria for multi-objective optimization. In it we propose a global stopping criterion, which we have named MGBM. MGBM combines a novel progress indicator, called mutual domination rate (MDR) indicator, with a simplified Kalman filter, which is used for evidence-gathering purposes. The MDR indicator, which is also introduced, is a special-purpose solution designed for the purpose of stopping a multi-objective optimization.

In this paper we describe the criterion from a theoretical perspective and examine its performance on a number of test problems. We also compare this method with similar approaches to the issue. The results of these experiments suggest that MGBM is a good starting point for research in this direction.

Keywords: Stopping criteria, progress indicators, multi-objective evolutionary algorithms, multi-objective optimization, Kalman filters

1. Introduction

Most soft-computing, heuristic, non-deterministic or numerical methods all have in common that they need a stopping criterion. The stopping criterion, which

[*]Corresponding author.

Email addresses: lmarti@ic.uff.br (Luis Martí), jgherrer@inf.uc3m.es (Jesús García), aberlan@ia.uc3m.es (Antonio Berlanga), molina@ia.uc3m.es (José M. Molina)

URL: http://lmarti.com (Luis Martí)

is usually a heuristic itself, is responsible for minimizing the wastage of computational resources by detecting scenarios where it makes no sense to continue executing the method.

The success or failure of any practical application relies heavily not only on the techniques applied but also the support methodologies, including the stopping criterion. Paradoxically, this is a matter that has often been overlooked by the community, probably because it plays a supporting part. This relegates the issue to an apparently secondary role. Consequently, the theoretical and practical implications concerning this topic have not yet been properly explored. Indeed, many real-world applications of theoretically outstanding methods may have underperformed due to an incorrect algorithm termination scheme.

Stopping criteria can be grouped into local (iteration-wise) criteria and global (execution-wise) criteria. Local criteria have access only to data pertaining to each iteration of the method. They measure the difference between the current solution and a predefined reference or optimal value and then decide when they are close enough. This type of criterion has the obvious and paradoxical shortcoming of requiring a priori knowledge of the desired optimal value of the solution. This potential weakness has no significant impact if the class of problem being addressed allows the reference value to be replaced by the axis “zero” reference. This applies, for example, to function approximation and other types of problems that can be reduced to an error minimization problem.

On the other hand, global criteria keep track of the process progress across different iterations in order to make decisions relying on the long-term behavior of the algorithm being monitored. This evidence-gathering process has two positive impacts: (i) algorithm progress can be assessed in a relative fashion by comparing the outcome of different iterations and (ii) algorithm progress is more resilient to local optima and noise as it takes into account different iterations.

Evolutionary algorithms (EAs) [1] are a class of population-based metaheuristic optimization methods. They also require a stopping criterion, but the vast majority of applications have bypassed this matter by using a termination scheme that specifies a finite number of iterations. Also, the research [2–5] that has addressed this issue does not appear to have propagated to the rest of the research community.

This is especially applicable to multi-objective optimization evolutionary algorithms (MOEAs) [6, 7]. MOEAs are a type of evolutionary algorithm specially conceived for solving multi-objective optimization problems (MOPs) [8, 9]. MOPs are optimization problems where two or more functions should be jointly optimized. The solution to these problems is a set, known as the Pareto-optimal set, which contains one or more feasible solutions, including the best trade-off values (either maximum or minimum) of the functions.

In the multi-objective case, a local criterion must measure the similarity between the current and the Pareto-optimal front and decide when they are close enough [10]. This type of criterion has the obvious paradoxical shortcoming of requiring an a priori known Pareto-optimal front. On the other hand, global approaches may apply relative improvement metrics that analyze the partial results of the algorithm across iterations. Therefore, there is no need to resort to an absolute comparison with an a priori established threshold. In the particular case of MOEAs, this type of criterion should compare the non-dominated solution fronts yielded by different iterations in order to determine how the optimization process is progressing.

The formulation of an effective criterion is particularly complex in the MOP case, as judging the optimization progress can turn out to be as complex as the optimization itself. In other types of problems, such as function approximation, pattern recognition or single-objective optimization, on the other hand, the axis can be used as a "zero" reference for progress measurement, as previously explained. This approach is inviable for MOPs since its solution is a set of points. Therefore, progress must be assessed in a relative manner using progress indicators rather than the actual solution set. There are a number of quality indicators [11, 12] that can be repurposed for this task, but their high computational cost is an obstacle to their application.

There has been little theoretical research dealing with MOEA convergence [13, 14]. Probably on the above grounds, the formulation of an efficient stopping criterion for MOEAs and other MOP optimizers has been left aside, although it has been repeatedly named as one of the key topics in need of proper attention in the research area [15, 16].

The status of this issue has recently started to change, as interest in these matters has grown. This can be inferred from the relatively large body of research that has gradually started to deal with this question (cf. [17–26]). These papers have led to a substantial improvement in the understanding of the dynamics of evolutionary multi-objective processes. They have put forward different approaches that effectively deal with the stopping criteria issue and that are usable in real-world applications.

In this work we put forward a comprehensive study of the design of a global stopping criteria for multi-objective optimization. We propose a global stopping criterion, which we have called MGBM (after the surnames of the authors). MGBM combines a novel progress indicator, named mutual domination rate (MDR) indicator, with a simplified Kalman filter [27], which is used as an evidence-gathering process. The MDR indicator, which is also introduced here, is a special-purpose solution designed to deal with stopping. It is capable of gauging the progress of the optimization at a low computational cost and is therefore suitable for solving complex or many-objective problems [28].

The viability of the proposal is established by comparing it with some other possible alternatives. In particular, it is compared with the relative versions of the hypervolume indicator and the additive epsilon indicator [12] as progress indicators, and the application of statistical hypothesis testing to evidence assessment.

The theoretical and computational properties of the each of the components are discussed and contrasted. We also run a set of experimental tests. These tests are intended to assess each component combination under different circumstances in order to confirm that the method is capable of detecting "success" and "failure" stopping conditions. In these experiments we address some community-accepted test problems with the elitist non-dominated sorting genetic algorithm (NSGA-II) [29], the improved strength Pareto evolutionary algorithm (SPEA2) [30] and the Pareto envelope-based selection algorithm (PESA) [31].

The main contributions of this paper can be summarized as:

- detailed discussion of the stopping criterion issue and its current state, requirements and problem-solving strategies;
- discussion of different approaches for addressing this issue, and;
- the proposal and testing of a novel stopping criterion.

It should be noted that, although the criteria discussed here are meant for MOPs and MOEAs, they could be easily adapted to other soft computing or numerical methods by replacing the local improvement metric as appropriate.

The rest of this paper is organized as follows. In Section 2, we discuss the background. In Section 3, we dissect and analyze the stopping criteria issue in detail. In Section 4, we present the different components of our method, dealing with the local improvement determination and the evidence-gathering strategies. In Section 5, we discuss and present the experimental results in order to review the properties of the criteria from a practical perspective. Finally, in Section 6 we outline some concluding comments and remarks.

2. Multi-objective optimization

The concept of multi-objective optimization refers to the process of finding one or more feasible solutions to a problem by trading off the equally optimal values of two or more functions subject to a set of constraints.

Stated more formally, a multi-objective optimization problem (MOP) can be defined as:

Definition 1 (Multi-objective Optimization Problem).

$$\left.\begin{array}{rl} \text{minimize} & \boldsymbol{F}(\boldsymbol{x}) = \langle f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})\rangle\,, \\ \text{subject to} & c_1(\boldsymbol{x}), \ldots, c_C(\boldsymbol{x}) \leq 0\,, \\ & d_1(\boldsymbol{x}), \ldots, d_D(\boldsymbol{x}) = 0\,, \\ & \text{with } \boldsymbol{x} \in \mathcal{D}\,, \end{array}\right\} \tag{1}$$

where $\mathcal{D}$ is known as the decision space. *The functions $f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})$ are the* objective functions. *The image set, O, product of the projection of $\mathcal{D}$ through $f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})$ is called* objective space *($\boldsymbol{F} : \mathcal{D} \rightarrow O$). Finally, $c_1(\boldsymbol{x}), \ldots, c_C(\boldsymbol{x}) \leq 0$ and $d_1(\boldsymbol{x}), \ldots, d_D(\boldsymbol{x}) = 0$ express the constraints imposed on the values of $\boldsymbol{x}$.*

In general terms, there is no one optimal solution to this class of problems. Instead an algorithm solving the problem defined in (1) should output a set containing equally good (trade-off) solutions. The optimality of a set of solutions can be defined relying on the so-called *Pareto dominance relation* [32]:

Definition 2 (Pareto Dominance Relation). *For the optimization problem specified in (1) and having $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}$, $\boldsymbol{x}_1$ is said to dominate $\boldsymbol{x}_2$ (expressed as $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$) iff $\forall f_j$, $f_j(\boldsymbol{x}_1) \leq f_j(\boldsymbol{x}_2)$ and $\exists f_i$ such that $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$.*

The solution of (1) is a subset of $\mathcal{D}$ that contains elements that are not dominated by other elements of $\mathcal{D}$.

Definition 3 (Pareto-optimal Set). *The solution of problem (1) is the set $\mathcal{D}^*$ such that $\mathcal{D}^* \subseteq \mathcal{D}$ and $\forall \boldsymbol{x}_1 \in \mathcal{D}^* \nexists \boldsymbol{x}_2 \in \mathcal{D}$ that $\boldsymbol{x}_2 \prec \boldsymbol{x}_1$.*

$\mathcal{D}^*$ is known as the *Pareto-optimal set*, and its image in the objective space is called the *Pareto-optimal front*, O^*.

It is often impossible to find the explicit formulation of $\mathcal{D}^*$. Generally, an algorithm solving (1) yields a discrete non-dominated set, $\mathcal{P}^*$, that approximates $\mathcal{D}^*$. The image of $\mathcal{P}^*$ in objective space, $\mathcal{PF}^*$, is known as the non-dominated front.

2.1. Multi-objective evolutionary algorithms

A number of methods have been used to deal with MOPs [8, 9]. Of these, evolutionary algorithms (EAs) [1] have proved to be a valid and proficient approach from both the theoretical and practical point of view. This has led to what has been called multi-objective optimization evolutionary algorithms (MOEAs) [6, 7]. Their success is due to the fact that EAs do not make any assumptions about the underlying fitness landscape. Another important feature is the parallelism of the search process. Thanks to parallelism the algorithm can output not just one, as many other algorithms do, but a set of equally optimal solutions.

With a view to giving a comprehensive overview of the experimental results to be presented later, let us now briefly describe the MOEAs used and state how they were configured in our experiments.

The MOEAs used in the experiments exploit elitism by explicitly maintaining a subpopulation of non-dominated solutions for comparison by the selection operators. They use density information and Pareto-dominance sorting in order to guide the search, but take different approaches to implementing elitism and density estimation.

The strength Pareto evolutionary algorithm (SPEA) [33] implements elitism by preserving an external population. This population stores a fixed number of non-dominated individuals discovered since the beginning of the simulation. After every iteration of the algorithm, any new non-dominated solution that is found is compared with the solutions present in the external population to preserve the best solutions. It uses the stored solutions along with the dominated solutions in all genetic operations with the hope of inducing a better performance of the solution space search.

Although SPEA has produced a number of significant results, some potential weaknesses have been pointed out [30]. SPEA2 [30] was proposed as an attempt to overcome the limitations of SPEA. It retains the overall scheme of its predecessor, but, in contrast to SPEA, SPEA2 uses a fine-grained fitness assignment strategy that incorporates density information. Furthermore, the size of the external population is fixed; therefore, whenever the number of non-dominated solutions is less than the predefined archive size, the archive is filled up by dominated individuals. Finally, the clustering technique used to prune the external population has been replaced by an alternative truncation method that has similar features but does not miss boundary points.

The NSGA-II algorithm is an improvement on the non-dominated sorting genetic algorithm (NSGA) [34]. There are two key concepts in the NSGA family: fast non-dominated population sorting and a crowding distance calculation to maintain diversity in the population. NSGA-II introduces a faster algorithm to sort the population that requires fewer computations. A crowding distance considers the size of the largest cuboid enclosing each individual without including any other member of the population. This feature is used to maintain diversity in the population, and points belonging to the same front and with a higher crowding distance are assigned a better fitness than points with a smaller crowding distance, meaning that no fitness sharing factor is required.

The Pareto envelope-based selection algorithm (PESA) [31, 35] is a hybrid algorithm combining the Pareto-archived evolution strategy (PAES) [36, 37] and SPEA. It uses a small internal population and a larger external population (archive), where non-dominated solutions found in the main population are stored using a hyper-grid based scheme. The objective space is divided into several regions, and the selection mechanism is based on the degree of crowding in different regions of the archive.

2.2. Quality of solutions

Determining how good a given $\mathcal{P}^*$ is with regard to $\mathcal{D}^*$, or a $\mathcal{PF}^*$ with regard to $\mathcal{O}^*$ is not only a key but also a particularly complex task. It necessarily implies a reduction from an M-dimensional space to a scalar value. Therefore, as in any dimensionality reduction, valuable information could be lost, leading to invalid conclusions. This point has been well documented [10, 11].

Nevertheless, there are some community-accepted indicators of the quality of a solution $\mathcal{P}^*$ [12]. Such indicators can be grouped into three broad categories:

1. distance from the elements of $\mathcal{PF}^*$ to their corresponding closest element of $\mathcal{O}^*$, which measures how close the solution is to the optima;
2. distance from every element of $\mathcal{O}^*$ to its closest element of $\mathcal{PF}^*$, which complements the first class of indicators and expresses how well $\mathcal{PF}^*$ covers $\mathcal{O}^*$, and
3. distribution of the elements of $\mathcal{P}^*$ and $\mathcal{PF}^*$, which gauges how well spread the elements of these sets are.

Although quality indicators were conceived to evaluate the performance of a given optimizer, they can be reformulated to form part of a stopping criterion framework. Binary indicators [10] are best suited for this task as they compare two sets of solutions. Two of these indicators are particularly appropriate: the hypervolume indicator and the additive epsilon indicator.

2.2.1. Hypervolume indicator

The hypervolume indicator, $I_{\text{hyp}}(\mathcal{A})$, [12, 38–40] computes the volume of the region, H, delimited by a given set of points, $\mathcal{A}$, and a set of reference points, $\mathcal{N}$.

$$I_{\text{hyp}}(\mathcal{A}) = \text{volume}\left(\bigcup_{\forall \boldsymbol{x}\in\mathcal{A};\forall \boldsymbol{y}\in\mathcal{N}} \text{hypercube}(\boldsymbol{x},\boldsymbol{y})\right). \tag{2}$$

Therefore, larger values of the indicator will correspond to better solutions.

To measure the absolute performance of an algorithm the reference points should ideally be nadir points. These points are those whose coordinates are the maximum of the individual objective functions over the Pareto set [41]. To contrast the relative performance of two sets of solutions, though, one can be used as the reference set. These matters are further detailed in [10, 12].

Having $\mathcal{N}$, the computation of the indicator is a non-trivial problem. Indeed, its determination is known to be computationally intensive, thus rendering it unsuitable for problems with many objectives.

A lot of research has focused on improving the computational complexity of this indicator [42–46]. According to the most recent results, the indicator is currently known to be $O(n^{M/2} \log n)$. [46] for more than three objectives ($M > 3$); $O(n \log n)$ for $M = 2, 3$ [44].

In order to measure the progress of an algorithm the indicator should be transformed into a relative formulation, as proposed by the binary hypervolume indicator [12]:

$$I_{\text{hyp}}(\mathcal{A}, \mathcal{B}) = I_{\text{hyp}}(\mathcal{A}) - I_{\text{hyp}}(\mathcal{B}) \ . \tag{3}$$

Substituting $\mathcal{A}$ and $\mathcal{B}$ by the non-dominated elements of the current and the previous iteration, $\mathcal{PF}^*_t$ and $\mathcal{PF}^*_{t-1}$, respectively, the indicator can be expressed as

$$I_{\text{hyp}}(t) = I_{\text{hyp}}(\mathcal{PF}^*_t) - I_{\text{hyp}}\left(\mathcal{PF}^*_{t-1}\right) \ . \tag{4}$$

2.2.2. Epsilon indicator

Epsilon indicators [11, 12] are a set of performance indicators that rely on the epsilon dominance concept. This indicator was proposed to measure how close the current non-dominated solution individuals front, $\mathcal{PF}^*_t$, is to the Pareto-optimal front, O^*.

Epsilon dominance is a relaxed version of the domination relation presented in Definition 2. It can be defined in multiplicative and additive terms, but our discussion will be confined to the additive version, as this is the one employed in this paper.

Additive epsilon dominance is defined as:

Definition 4. *Additive Epsilon Dominance Relation. For the optimization problem specified in (1) and having $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}$, $\boldsymbol{x}_1$ is said to additively ϵ-dominate $\boldsymbol{x}_2$ (expressed as $\boldsymbol{x}_1 \preccurlyeq_{\epsilon+} \boldsymbol{x}_2$) iff $f_j(\boldsymbol{x}_1) \leq \epsilon + f_j(\boldsymbol{x}_2)$.*

The additive epsilon indicator, $I_{\epsilon+}$, is a relative indicator that expresses the minimum value of ϵ that is necessary to make a set $\mathcal{A}$ ϵ-dominate a set $\mathcal{B}$, that is,

$$I_{\epsilon+}(\mathcal{A}, \mathcal{B}) = \inf_{\epsilon \in \mathbb{R}} \{\forall \boldsymbol{y} \in \mathcal{B},\ \exists \boldsymbol{x} \in \mathcal{A} \text{ such that } \boldsymbol{x} \preccurlyeq_{\epsilon+} \boldsymbol{y}\} \ . \tag{5}$$

The value of the indicator is to be minimized. If $I_{\epsilon+} < 0$, then $\mathcal{A}$ strictly dominates $\mathcal{B}$. It can be computed in time $O(M\,|\mathcal{A}|\,|\mathcal{B}|)$.

$I_{\epsilon+}$ can be directly applied to progress assessment by substituting $\mathcal{A}$ and $\mathcal{B}$ by the Pareto-optimal sets of two consecutive iterations, $\mathcal{P}^*_t$ and $\mathcal{P}^*_{t-1}$. Nevertheless, it has the weakness that, for some cases,

$$|I_{\epsilon+}(\mathcal{A},\mathcal{B})| \neq |I_{\epsilon+}(\mathcal{B},\mathcal{A})| \ . \tag{6}$$

A reformulation aimed at overcoming this asymmetry would imply an increment in the order of the algorithm complexity.

3. Stopping criteria for MOP optimizers

As already stated in the introduction, the stopping criterion issue has been repeatedly named as one of the key topics requiring proper attention in the multi-objective optimization and MOEAs research areas [15, 16]. Even so, it has been continually neglected. This is not surprising since this matter plays a secondary role compared with the main lines of research in the area. In face of complex real-world problems, though, the lack of a firm theoretical understanding of the problem stands in the way of finding appropriate solutions.

3.1. The problem of knowing when to stop

Typically, the stopping criterion is invoked at the end of an iteration of the algorithm. At that point, it is decided whether algorithm execution should continue or can be aborted. We have identified four scenarios when the execution of an algorithm should terminate:

1. the current solution is satisfactory;
2. the method is able to output a feasible solution, which, although not optimal, is unlikely to be improved;
3. the method is unable to converge to any solution, or
4. the computation already performed is sufficient to reach a solution or further computation is unjustified.

Besides detecting the situations in which the algorithm should be stopped, a stopping criterion should be as lightweight as possible in terms of computational complexity. Computing resources have to be expended on the algorithm itself, and, if the criterion is burdensome, it is more likely to be a weakness than a plus point.

The simplest approach to stopping is to compute how well the current the algorithm state satisfies a given quality threshold. These local (or iteration-wise) criteria only exploit information present in the context of the iteration. Therefore, their analysis horizon is limited, and it is impossible for them to assess the progress of the algorithm across consecutive iterations.

The solution to this is to employ a metaprocess that gathers evidence of progress measured across iterations. This class of global or execution-wise scheme can apply a statistical or machine learning approach to combine the different measurements. A rather popular approach in this regard is to apply a statistical hypothesis test [47, 48].

3.2. Stopping criteria for MOEAs

Stopping criteria are also necessary in the context of single-objective evolutionary algorithms [2]. The theoretical upper and lower bounds for the required number of iterations have been properly established for some classes of genetic algorithms [49, 50]. In more general terms, different local-level criteria have been proposed [4, 5]. They take advantage of the properties of homogeneity of a stagnated population.

Nevertheless, the most common practice to stop a MOEA in the multi-objective case is simply to halt the execution when the algorithm has reached a given number of iterations (scenario 4 in the above list). A survey of papers that perform different sorts of comparative or experimental studies readily illustrates this situation.

Although this class of solution is probably viable for simple problems with two or three objective functions where the required number of iterations can be determined in a quite straightforward way by trial and error, this procedure is computationally unaffordable for more complex problems.

This issue is particularly applicable when dealing with MOPs with a relatively large number of objectives. In these problems, it is unfeasible to estimate the number of iterations required in order to converge to a given solution. Such an estimation would require some knowledge of the nature of the problem or the assumption of some mathematical properties, which contradicts the central idea of evolutionary computation.

In a paper published elsewhere [51], for example, we tuned an interactive multiple model (IMM) filter [52] for use in EUROCONTROL's air traffic management surveillance tracker and server (ARTAS) system. This problem was posed as a 36-objective MOP. Furthermore, the evaluation of each objective was very computationally expensive, they all entailed a Monte Carlo approximation. For this class of problem it is clearly unfeasible to let the optimizer run for too many iterations.

Computing the population homogeneity is not a suitable solution in MOEAs, since the population should spread along the Pareto-optimal front. Therefore, we need to look for novel, purposely designed approaches.

The stopping criterion issue has been previously addressed in the context of MOEAs in [17]. In that work the authors present a stopping criterion to be used in conjunction with the NSGA-II algorithm [29]. They compute a mean stability measure of the spread of the non-dominated individuals. The spread is determined using the crowding distance, which is calculated as part of the NSGA-II optimization process. These spreads are then averaged over a given number of iterations. The decision on whether or not to stop is taken by analyzing the variance. The connection with the NSGA-II algorithm prevents this approach from being easily extended to other algorithms.

A more comprehensive approach was presented by [19]. They proposed a series of statistical tests performed on parallel algorithm executions across a limited number of iterations. These executions are analyzed using three performance indicators, in particular, the generational distance [53], the hypervolume indicator, and the Pareto-optimal solutions spread [54]. The resulting indicator values are contrasted with their corresponding previous values using a statistical hypothesis test. Using these tests the criterion is able to spot situations when no further progress is perceivable. This approach, although theoretically sound, has the drawback of being overparameterized, as it has five free parameters, and having an intensive computational footprint. Overparameterization implies that the user applying the approach has to set the criterion parameters, as well as the correct parameters for the optimization algorithm. The criterion parameters are problem dependent and hard to determine a priori. Therefore, this relatively large number of parameters is an obstacle to the success of the procedure. On the other hand, the need for a large amount of resources to carry out the computation associated with this criterion stands in the way of its application to complex or high-dimensional real-world problems, where, as we already mentioned, the use of stopping criteria is more necessary.

A viable alternative to approaches like the above is to use an online criterion that follows an online analysis scheme. This is the case of the MGBM [18, 22] and the OCD [20, 26] criteria. Both criteria rely on local measurements of progress that are gathered across iterations and used for decision making. MGBM uses a modified Kalman filter and a special-purpose indicator called the mutual domination rate for this purpose. OCD performs a statistical hypothesis test with a window containing the indicators yielded by recent iterations.

4. The MGBM stopping criterion

As mentioned in Section 3, a stopping criterion should be composed of two components. One component measures the improvement in the solutions obtained after an iteration and the other keeps track of these measurements in order to decide whether or not the execution of the algorithm should be stopped.

MGBM is a global criterion that combines a local improvement indicator, called the mutual domination rate (MDR) indicator, and a global evidence-gathering criterion that decides when the evolution of values yielded by the local metric indicates that the algorithm should be stopped. The local indicator contrasts the non-dominated individuals of the current and preceding iterations in order to compute a measure of the improvement produced by the current iteration. This indicator is discussed in detail in the next section. The evidence-gathering process tracks the values of the indicator across iterations using a Kalman filter (described in Section 4.2). The Kalman filter settings are unusual, as it is not designed to predict the outcome of the indicator across iterations. Instead, it is used to detect situations where no further progress will be made. Section 4.2.2 presents the fine points of this matter.

4.1. Mutual domination rate indicator

Intuitively, the performance assessment area, briefly introduced in Section 2, provides a natural grounding for addressing this issue. Performance indicators, although designed to determine how similar a solution is to the Pareto-optimal front, can be reformulated to compare two solutions output by two consecutive iterations. The main drawback of directly applying this class of solution is the high computational complexity of the indicators.

The solution to this problem is to create an indicator from scratch designed specially for the intended purpose. This was the idea that prompted the formulation of the mutual domination rate (MDR) indicator.

To measure the progress of the evidence-gathering process, we use a metric based on the set of non-dominated solutions of two consecutive iterations, $\mathcal{P}^*_t$ and $\mathcal{P}^*_{t-1}$.

In order to simplify the explanation we introduce the $\Delta\left(\mathcal{A},\mathcal{B}\right)$ function that returns the set of elements of $\mathcal{A}$ that are dominated by at least one element of $\mathcal{B}$. Expressed more formally,

$$C = \Delta\left(\mathcal{A},\mathcal{B}\right)\,, \tag{7}$$

such that

$$\forall \boldsymbol{x} \in C, \boldsymbol{x} \in \mathcal{A},\ \text{and}\ \exists \boldsymbol{y} \in \mathcal{B}\ \text{with}\ \boldsymbol{y} \prec \boldsymbol{x}\,. \tag{8}$$

The progress indicator $I_{\text{mdr}}\left(t\right) \in [-1,1]$ contrasts how many non-dominated individuals of iteration $t-1$ are dominated by the non-dominated individuals of iteration t and vice versa,

$$I_{\text{mdr}}\left(\mathcal{P}^*_t,\mathcal{P}^*_{t-1}\right) = \frac{\|\Delta\left(\mathcal{P}^*_{t-1},\mathcal{P}^*_t\right)\|}{\|\mathcal{P}^*_{t-1}\|} - \frac{\|\Delta\left(\mathcal{P}^*_t,\mathcal{P}^*_{t-1}\right)\|}{\|\mathcal{P}^*_t\|}\,. \tag{9}$$

The I_{mdr} indicator provides different types of information. If $I_{\text{mdr}} = 1$, the entire population of iteration t is better than its predecessor. If $I_{\text{mdr}} = 0$, there has not been any substantial progress. The worst case, $I_{\text{mdr}} = -1$, indicates that iteration t has not improved any of the solutions of its predecessor.

As we mentioned in the introduction of this paper, a stopping criterion should be able to discover three situations where the algorithm execution should be stopped. From the stopping point of view, all these situations can be interpreted as whether or not the algorithm has made progress. Note that if we focus on the dominance relation, we disregard the spread of elements along the Pareto-optimal front. This means that the algorithm will be able to detect when there is no improvement towards this front but not when the spread of solutions along the Pareto-optimal front is better.

The determination of the non-dominated individuals after each iteration can be computationally expensive. As most MOEAs extract such individuals for their own purposes, however, it would be reasonable to embed this part of the criterion into the actual MOEAs.

Having the current and previous non-dominated sets $\mathcal{P}^*_t$ and $\mathcal{P}^*_{t-1}$, the order of complexity of calculating I_{mdr} is $O\left(M \cdot \left|\mathcal{P}^*_t\right| \cdot \left|\mathcal{P}^*_{t-1}\right|\right)$.

4.2. Gathering evidence

Our approach is based on the recursive estimation prediction and update framework proposed by Kalman filters. For this reason, we will assume that there is no correlation between the noise present in the measured progress indicator in consecutive iterations. Furthermore, the estimated value of the progress indicator and its associated covariance are governed by a Markov process, and therefore the outcome of each iteration depends on the previous iteration only.

4.2.1. Kalman filters

The Kalman filter [27, 55] provides an efficient computational means to estimate the state of a dynamic system from a series of incomplete and noisy measurements. This filter is the linear estimator with minimum squared error that can be applied to any dynamic system with errors following any distribution where the two first moments of the distribution are known. Furthermore, if we know that probability distributions are Gaussian and the system dynamics are linear, the Kalman filter is the globally optimal state estimator. It is very powerful since it supports estimations of past, current, and future states, even when some aspects of the modelled system are unknown.

The Kalman filter addresses the general problem of estimating the state of a discrete-time controlled process that is ruled by a linear stochastic difference equation.

The state of the filter is represented by two variables:

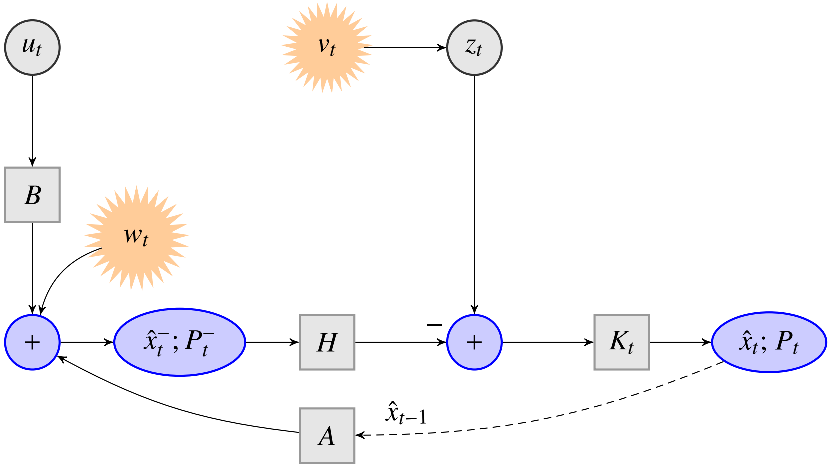


Figure 1: Schematic representation of a Kalman filter iteration. $\hat{x}_{t-1}^-$, u_t and process noise are use to compute a priori estimation $\hat{x}_t^-$, and its error covariance, P_t^-. A prediction is made and subtracted from measurement z_t to calculate an error vector. This error is multiplied by the Kalman gain K_t, that generates a correction added to the prediction to yield the final estimate $\hat{x}_t$.

- $\hat{x}_t$, the estimate of the state at time t, and
- P_t, the error covariance matrix, which is a measure of the estimated accuracy of the current state estimate.

The Kalman filter estimates a process state by a recursive feedback control that can be separated in the *prediction* and *update* phases. The prediction phase is responsible for making an *a priori estimation* of the future state of the system relying on the current state and error covariance estimates. The update phase is responsible for feeding back the (noisy) measurement of the state of the system to output an improved *a posteriori estimate*. Figure 1 summarizes these processes in an schematic form.

The Kalman filter assumes a dynamic model given by

$$x_t = Ax_{t-1} + Bu_t + w_t \,, \tag{10}$$

where u_t is an optional control input and the random variables $w_t \sim N(0, Q)$ represent the process noise.

Additionally, the measurement process is modeled by

$$z_t = Hx_t + v_t \,, \tag{11}$$

where H relates the real state of the process x_t to the measurement z_t and $v_t \sim N(0, R)$ is the measurement noise.

First, the a priori estimation, $\hat{x}_t^-$, and its error covariance, P_t^-, are calculated as

$$\begin{aligned} \hat{x}_t^- &= A\hat{x}_{t-1} + Bu_t\,, \qquad (12)\\ P_t^- &= AP_{t-1}A^T + Q\,. \qquad (13)\end{aligned}$$

Then the update phase proceeds by computing the Kalman gain,

$$K_t = \frac{P_t^- H^T}{HP_t^- H^T + R}\,. \qquad (14)$$

The a posteriori estimation is calculated as the feedback is entered in the filter as

$$\hat{x}_t = \hat{x}_t^- + K_t\left(z_t - H\hat{x}_t^-\right)\,. \qquad (15)$$

Finally, an a posteriori error covariance estimate is output by

$$P_t = (I - K_t H)P_t^-\,, \qquad (16)$$

where I is the identity matrix.

4.2.2. Using Kalman filters to gather evidence

The application of Kalman filters is an unconventional approach to evidence gathering. Instead of trying to predict the outcome of a given variable across time, we are interested in detecting when a variable (in this case the MDR indicator) has stabilized around zero. Due to its recursive formulation, the estimated variable at time t summarizes all the evidence gathered until then, plus the associated covariance error, P_t, which would be the minimum possible error under linear conditions.

For this reason, in our case, we keep track of the algorithm progress indicator at iteration t, $I_{\text{mbr}}(\mathcal{P}_t^*, \mathcal{P}_{t-1}^*)$.

After each iteration, we compute the *a priori* estimated indicator $\hat{I}_t^-$ using a simple version of the dynamic model (10) with $A = 1$ and $B = 0$. This implies that we are taking a positivist stance and predicting that the indicator will remain constant across iterations and, therefore, be equal to the *a posteriori* estimation, $\hat{I}_{t-1}$,

$$\hat{I}_t^- = \hat{I}_{t-1}\,, \qquad (17)$$

disregarding the control input, u, as there is no direct information on changes of I.

In this simplified prediction model, assuming stable conditions between $t-1$ and t, we have decided to disregard the prediction error in our dynamic model ($Q = 0$). The neglect of the plant noise covariance matrix Q is assumable, provided that the deviation from the linear approximation is much less than the measurement noise (R). This is valid if the dynamics of the convergence process is smooth, which will be true in the last phase of convergence of any search algorithm under normal conditions. On the other hand, under these assumptions, we would not be rigorously modeling the behavior during the initial transient period. However, as the estimation is not intended to be used to track the values of I, but to detect when it has reached a stable state that represents algorithm stagnation, we are interested in having a precise model for the latter part of the convergence process.

Correspondingly, the a priori error covariance becomes

$$P_t^- = P_{t-1}\,. \tag{18}$$

We then rewrite (11) as

$$\begin{aligned} z_t &= I_{\mathrm{mdr}}\left(\mathcal{P}_t^*, \mathcal{P}_{t-1}^*\right) & (19)\\ &= I_t + v_t\,, v_t \sim N(0, R), & (20)\end{aligned}$$

where $I_{\mathrm{mdr}}(\mathcal{P}_t^*, \mathcal{P}_{t-1}^*)$ is calculated following (9). Here we assume that $I_{\mathrm{mdr}}()$ is affected by a Gaussian process attributable to the search process taking place as part of the evolutionary algorithm.

The correction step of the process becomes

$$K_t = \frac{P_t^-}{P_t^- + R}\,. \tag{21}$$

Here R can be interpreted as the rate at which the criterion will take into account a single measurement and therefore provide a faster reaction to changes or if, on the contrary, the criterion is biased toward a more global (or more inertial) approach.

Therefore, the a posteriori estimation of the indicator can be expressed as the current result of the indicator

$$\hat{I}_t = \hat{I}_t^- + K_t(z_t - \hat{I}_t^-)\,. \tag{22}$$

The above assumptions merit further discussion, as they imply an alternative use of Kalman filters and are, therefore, likely to lead to a misunderstanding of the inner workings of the criterion. As already discussed, Kalman filters are generally used for estimating the time-sequence values of a definite variable from a set of (noisy) measurements. This is not the case here. In this case, we have configured the filter to capture a “no-progress state” where the evolutionary search process has stagnated. Therefore, the criterion recognizes when the dynamics of the evolutionary process matches the no-progress state represented by the filter.

One main concern is to assure that the algorithm will not stop too early. It can be assumed that there will be temporary stagnation scenarios in the early stages of the evolutionary process that should not be taken into account. The value of R

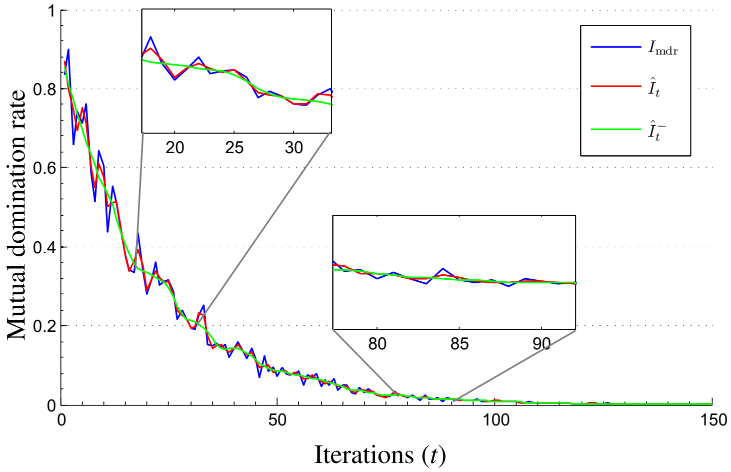


Figure 2: Evolution of the MDR indicator, $I_{\mathrm{mdr}}\left(\mathcal{P}_t^*, \mathcal{P}_{t-1}^*\right)$, and the a priori and a posteriori estimations $\hat{I}_t^-$ and $\hat{I}_t$ across iterations. Here the NSGA-II algorithm is supervised as it successfully solves the DTLZ3 problem (see Section 5 for more details).

plays a key role for this purpose. By assuming $Q = 0$, P_t^- and K_t will converge to zero at a rate that depends on the values of P_0 and R, as detailed in the following section. This implies that R controls how sensitive the criterion will be in the initial part of the execution. This assumption has been validated in [24], where we analyzed different alternatives for Q and some adaptive configurations. A noteworthy conclusion drawn from the above research is that the selection of different values of Q had little impact on the stopping decision.

Figure 2 contains plots of the values of the MDR indicator, $I_{\mathrm{mdr}}\left(\mathcal{P}_t^*, \mathcal{P}_{t-1}^*\right)$, and the a priori and a posteriori estimations, $\hat{I}_t^-$ and $\hat{I}_t$, used in an NSGA-II run to solve the DTLZ3 problem (see Section 5 for details). These plots illustrate how the three values interact with each other across the algorithm iterations and how the a posteriori indicator smooths out the readings yielded by the indicator with a definite shift in time.

There are different situations where the values of the indicator and the estimators suggest that algorithm execution can be safely stopped. For the MGBM criterion we chose a scheme that will activate if the a posteriori estimation $\hat{I}_t$ and associated confidence interval falls below a definite threshold

$$\hat{I}_t + 2\sqrt{P_t} < \hat{I}_{\min}\,. \tag{23}$$

In particular, as we are interested in stopping when no further progress is predicted, it should stop in a situation that is represented by $\hat{I}_{\min} = \varepsilon$, with $\varepsilon \to 0$.

4.2.3. Convergence of Kalman estimator

The assumed model basically exploits the fact that the process noise is much lower than the indicator noise ($Q \ll R$). This is valid if dynamic evolution of indicator is slow and the observation noise clearly dominates the uncertainty in the estimator.

With this simplification, and also making use of the quite simple relations indicated above (one dimensional estimation with $A = 1$ and $B = 0$), the Kalman filter equations particularize to friendly analytical expressions. After substitution of Kalman gain (21) in estimated covariance (16) the recursion of estimated covariance becomes

$$P_{t+1} = \frac{R}{R + P_t} P_t \,. \tag{24}$$

This can be more conveniently expressed as

$$P_{t+1}^{-1} = P_t^{-1} + R^{-1} \,. \tag{25}$$

In this form the covariance dependence with time can be explicitly computed. After t iterations, assuming it was initialized in $t = 0$ with value P_0, it is given by

$$P_t^{-1} = P_0^{-1} + tR^{-1} \,, \tag{26}$$

which is equivalent to

$$P_t = \frac{RP_0}{tP_0 + R} \,, \tag{27}$$

and the Kalman gain at iteration t particularizes to

$$K_t = \frac{P_0}{tP_0 + R} \,. \tag{28}$$

Finally, in order to remove the dependence of criterion with the initialization covariance, we have used the typical initialization with the observation noise for the first estimate ($P_0 = R$). In that case, previous expressions are further simplified to

$$P_t = \frac{R}{t + 1} \,; \tag{29}$$

$$K_t = \frac{1}{t + 1} \,. \tag{30}$$

As we can see, this particular Kalman filter converges to zero gain (as any Kalman filter with process noise $Q = 0$), but the convergence rate is quite slow. The confidence interval associated to the estimated progress indicator is given by

$$I_{95\%} = \hat{I}_t \pm 2\sqrt{\frac{R}{t + 1}} \,. \tag{31}$$

Therefore, although the gain could become zero and theoretically make the filter not to follow the indicator trajectory, the number of generations needed would be higher than conventional number (several hundreds), which never happened in practice. In all the forthcoming experimental analysis, the estimator converged much earlier with the three values employed for parameter R (0.05, 0.1 and 0.15). An additional protection could avoid the use of this estimator after a reasonable minimum number of generations, $t_{\min}$, for instance $t_{\min} = 500$ would be out of the unstable region. However, using this approach would not ultimately solve the problem as the values of $t_{\min}$ would be problem-dependant.

4.3. Algorithmics of the criterion

Relying on the equations introduced above we can formulate the algorithmic scheme of our stopping criterion. This algorithm is outlined in Figure 3.

Apart from the positivist stance expressed above in the formulation of (17), we will use an initial a posteriori progress estimation, $\hat{I}_0$, equal to 1 and the associated initial covariance, P_0, equal to R. This means that we will be assuming full progress from the start and will let this indicator decay as the process advances.

On the other hand, we have not yet demonstrated that our criterion converges and, therefore, there is no theoretical guarantee of the optimization process stopping. This implies that we have to set a maximum limit on the number of iterations, $t_{\max}$, as a safety measure.

The remaining issue is the choice of the process noise variance R, which, in our case, represents the degree of system inertia. As this is the only free parameter of the criterion, an incorrect choice could lead to undesired behavior. In the next section we show how criterion performance and robustness vary for different values of R.

Due to the particular assumptions enforced in the dynamic model, the values of Kalman gain K_t can be directly computed with the equations presented in previous section in order to speed up computation during the execution of the evolutionary algorithm.

5. Experiments

We now experimentally illustrate the accuracy of the MGBM criterion by analyzing the performance of the estimations in a set of experiments. In order to establish the validity of MGBM we proposed two experiments. One is designed to gain a proper understanding of the properties of MBGM and its controlling parameter. The other compares MBGM with similar approaches with the aim of testing its validity and viability.

```
Initialize $t = 0$ and the a posteriori progress estimation $\hat{I}_0 = 1$.
Set $R$.
Set $t_{\max}$, the maximum number of iterations.
Set $\hat{I}_{\min}$, the minimum accepted value of the a posteriori estimation.
**while** $\hat{I}_t \geq \hat{I}_{\min}$ and $t < t_{\max}$ **do**
  Execute one iteration of the MOEA.
  $t = t + 1$.
  Compute the a priori progress estimation, $\hat{I}_t^-$, following (17).
  Calculate measured rate of improvement, $z_t$, as specified in (9) and (19).
  Determine the a posteriori estimation $\hat{I}_t$ from equations (18)-(22).
**end while**
```

Figure 3: Algorithmic description of the MGBM stopping criterion.

In particular we will present the results of applying three well-established MOEAs —-the elitist non-dominated sorting genetic algorithm II (NSGA-II) [29], the improved strength Pareto evolutionary algorithm (SPEA2) [30] and the Pareto envelope-based selection algorithm (PESA) [31]—– to solve three scalable multi-objective test problems —–DTLZ3, DTLZ6 and DTLZ7 [56]—– under different initial conditions. The choice of initial conditions is intended to bias the algorithm a priori so that we can test whether our criterion can resolve all possible target situations, i.e. either success or failure.

5.1. Shared experimental setup

As both experiments address the same test problems using the same MOEAs under the same biasing conditions, we will now describe the shared characteristics of both experiments.

5.1.1. Test problems

The DTLZ3, DTLZ6 and DTLZ7 problems are part of a family of scalable multi-objective test problems originally introduced to study and compare the performance of different MOEAs on high-dimensional problems.

These problems were selected for the experiments because of the relative simplicity of their specification and the existence of an a priori known Pareto-optimal front.

The DTLZ3 problem is an M-objective problem with an n-dimensional decision vector. Its Pareto-optimal front lies on the first orthant of a unit hypersphere This problem was introduced to test the ability of a MOEA to converge to the global Pareto-optimal front, since there are $3^{n-M+1} - 1$ parallel suboptimal fronts.

The DTLZ6 problem is actually based on a simpler problem, in this case the DTLZ5 problem. As in the previous case, suboptimal fronts are also present with the intention of deceiving the optimizer.

On the other hand, the DTLZ7 problem has a Pareto-optimal front that consists of a heavily disconnected set of 2^{M-1} Pareto-optimal regions that test an algorithm's ability to maintain a robust coverage of all optimal regions.

The first results have to do with the WFG4 problem. WFG4 is a separable and strongly multi-modal problem that, like the other problems, has a concave Pareto-optimal front. This front lies on the first orthant of a hypersphere of radius one located at the origin.

WFG5 is also a separable problem but it has a set of deceptive locally optimal fronts. This feature is meant to evaluate the capacity of the optimizers to avoid getting trapped in local optima. The next problem, WFG6, is a separable problem without the strong multi-modality of WFG4.

The remaining three problems have the added difficulty of having a parameter-based bias. WFG7 is uni-modal and separable, like WFG4 and WFG6. WFG8 is a non-separable problem while WFG9 is non-separable, multi-modal and has deceptive local-optima.

5.1.2. Biasing the optimization outcome

As mentioned in Section 3, a stopping criterion should spot the conditions in which the execution of its associated algorithm should be terminated because it was either successful or failed to reach any solutions. To explore how good the stopping criteria is at doing this, the experiment parameters should be configured in such a way that the outcome of the optimization process is a priori biased towards a success or a failure.

For the success-biased experiments, we used three-dimensional problems ($M = 3$). The population size was set to 100 elements and the algorithms were left to run for 500 iterations. For the sake of reproducibility and to compare results, the internal parameters that we used in this study were the same as the values described in [57] (see Table 1). In that paper and in the preliminary exploratory experiments that we ran, this configuration was shown to correctly solve the three problems under study.

For failure biasing, the problems were configured with ten objectives ($M = 10$) and the other experimental conditions were unchanged. A series of experimental studies, including [58–60] and [6, pp.414–419], showed that there is an exponential dependence between the dimension of the objective space and the population size required to solve the problem correctly. When this ratio is not met because the population is smaller than it should be, then most of the population becomes non-dominated, and dominance-based ranking becomes useless, as it is unable to guide

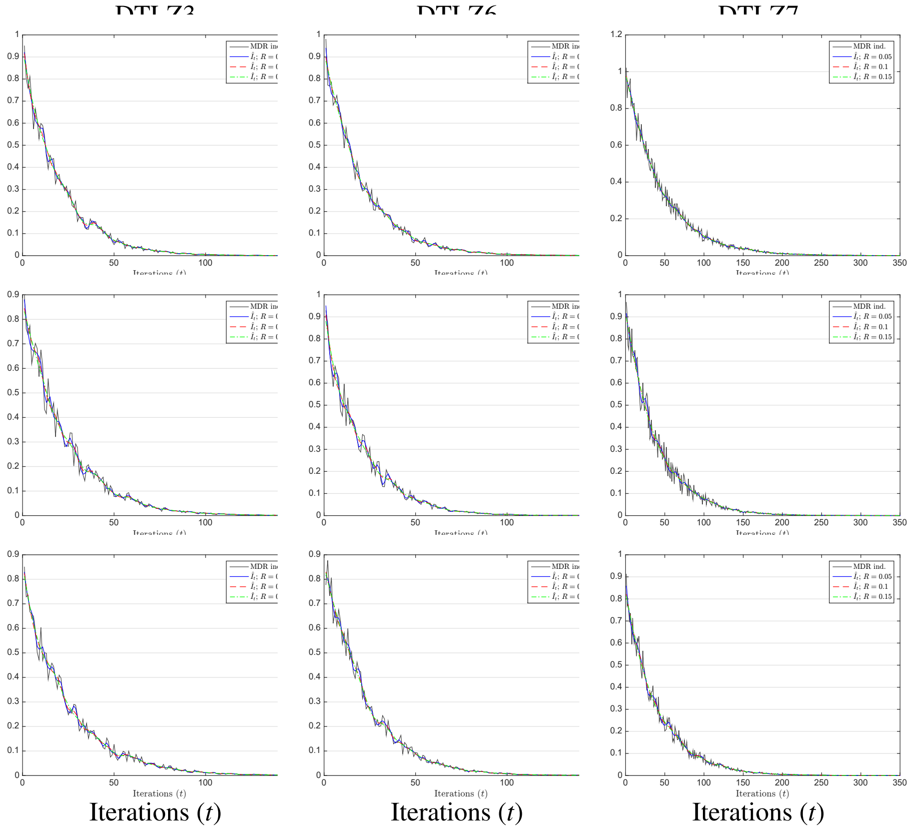


Figure 4: Mean values after each algorithm' iterations of the MDR progress indicator and the a posteriori estimation of progress, $\hat{I}_t$, for different values of R in a biased success experimental setup.

the search. The selected population size/number of objectives ratio has been shown to exhibit this behavior [61]. Furthermore, some preliminary experiments were carried out in order to corroborate this point.

Experiments were carried out within the PISA [62] framework. An Intel Quad Core 3.4 GHz personal computer with 4 GB of RAM memory running the Linux operating system was used. The results reported here were output after 30 independent runs of the algorithms solving each of the problems.

5.2. *Understanding MGBM*

The purpose of this experiment was to explore the processes that take place under the hood of MGBM. We were particularly interested in observing how the Kalman filter keeps track of the evidence of progress provided by the indicator. Similarly, we wanted to study the impact of R, the free parameter of the criterion.

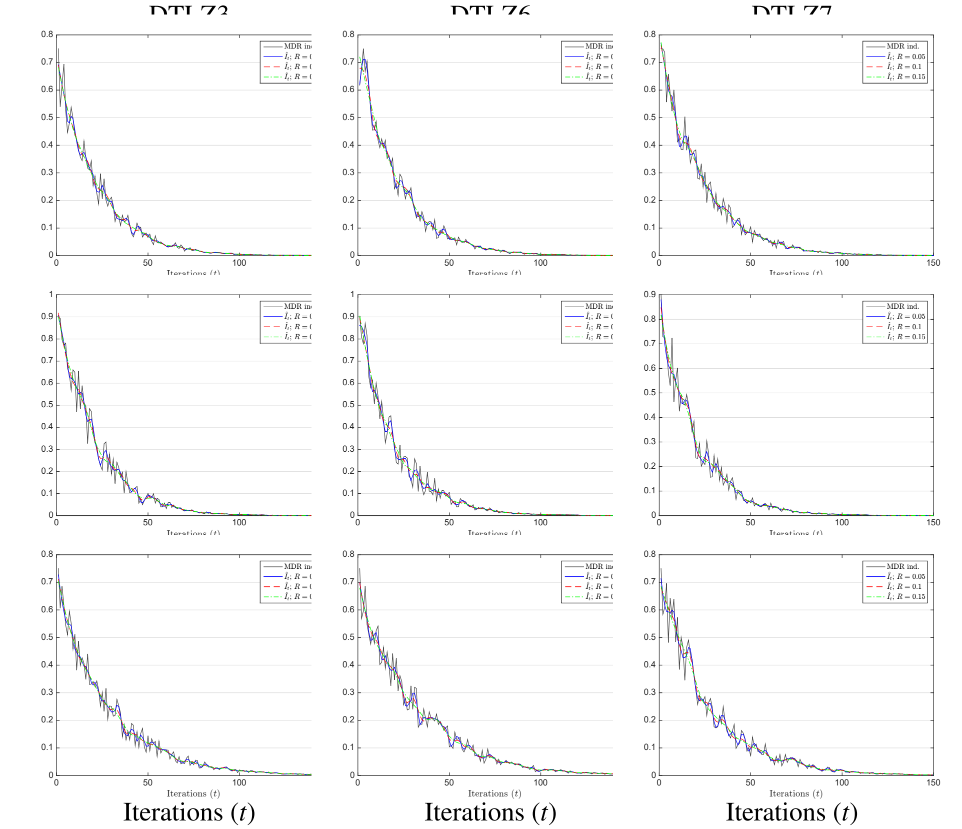


Figure 5: Evolution as iterations advance of the MDR progress indicator and the a posteriori estimation of progress, $\hat{I}_t$, for different values of R in a biased failure experimental setup. The plots represent the mean values of the quantities involved.

For this purpose, we applied MGBM with different values of R, in particular, $R = 0.05$, $R = 0.1$ and $R = 0.15$. The stopping threshold was set to $\varepsilon = 0.0001$.

Figure 4 summarizes the evolution of the a posteriori estimators, $\hat{I}_t$, output when analyzing the execution of the three MOEAs solving the three test problems under study. In all cases the stopping condition was met when the algorithms became stable and their solutions were as close to the Pareto-optimal front as they would be at later generations. The criterion response was quicker or more inertial depending on the value of R. In the following experiment the performance indicator values of the solutions were measured at the iterations marked by the criterion. We found that the indicator values derived from the criterion are similar to the values output at the later algorithm execution stages.

Note that, in similar tests performed in [57], the algorithms were left to run for more iterations than suggested by the criterion. A set of analogous experiments were performed in [56] with the same population size as ours but with unspecified internal parameters. For DTLZ3 and DTLZ6, our criterion also suggested halting the optimization with fewer iterations than they used. However, in the case of DTLZ7, the criterion suggested keeping the processes running for a longer than used in the above research, indicating that further processing was needed to reach the optimum. These results are summarized in Table 2.

Although the criterion appears to signal the algorithm to stop iterating earlier than in previous tests, these results raise a logical question. Are the solutions output at the iterations where the algorithm was stopped as good as the solutions output at the end of the simulation? Figure 6 summarizes the mean hypervolume and additive epsilon indicator values measured at the iterations where the criterion fired and at the end of each execution of the algorithm. It indicates that there is no substantial difference between the quality of the solutions in the iterations selected by the criterion and the final iterations of the algorithms. One interesting feature is that the additive epsilon indicator values appear to be more homogeneous than the hypervolume values. This can be attributed to the fact that in later stages of the execution the solutions are improved in terms of diversity. This improvement is better captured by the hypervolume indicator.

The failure-biased experiments (see Figure 5) complement the above results. The behaviour of MGBM configured with the different values of R has similar characteristics as the success-biased case, only that in these experiments fewer iterations were required. This is an important conclusion, as in these experiments the criterion was supposed to stop an optimization process that would not yield a valid solution and, as consequence any leaving the process to proceed would imply a waste of resources.

5.3. MGBM and its possible alternatives

In order to establish the validity and viability of MGBM, we proposed an experiment that contrasts MBGM with possible alternative approaches. The set of binary quality indicators formulated for MOP solvers includes possible alternatives to the MDR indicator. We have chosen two popular indicators that were described in Section 2: the binary forms of the hypervolume and additive epsilon indicators [12].

Similarly, we chose a statistical hypothesis test scheme for comparison with the Kalman-based approach. In our case we have taken an approach based on the online convergence detection (OCD) method [20, 26]. This method applies a one-sided χ^2 test [63] to determine if the variance of the measured indicator is below a set threshold, and a two-sided t-test [64] to establish the linear trend of the indicator values. In our experiments we conformed to the parameter setup suggested by the

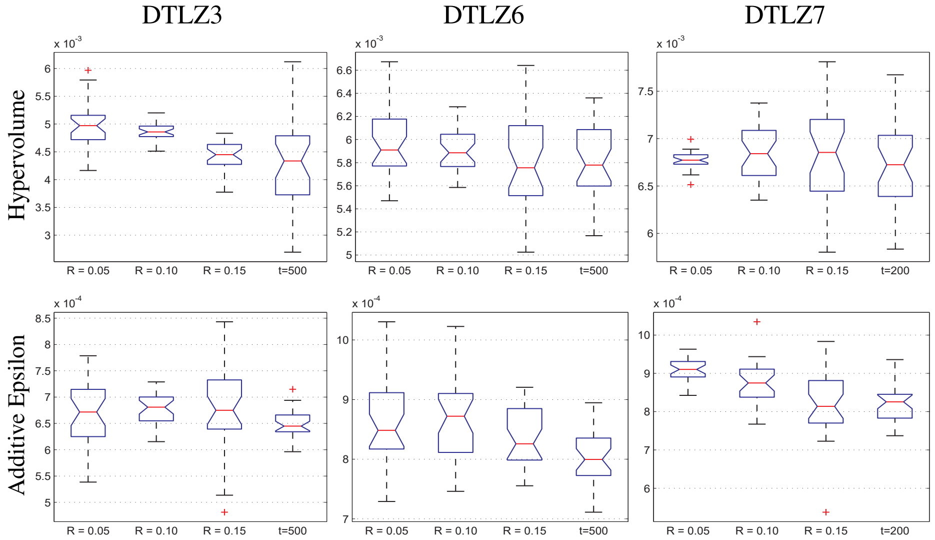


Figure 6: Box plots of the values of quality indicators comparing the Pareto–optimal front, $\mathcal{O}^*$, and the non–dominated solutions front, $\mathcal{PF}^*$, of the problems output when the MGBM criterion was met and in the final iteration reported by Deb et al. [56] and Khare [57].

OCD authors, that is, an iteration window of size 10, a variance threshold of -10^3 and a significance level of 0.05 for the statistical tests.

To make the study as comprehensive as possible, the components of MGBM and its alternatives were shuffled in all possible combinations. In other words, every progress indicator was tested with both evidence-gathering approaches. The results reported here were obtained after 30 independent runs of the algorithms solving each of the problems. The value of the MGBM parameter R was set to 0.1 and ε was 0.0001, as in the previous experiment. The hypothesis test was conducted from a sample consisting of 25 consecutive iterations with a confidence of 95%.

Figures **??** and **??** show the performance of the different criteria in the success-biased and failure-biased experiments, respectively. The points in time where each criterion suggested stopping are marked. These figures offer an quick glance of the outcome of each criterion, however, in order to gain a better understanding of the results a more detailed information regarding the stop iterations is presented in Figures **??** and **??** for the success- and failure-biased experiments respectively.

The first, success-biased, case prompts one key conclusion: evidence gathering via Kalman filters is able to detect the optimizer stagnation at earlier stages than the statistical hypothesis test. This is because Kalman filters only require measurement-wise decisions, although they do take into account previous measure-

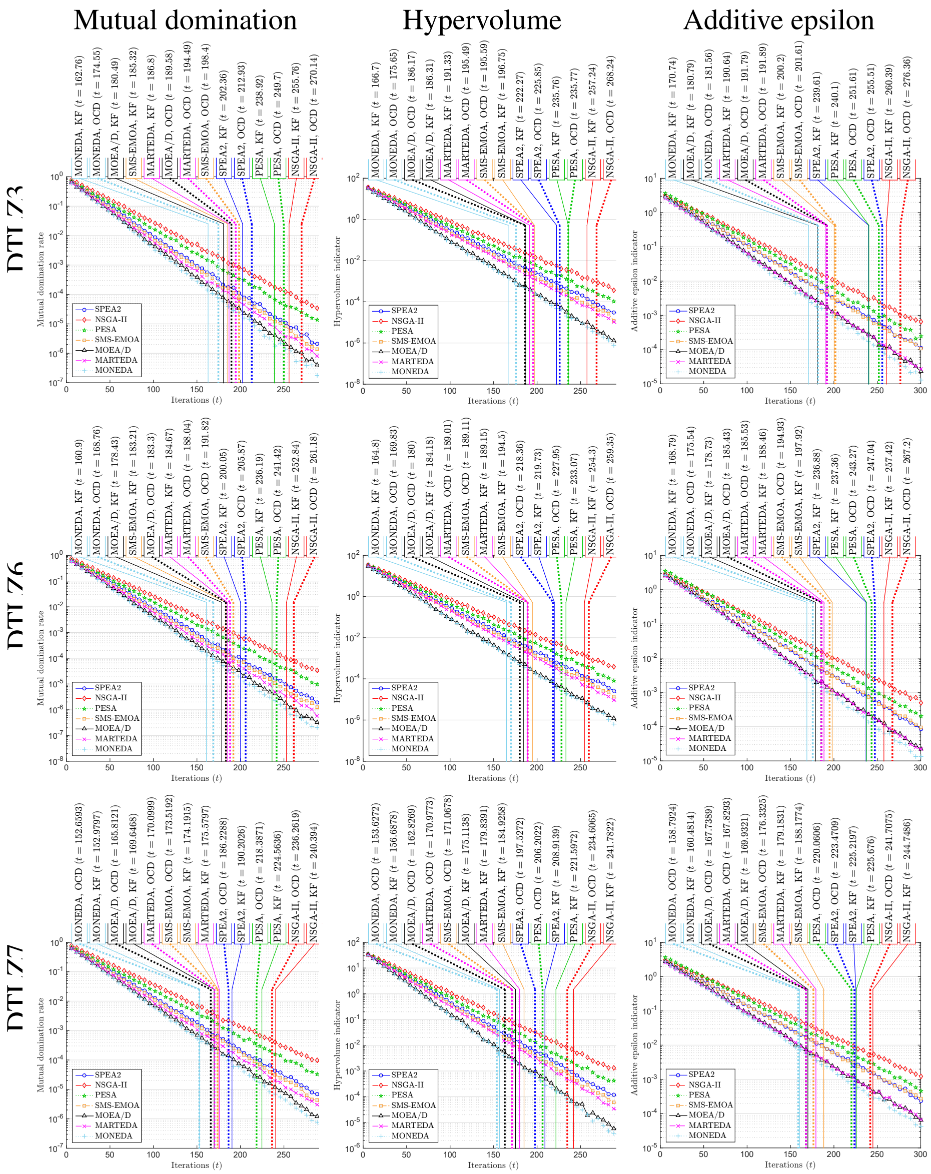


Figure 7: Success

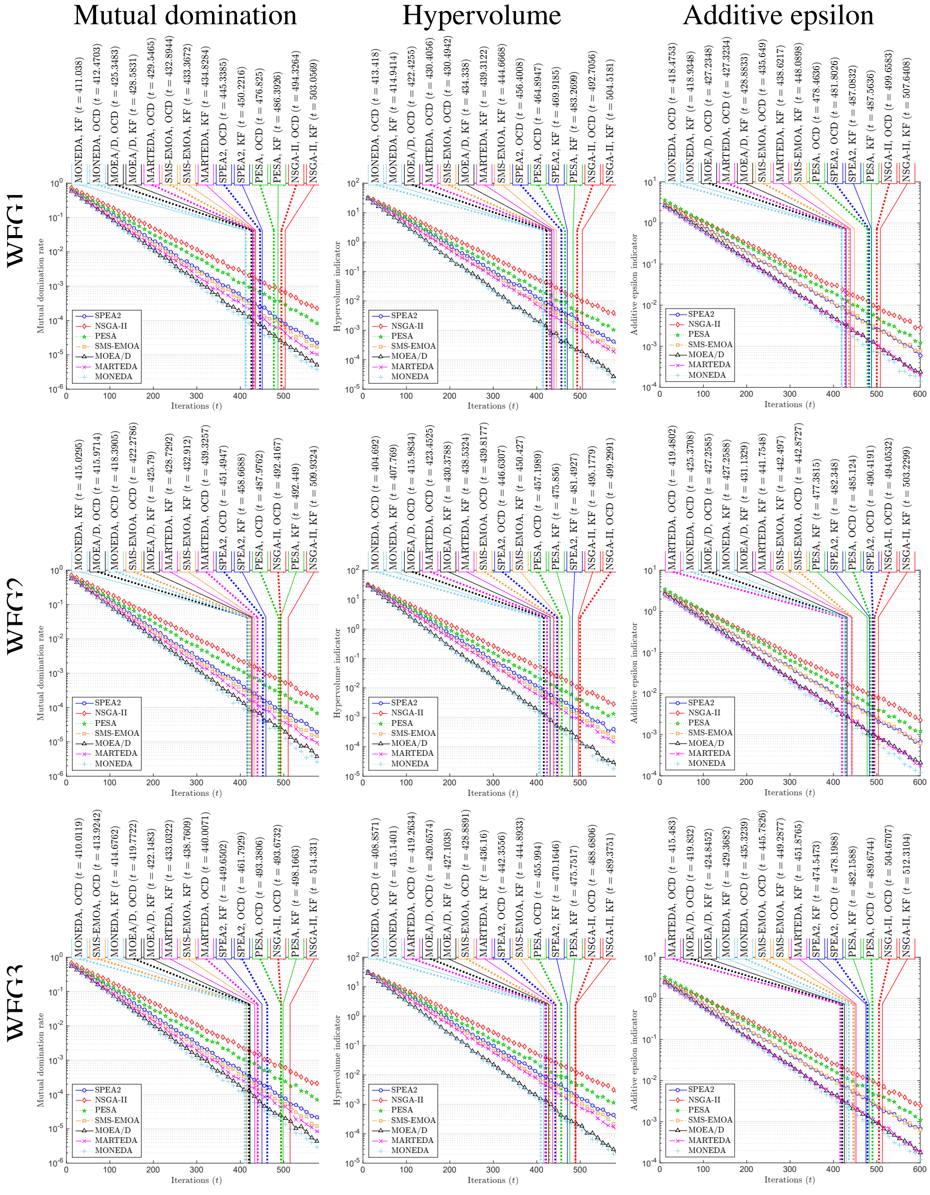


Figure 7: (Continued).

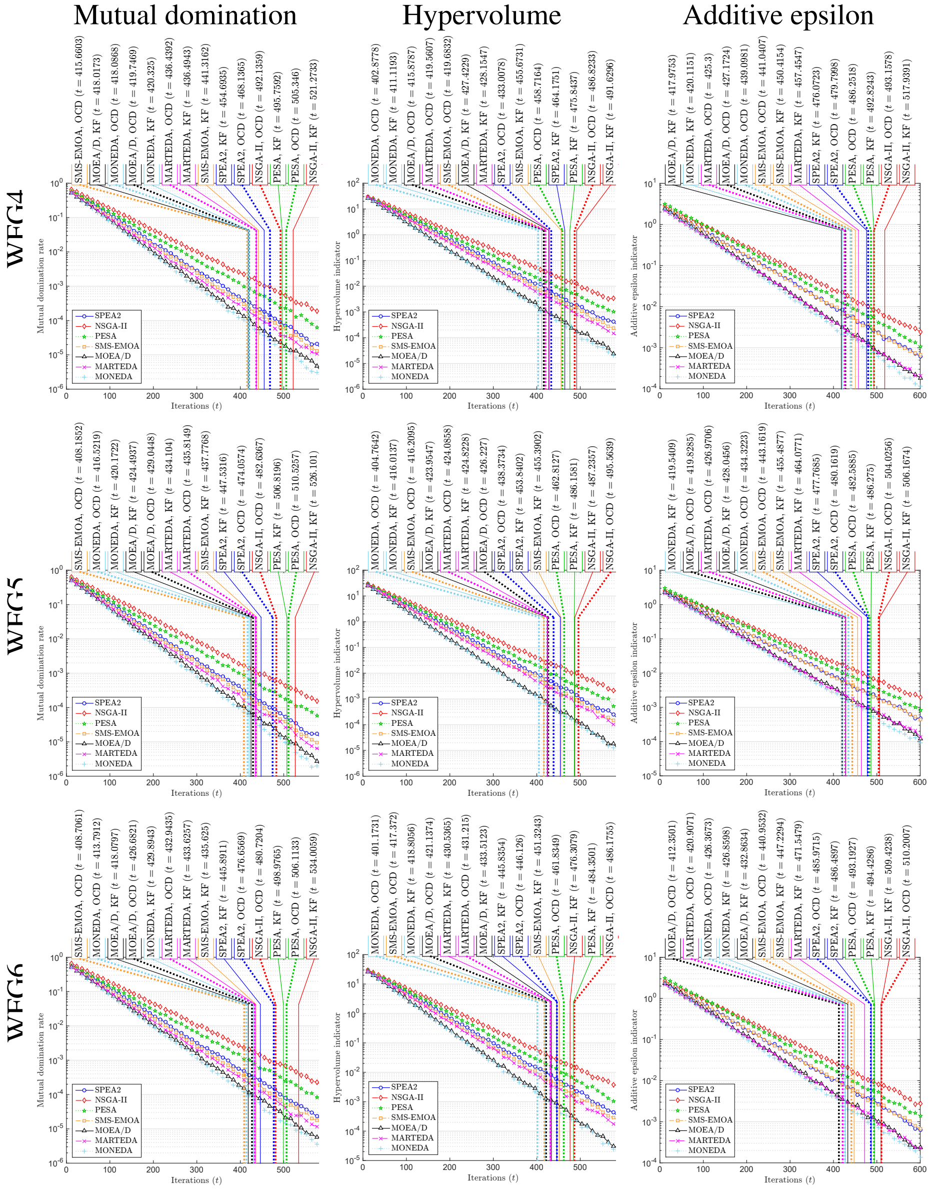


Figure 7: (Continued).

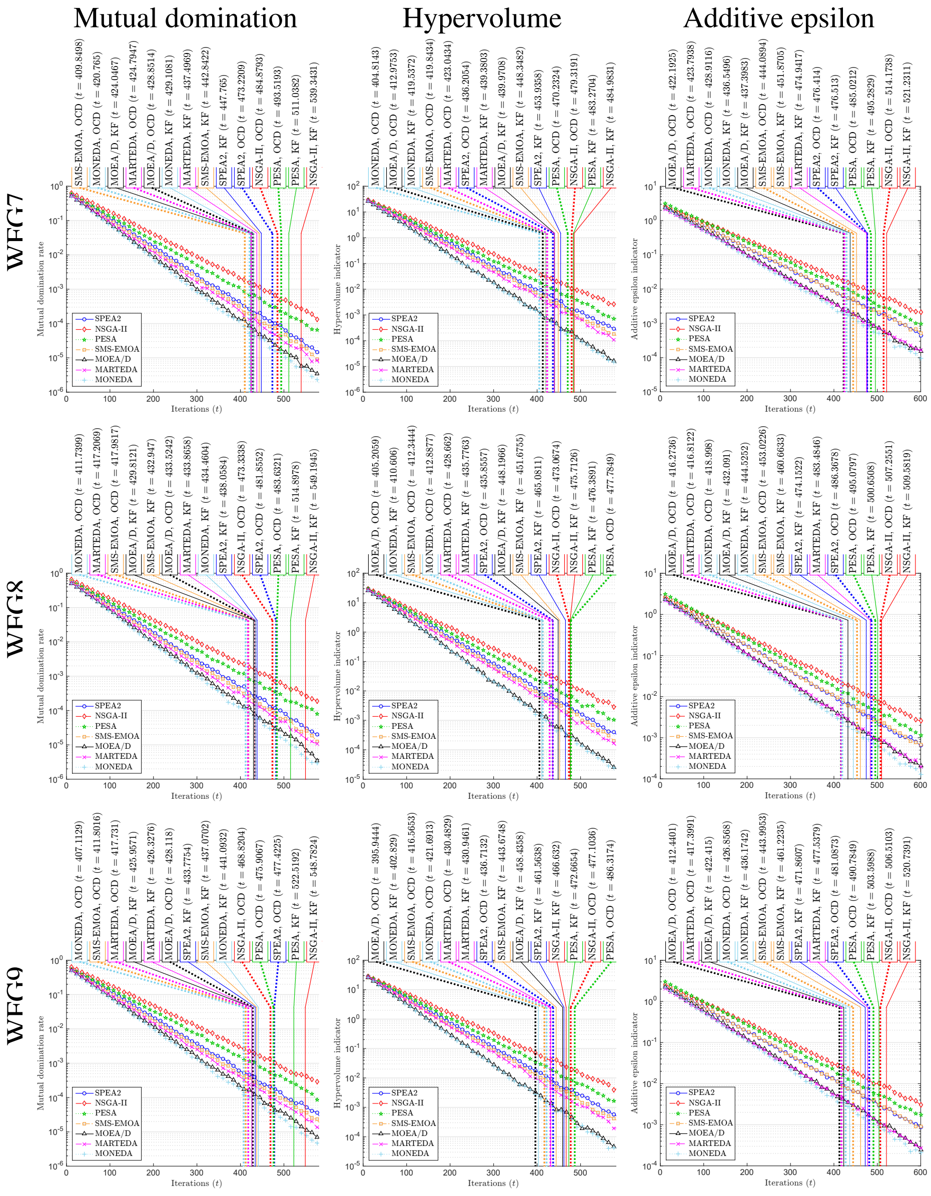


Figure 7: (Continued).

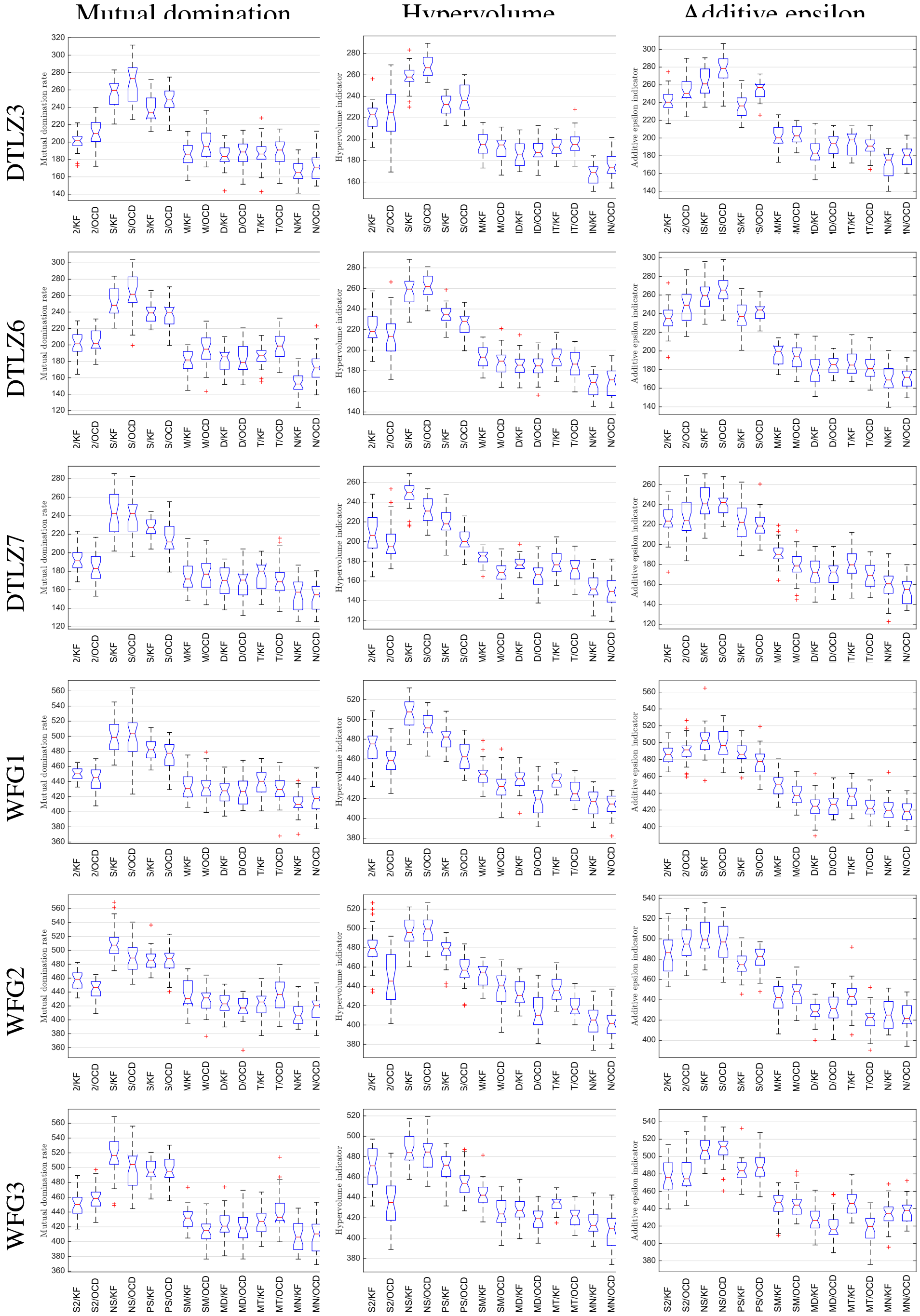


Figure 8: Success

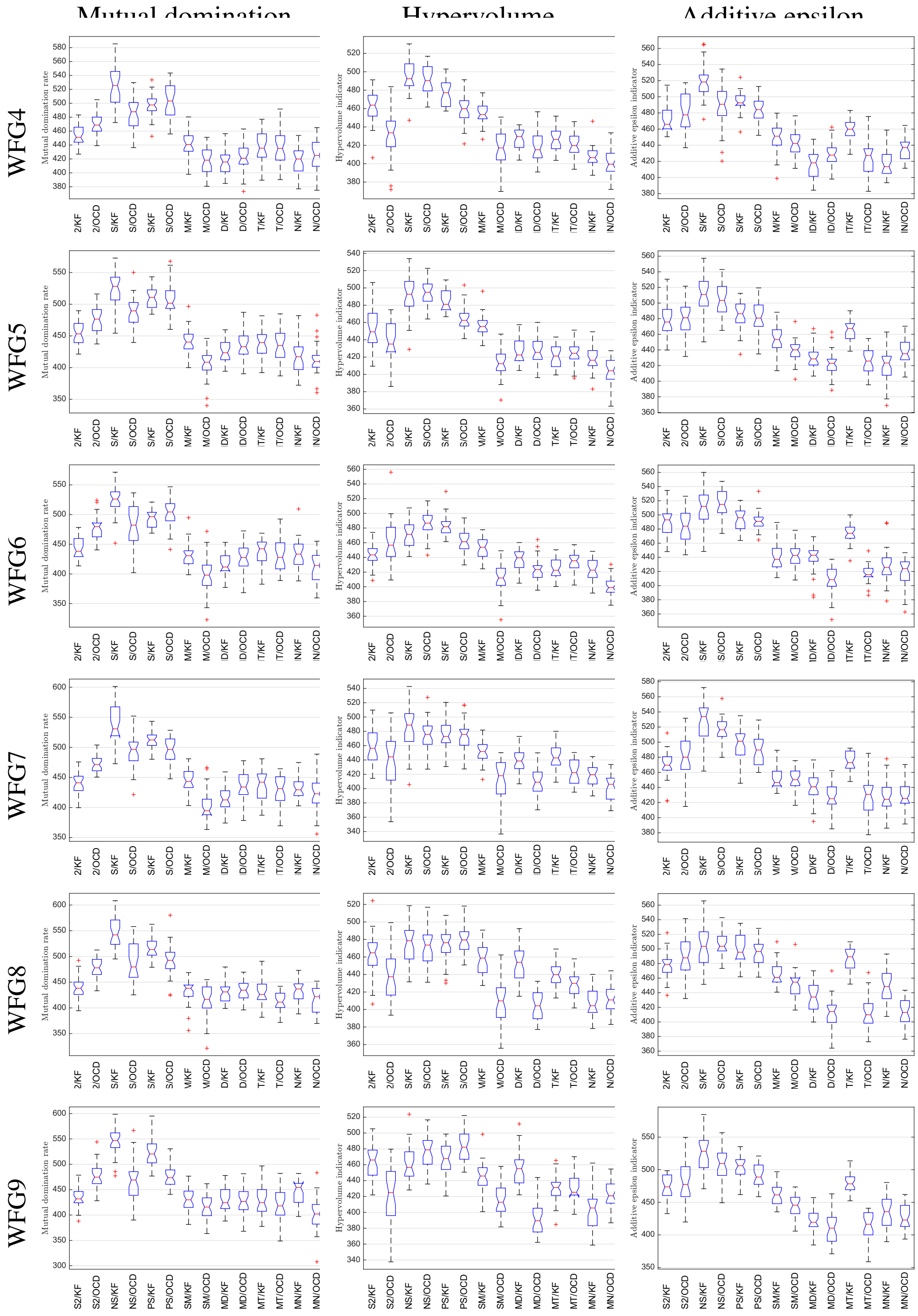


Figure 8: (Continued).

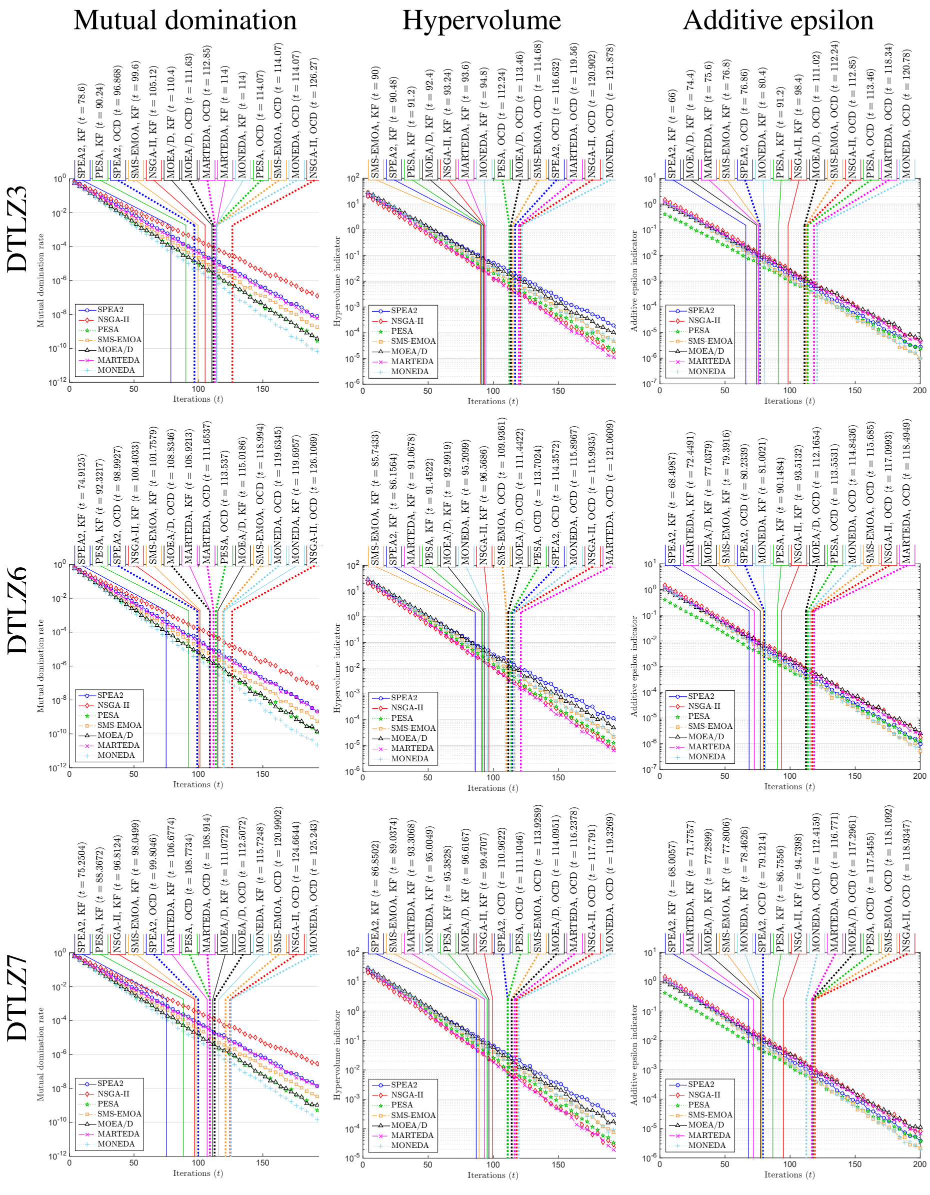


Figure 9: Failure

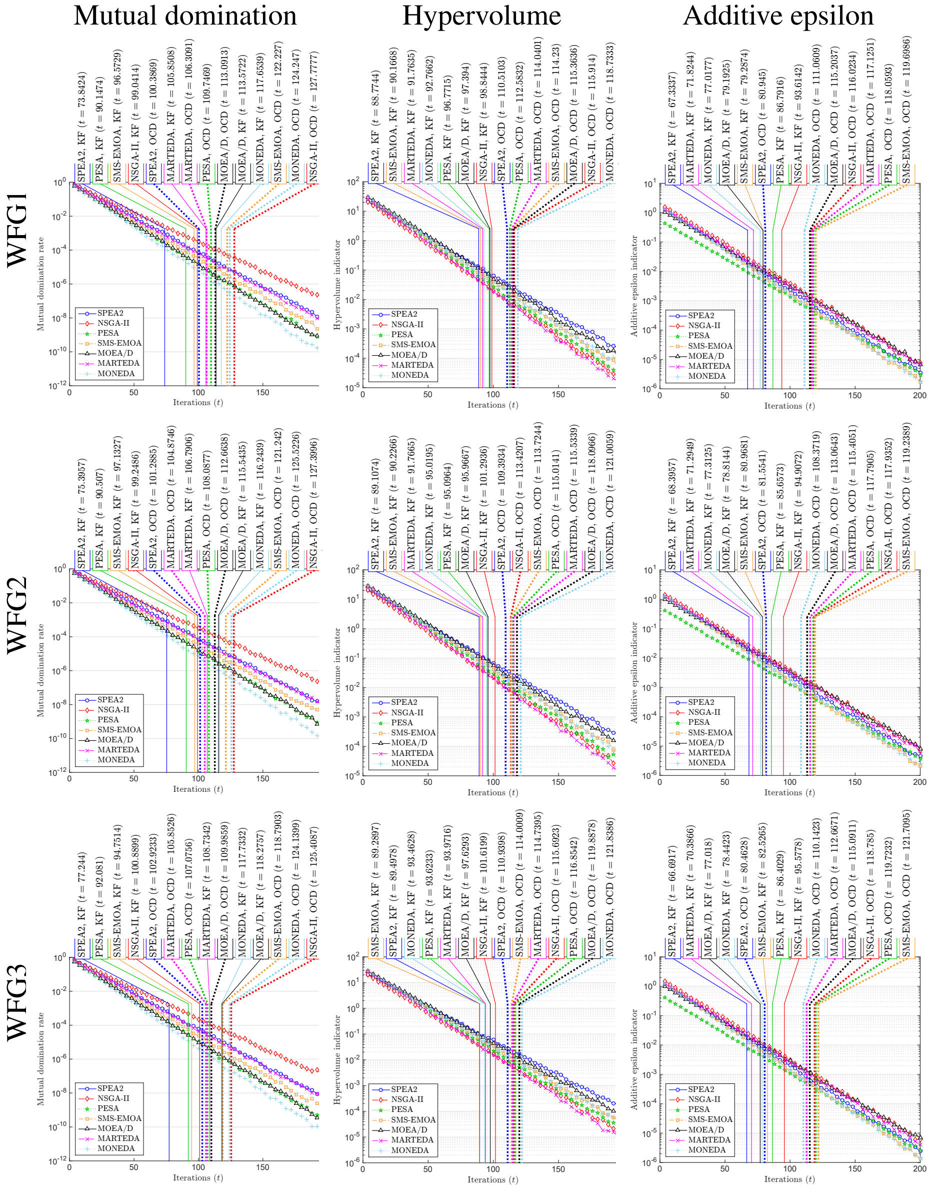


Figure 9: (Continued).

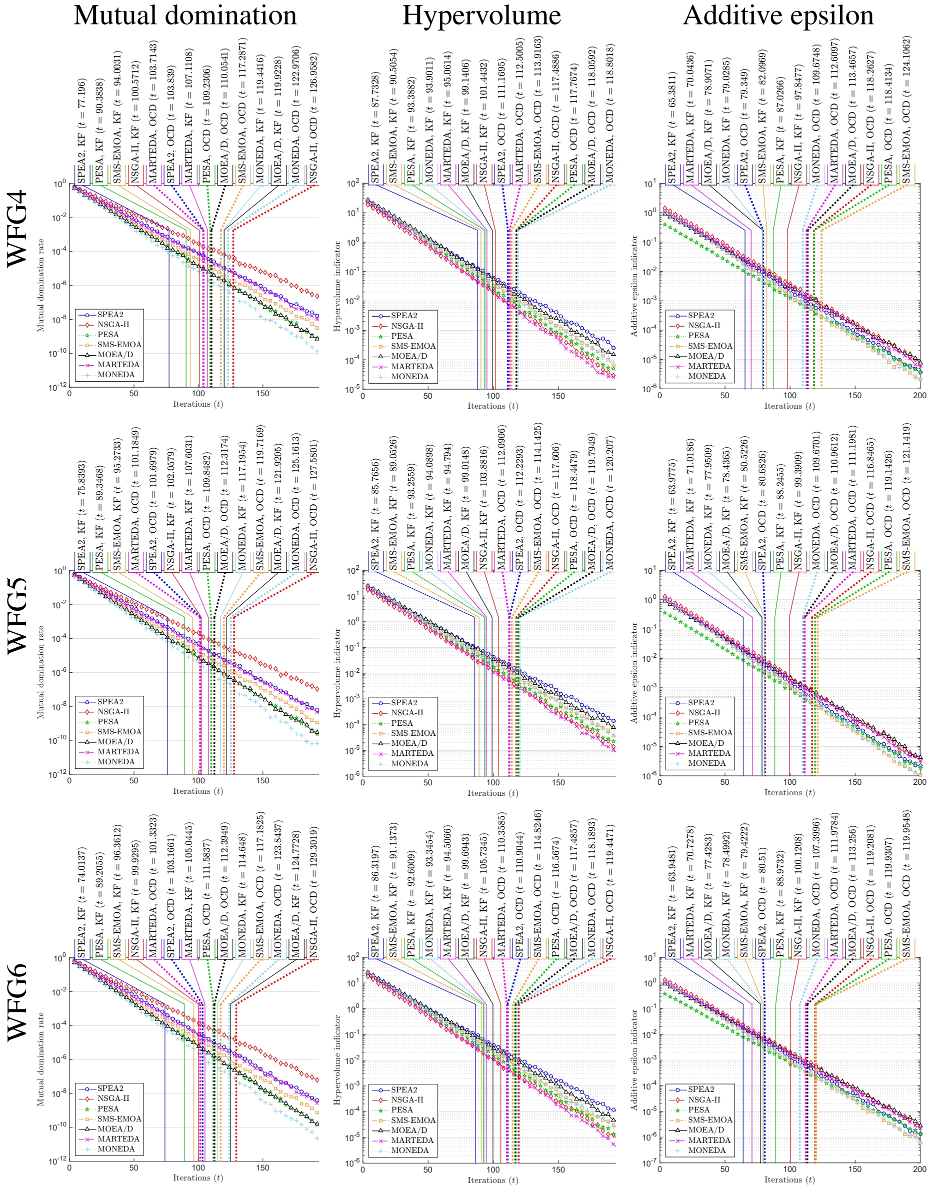


Figure 9: (Continued).

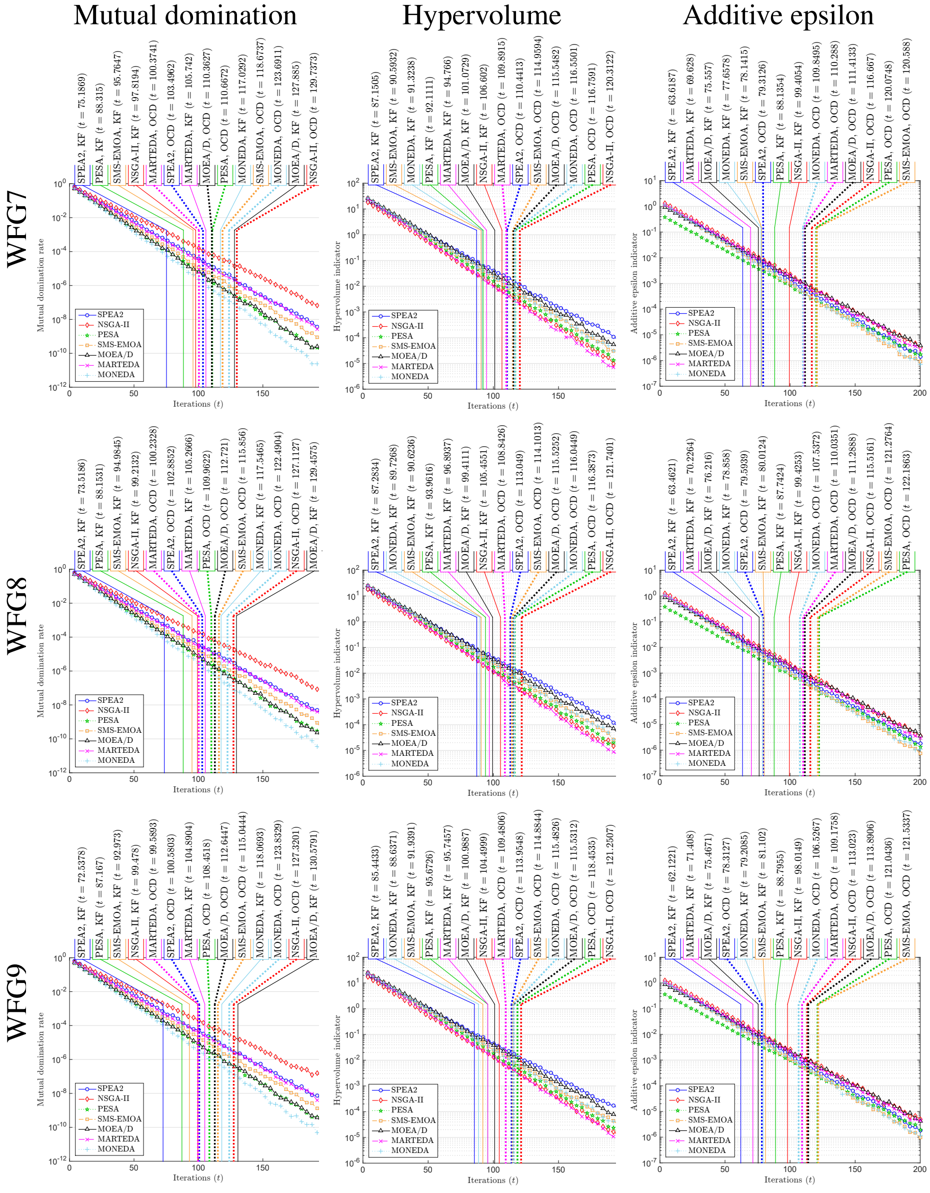


Figure 9: (Continued).

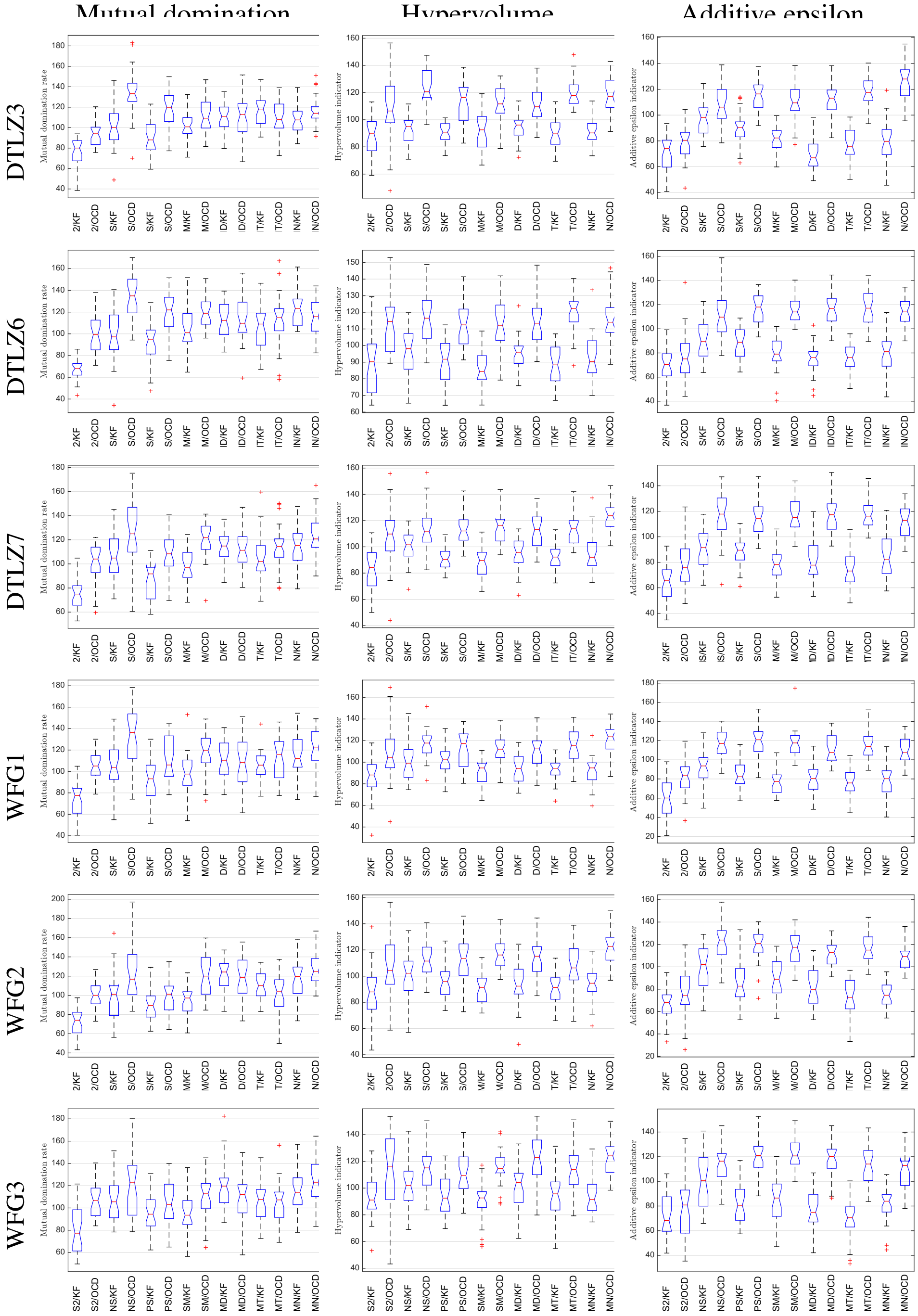


Figure 10: Failure

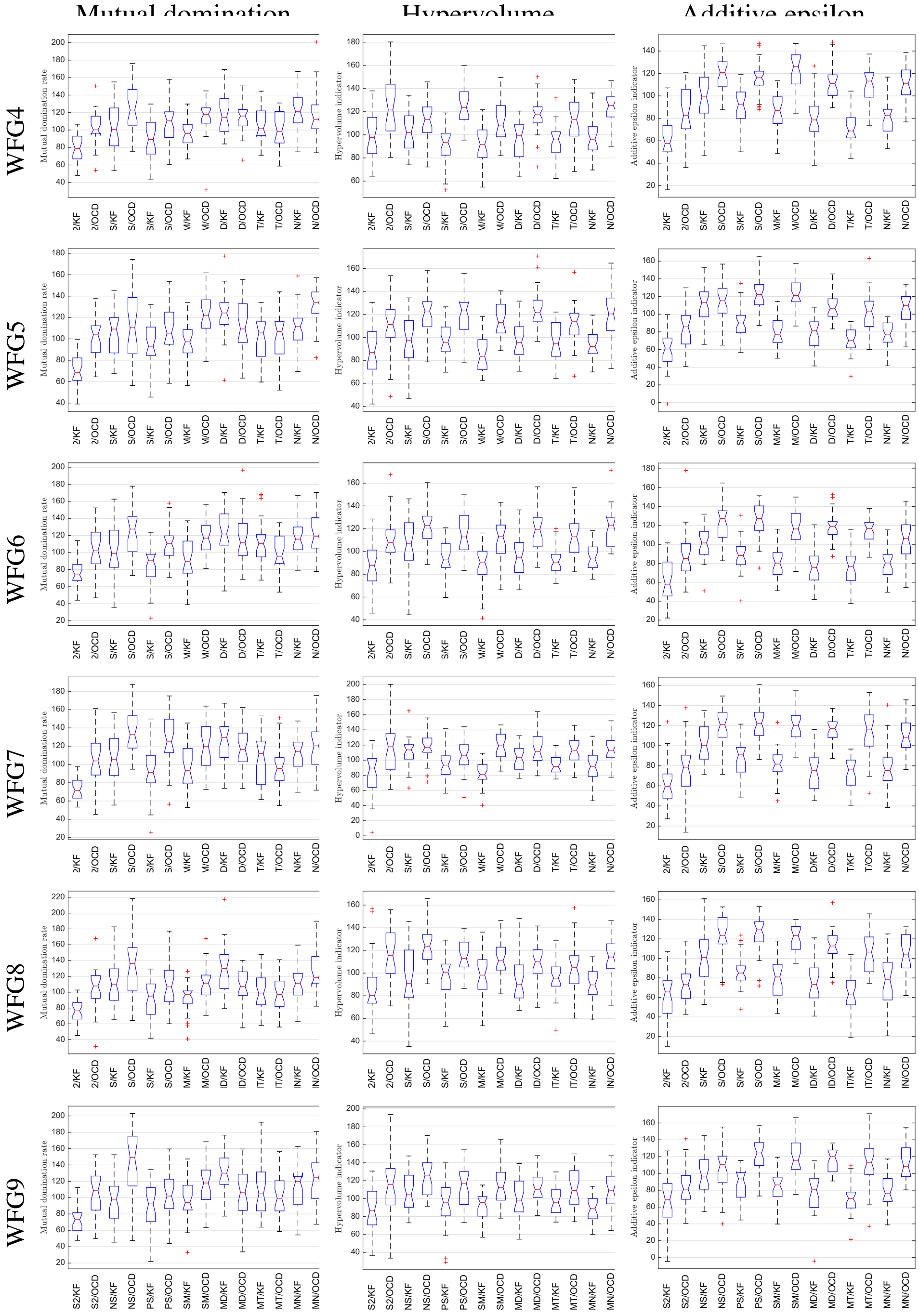


Figure 10: (Continued).

ments, while hypothesis tests must analyze a relatively large sample of measurements. This difference also implies that Kalman filters require less computation to produce their results. This difference could perhaps be narrowed by reformulating the hypothesis test in a recursive form.

On the other hand, regardless of the evidence tracker used, the application of the MDR indicator is able to signal the algorithm to stop executing before the other alternatives. This raises the question of whether MDR makes the criterion activate before the optimization process has actually stopped. In order to clarify this issue we measured the hypervolume and additive epsilon indicator values comparing the problem Pareto-optimal set $\mathcal{F}^*$ and the non-dominated set output when the corresponding criterion was met, $\mathcal{P}^*$. These results are summarized in Figure **??**. We can safely say here that the indicator values yielded by the criteria are similar and adequate.

Clearly if the MGBM criterion were able to signal a stop condition earlier than the other variants but the solutions output at that iteration were of a lesser quality, the criterion would be useless. Figure **??** shows that there is no substantial difference between the mean values of the quality indicators yielded by the different criteria. It can be inferred that, even though MGBM did fire earlier, it did not fire when the optimization was still in progress.

The early stopping indicated by MDR can be attributed to the fact that the other indicators, especially the hypervolume indicator, take into account the potential diversification in the non-dominated front of the algorithms. This is relatively less important when dealing with many-objective problems like the ones discussed above. The main concern in this class of application is to get as close as possible to the Pareto-optimal front, as the spread of solutions would place an even greater demand on computational resources.

The failure-biased experiments (Figure **??**) complement the results described above. The above rationale cannot be extrapolated as-is to the failure-biased experiments. Note that this class of experiment has not been previously proposed elsewhere, although we think it is indispensable to gain a complete experimental understanding of the matter. It is probably conceptually impossible to determine if the criteria actually managed to detect this situation. What we did find is that all the criteria did signal the algorithm to stop, and none left the algorithm running for an indefinite (and possibly infinite) number of iterations.

Still, there are some noteworthy points. It is again noticeable that the Kalman filter is able to detect the non-progress condition earlier. However, this difference is not as big as in the success-biased experiments. Similarly, the fact that all the evidence gathered performed more or less the same, regardless of the indicator used, is very illustrative. This is particularly noticeable in the case of the DTLZ7 problem.

The above results raise another question. What is the resource consumption of each of the combinations? This property is usually reported as the time taken by the process to terminate. However, this approach is hard to reproduce. For this reason, we will measure the number of CPU operations carried out to process the stopping criterion in each iteration instead. We found this approach to be sounder and more easily reproducible than just measuring the duration. To do this, we employed the OProfile software profiling tool [65]. The profiling tool was configured in such way that it only reported the CPU operations run by the processes of interest.

The mean number of CPU operations run by each evidence gatherer and progress indicator combination are summarized in Figures 11 and 12. We find that MGBM (the combination of MDR and Kalman filter) runs the fewest operations of all in most cases. In all cases, the application of the statistical hypothesis test is far more computationally expensive than its Kalman counterpart. This is due to the simplicity of the operations carried out by the Kalman filter compared to the repetitive assessment of indicator values of previous iterations used by the statistical hypothesis test. Also notable is the increment in the amount of computation of the hypervolume indicator, particularly evident in the failure-biased experiments. This can be attributed to its exponential relation to the number of objectives. Last but not least, let us look at the homogeneity of the results, a point that lends support to the possibility of outputting similar performances when dealing with other problems of the same magnitude.

Another interesting and important analysis regarding the computational impact of the criteria is to determine how much effort is dedicated to stopping with regard to the optimization process itself. Figure 13 shows this comparison. It is visually clear that when the hypervolume is involved the computational requirements of the criterion are increased. Any how, these results should not be taken literally. In this study the stopping criteria are an independent module. However, in practical applications the computation of the progress indicators, MDR in particular, can be easily embedded inside of the MOEA processes, thus reducing the computational costs.

6. Final remarks

In this paper we have presented a novel stopping criterion to be used in multi-objective optimization problems. In particular, we proposed a global stopping criterion, named MGBM criterion, which combines the mutual domination rate (MDR) improvement indicator with a simplified Kalman filter that is used in the evidence-gathering process. The MDR indicator is a special-purpose solution designed for the stopping task. It is capable of gauging the progress of the optimization with a low computational cost and is therefore suitable for solving complex or many-objective problems. Although the stopping criterion issue apparently plays a secondary role, real-world practical experiences underscore its importance.

As part of this paper we have described the criterion theoretically and have examined its performance on some test problems. It was also compared with similar approaches to the issue. From these experiments we have found that MGBM is a good starting point for research in this direction. Obviously, more experimentation is required, and other types of filters must be tested. Research on creating the necessary assessment tools to be able to gauge the performance of the criteria would perhaps not go amiss either. It should be noted, however, that the criterion has been successfully applied by the authors in a series of studies that deal with high-dimensionality multi-objective problems [66–69].

A salient issue is the interpretation of the final algorithm state in order to establish the reason for the process being stopped. Evidence gathered during this research indicates that some conclusions can be drawn on this point by analyzing the number of dominated and non-dominated individuals in the population. These results are consistent with outcomes previously presented by Khare et al. [58], Purshouse and Fleming [59], Knowles and Corne [60], Ishibuchi et al. [61], Praditwong and Yao [70] and Deb [6, pp.414–419].

Another key issue is to capture the diversification process that takes place as part of the optimization process. It has been documented that after the population hits a local Pareto front it starts exploring along that front. An indicator capable of measuring the degree to which the optimization algorithm is actively exploring the search space could perhaps improve the results presented here.

Acknowledgements

This work was partially funded by CNPq BJT Project 407851/2012-7 and CNPq PVE Project 314017/2013-5.

References

[1] T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), Handbook of Evolutionary Computation, Institute of Physics Publishing and Oxford University Press, 1997.

[2] M. Safe, J. Carballido, I. Ponzoni, N. Brignole, On stopping criteria for genetic algorithms, in: Advances in Artificial Intelligence – SBIA 2004, Springer, Berlin/Heidelberg, 2004, pp. 405–413.

[3] G. Hernandez, K. Wilder, F. Nino, J. Garcia, Towards a self-stopping evolutionary algorithm using coupling from the past, in: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM, New York, NY, USA, 2005, pp. 615–620.

[4] K. Zielinski, R. Laur, Stopping criteria for a constrained single–objective particle swarm optimization algorithm, Informatica 31 (2007) 51–59.

[5] K. Zielinski, D. Peters, R. Laur, Stopping criteria for single–objective optimization, in: Third International Conference on Computational Intelligence, Robotics and Autonomous Systems.

[6] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[7] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Genetic and Evolutionary Computation, Springer, New York, second edition, 2007.

[8] K. Miettinen, Nonlinear Multiobjective Optimization, volume 12 of *International Series in Operations Research & Management Science*, Kluwer, Norwell, MA, 1999.

[9] M. Ehrgott, Multicriteria Optimization, volume 491 of *Lecture Notes in Economics and Mathematical Systems*, Springer, 2005.

[10] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, V. Grunert da Fonseca, Why Quality Assessment of Multiobjective Optimizers Is Difficult, in: W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. Potter, A. Schultz, J. Miller, E. Burke, N. Jonoska (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), Morgan Kaufmann Publishers, San Francisco, California, 2002, pp. 666–673.

[11] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. Grunert da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2003) 117–132.

[12] J. Knowles, L. Thiele, E. Zitzler, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006. Revised version.

[13] G. Rudolph, A. Agapie, Convergence Properties of Some Multi-Objective Evolutionary Algorithms, in: Proceedings of the 2000 Conference on Evolutionary Computation, volume 2, IEEE Press, Piscataway, New Jersey, 2000, pp. 1010–1016.

[14] T. Hanne, On the convergence of multiobjective evolutionary algorithms, European Journal of Operational Research 117 (1999) 553–564.

[15] C. A. Coello Coello, An Updated Survey of GA-Based Multiobjective Optimization Techniques, ACM Computing Surveys 32 (2000) 109–143.

[16] C. A. Coello Coello, Recent trends in evolutionary multiobjective optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Advanced Information and Knowledge Processing, Springer–Verlag, 2004, pp. 7–32.

[17] O. Rudenko, M. Schoenauer, A steady performance stopping criterion for pareto–based evolutionary algorithms, in: The 6th International Multi–Objective Programming and Goal Programming Conference, Hammamet, Tunisia.

[18] L. Martí, J. García, A. Berlanga, J. M. Molina, A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms, in: D. Thierens, K. Deb, M. Pelikan, H.-G. Beyer, B. Doerr, R. Poli, M. Bittari (Eds.), GECCO '09: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM Press, New York, 2007, p. 911.

[19] H. Trautmann, U. Ligges, J. Mehnen, M. Preuss, A convergence criterion for multiobjective evolutionary algorithms based on systematic statistical testing, in: Parallel Problem Solving from Nature — PPSN X, Springer, Heidelberg, 2008, pp. 825–836.

[20] T. Wagner, H. Trautmann, B. Naujoks, OCD: Online convergence detection for evolutionary multi–objective algorithms based on statistical testing, in: M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, M. Sevaux (Eds.), 5th International Conference on Evolutionary Multi–Criterion Optimization (EMO 2009), volume 5467 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 198–215.

[21] L. T. Bui, S. Wesolkowski, A. Bender, H. Abbass, M. Barlow, A dominance–based stability measure for multi–objective evolutionary algorithms, in: 2009 IEEE Congress on Evolutionary Computation (CEC '09), IEEE Press, Piscataway, New Jersey, 2009, pp. 749–756.

[22] L. Martí, J. García, A. Berlanga, J. M. Molina, An approach to stopping criteria for multi–objective optimization evolutionary algorithms: The MGBM criterion, in: 2009 IEEE Conference on Evolutionary Computation (CEC 2009), IEEE Press, Piscataway, New Jersey, 2009, pp. 1263–1270.

[23] B. Naujoks, H. Trautmann, Online convergence detection for multiobjective aerodynamic applications, in: 2009 IEEE Congress on Evolutionary Computation (CEC '09), IEEE Press, Piscataway, New Jersey, 2009, pp. 332–339.

[24] J. L. Guerrero, J. García, L. Martí, J. M. Molina, A. Berlanga, A stopping criterion based on Kalman estimation techniques with several progress indicators, in: G. Raidl, E. Alba, J. Bacardit, C. Bates Congdon, H.-G. Beyer, M. Birattari, C. Blum, P. A. N. Bosman, D. Corne, C. Cotta, M. Di Penta, B. Doerr, R. Drechsler, M. Ebner, J. Grahl, T. Jansen, J. Knowles, T. Lenaerts, M. Middendorf, J. F. Miller, M. O'Neill, R. Poli, G. Squillero, K. Stanley, T. Stützle, J. van Hemert (Eds.), GECCO 2009: Proceedings 11th Annual Conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2009, pp. 587–594.

[25] H. Trautmann, B. Naujoks, Online convergence detection in evolutionary multiobjective algorithm, in: Optimizing Drug Design Workshop, Leiden, Netherlands.

[26] H. Trautmann, T. Wagner, B. Naujoks, M. Preuss, J. Mehnen, Statistical methods for convergence detection of multi-objective evolutionary algorithms, Evolutionary Computation 17 (2009) 493–509. PMID: 19916777.

[27] R. E. Kalman, A new approach to linear filtering and prediction problems, Transaction of the ASME — Journal of Basic Engineering 82 (1960) 35–45.

[28] R. C. Purshouse, On the Evolutionary Optimisation of Many Objectives, Ph.D. thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK, 2003.

[29] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[30] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, in: K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, T. Fogarty (Eds.), EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100.

[31] D. W. Corne, J. D. Knowles, M. J. Oates, The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, H.-P. Schwefel (Eds.), Proceedings of the Parallel Problem Solving from Nature VI Conference, Springer. Lecture Notes in Computer Science No. 1917, Paris, France, 2000, pp. 839–848.

[32] V. Pareto, Cours D'Économie Politique, F. Rouge, Lausanne, 1896.

[33] E. Zitzler, L. Thiele, An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach, Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1998.

[34] N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation 2 (1994) 221–248.

[35] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization, in: L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, E. Burke (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), Morgan Kaufmann Publishers, San Francisco, California, 2001, pp. 283–290.

[36] J. D. Knowles, D. W. Corne, The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation, in: 1999 Congress on Evolutionary Computation, IEEE Service Center, Washington, D.C., 1999, pp. 98–105.

[37] J. D. Knowles, D. W. Corne, Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy, Evolutionary Computation 8 (2000) 149–172.

[38] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty, in: A. S. Wu (Ed.), Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program, Orlando, Florida, pp. 121–122.

[39] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: S. Obayashi, et al. (Eds.), Conference on Evolutionary Multi–Criterion Optimization (EMO 2007), volume 4403 of *LNCS*, Springer, Berlin, 2007, pp. 862–876.

[40] J. D. Knowles, Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization, Ph.D. thesis, The University of Reading, Department of Computer Science, Reading, UK, 2002.

[41] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[42] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE Transactions on Evolutionary Computation 10 (2006) 29–38.

[43] L. While, L. Bradstreet, L. Barone, P. Hingston, Heuristics for Optimising the Calculation of Hypervolume for Multi-Objective Optimization Problems, in: 2005 IEEE Congress on Evolutionary Computation (CEC'2005), volume 3, IEEE Service Center, Edinburgh, Scotland, 2005, pp. 2225–2232.

[44] C. M. Fonseca, L. Paquete, M. López-Ibánez, An improved dimension–sweep algorithm for the hypervolume indicator, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), pp. 1157–1163.

[45] N. Beume, G. Rudolph, Faster S–metric calculation by considering dominated hypervolume as Klee's measure problem, in: B. Kovalerchuk (Ed.), Proceedings of the Second IASTED International Conference on Computational Intelligence, IASTED/ACTA Press, 2006, pp. 233–238.

[46] N. Beume, S–metric calculation by considering dominated hypervolume as Klee's measure problem, Evolutionary Computation 17 (2009) 477–492. PMID: 19916778.

[47] K. Srinagesh, The Principles of Experimental Research, Butterworth–Heinemann (Elsevier), Oxford, UK, 2006.

[48] K. S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, John Wiley and Sons, Chichester, UK, second edition, 2002.

[49] H. Aytug, G. J. Koehler, Stopping criteria for finite length genetic algorithms, INFORMS: Journal on Computing 8 (1996) 183–191.

[50] H. Aytug, G. J. Koehler, New stopping criterion for genetic algorithms, European Journal of Operational Research 126 (2000) 662–674.

[51] J. A. Besada, J. Garcia, G. de Miguel, A. Berlanga, J. M. Molina, J. R. Casar, Design of IMM filter for radar tracking using evolution strategies, IEEE Transactions on Aerospace and Electronic Systems 41 (2005) 1109–1122.

[52] H. A. P. Blom, Y. Bar-Shalom, The interacting multiple model algorithm for systems with Markovian switching coefficients, IEEE Transactions on Automatic Control 33 (1988) 780–783.

[53] D. A. V. Veldhuizen, G. B. Lamont, Evolutionary Computation and Convergence to a Pareto Front, in: J. R. Koza (Ed.), Late Breaking Papers at the Genetic Programming 1998 Conference, Stanford University Bookstore, Stanford University, California, 1998, pp. 221–228.

[54] K. Deb, S. Jain, Running Performance Metrics for Evolutionary Multi-Objective Optimization, in: L. Wang, K. C. Tan, T. Furuhashi, J.-H. Kim, X. Yao (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02), volume 1, Nanyang Technical University, Orchid Country Club, Singapore, 2002, pp. 13–20.

[55] P. S. Maybeck, Stochastic models, estimation, and control, volume 141 of *Mathematics in Science and Engineering*, Academic Press, 1979.

[56] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Advanced Information and Knowledge Processing, Springer Verlag, 2004, pp. 105–145.

[57] V. Khare, Performance Scaling of Multi-Objective Evolutionary Algorithms, Master's thesis, School of Computer Science, The University of Birmingham, Edgbaston, Birmingan, UK, 2002.

[58] V. Khare, X. Yao, K. Deb, Performance Scaling of Multi-objective Evolutionary Algorithms, in: C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, L. Thiele (Eds.), Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, 2003, pp. 376–390.

[59] R. C. Purshouse, P. J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Transactions on Evolutionary Computation 11 (2007) 770–784.

[60] J. Knowles, D. Corne, Quantifying the effects of objective space dimension in evolutionary multiobjective optimization, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Proceedings 4th International Conference Evolutionary Multi–Criterion Optimization (EMO 2007), Springer, Berlin/Heidelberg, 2007, pp. 757–771.

[61] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many–objective optimization: A short review, in: IEEE Congress on Evolutionary Computation, 2008 (CEC 2008) part of IEEE World Congress on Computational Intelligence, IEEE Press, Piscataway, New Jersey, 2008, pp. 2419–2426.

[62] S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler, PISA—A Platform and Programming Language Independent Interface for Search Algorithms, in: C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, L. Thiele (Eds.), Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, 2003, pp. 494–508.

[63] H. Chernoff, L. E. L., The use of maximum likelihood estimates in χ^2 tests for goodness-of-fit, The Annals of Mathematical Statistics 25 (1954) 579–586.

[64] W. S. Gosset a.k.a. Student, The probable error of a mean, Biometrika 6 (1908) 1–25.

[65] J. Levon, OProfile manual, Victoria University of Manchester, 2004.

[66] L. Martí, J. García, A. Berlanga, J. M. Molina, Scalable continuous multiobjective optimization with a neural network–based estimation of distribution algorithm, in: M. Giacobini, A. Brabazon, S. Cagnoni, G. A. Di Caro, R. Drechsler, A. Ekárt, A. I. Esparcia-Alcázar, M. Farooq, A. Fink, J. McCormack, M. O'Neill, J. Romero, F. Rothlauf, G. Squillero, A. c. Uyar, S. Yang (Eds.), Applications of Evolutionary Computing, volume 4974 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2008, pp. 535–544.

[67] L. Martí, J. García, A. Berlanga, J. M. Molina, Model–building algorithms for multiobjective EDAs: Directions for improvement, in: Z. Michalewicz (Ed.), 2008 IEEE Conference on Evolutionary Computation (CEC), part of 2008 IEEE World Congress on Computational Intelligence (WCCI 2008), IEEE Press, Piscataway, New Jersey, 2008, pp. 2848–2855.

[68] L. Martí, J. García, A. Berlanga, J. M. Molina, Introducing MONEDA: Scalable multiobjective optimization with a neural estimation of distribution algorithm, in: M. Keizer, G. Antoniol, C. Congdon, K. Deb, B. Doerr, N. Hansen, J. Holmes, G. Hornby, D. Howard, J. Kennedy, S. Kumar, F. Lobo, J. Miller, J. Moore, F. Neumann, M. Pelikan, J. Pollack, K. Sastry, K. Stanley, A. Stoica, E. G. Talbi, I. Wegener (Eds.), GECCO '08: 10th Annual Conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2008, pp. 689–696. EMO Track "Best Paper" Nominee.

[69] L. Martí, J. García, A. Berlanga, J. M. Molina, On the computational properties of the multi–objective neural estimation of distribution algorithms, in: N. Krasnogor, B. Melián-Batista, J. A. Moreno-Pérez, J. M. Moreno-Vega, D. Pelta (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2008), volume 236 of *Studies in Computational Intelligence*, Springer, Berlin/Heidelberg, 2009, pp. 239–251.

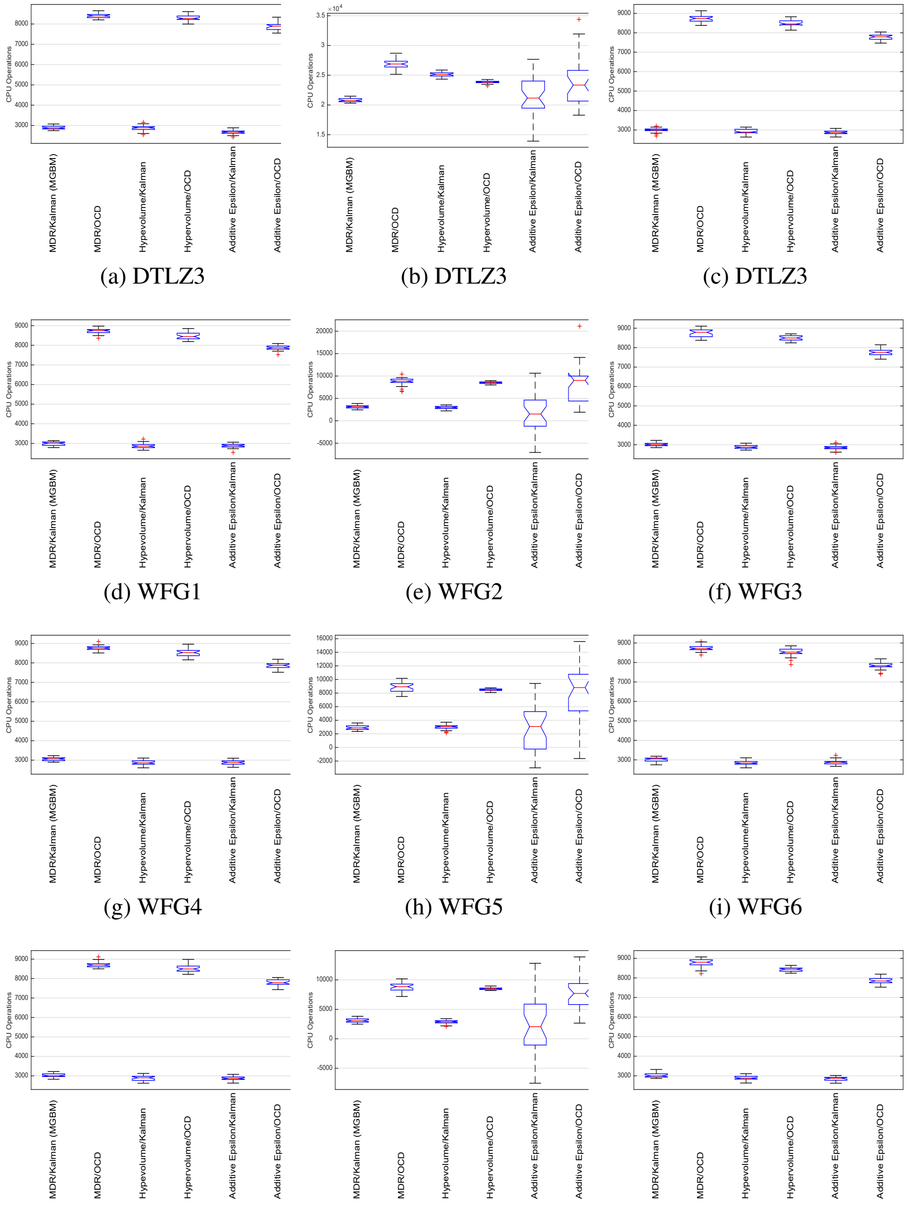
[70] K. Praditwong, X. Yao, How well do multi–objective evolutionary algorithms scale to large problems, in: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), IEEE Press, Piscataway, New Jersey, 2007, pp. 3959–3966.

Table 1: Parameters of the algorithms used in the experiments.

Common parameters (where applicable)	
Population size	100
Crossover probability	0.7
Dist. index for simul. binary crossover	15
Mutation probability	0.05
Dist. index for polynomial mutation	20
SPEA2 and PESA	
Ratio of sizes of population and archive	4:1
MARTEDA	
F2 vigilance threshold	0.05
Initial standard deviations	**0.01**
Selection percentile	0.3
$\hat{P}_t$ to N^* ratio	0.5
Substitution percentile	0.25
MONEDA	
Number of initial GNG nodes	2
Maximum edge age	40
Best node learning rate	0.1
Neighboring nodes learning rate	0.05
Insertion error decrement rate	0.1
General error decrement rate	0.1
Accumulated error threshold	0.2
Selection percentile	0.3
$\hat{\mathcal{P}}_t$ to $N_{\max}$ ratio	0.5
Substitution percentile	0.25

Table 2: Stop iterations suggested by the MGBM criterion with different values of R and the number of iterations used in [56] and [57] when solving DTLZ3, DTLZ6 and DTLZ7 with similar configurations of NSGA-II, SPEA2 and PESA.

MOEA	**MGBM**			**Deb et al. [56]**	**Khare [57]**
	$R = 0.05$	$R = 0.1$	$R = 0.15$		
			DTLZ3		
NSGA-II	91	104	115	500	500
SPEA2	121	132	149	500	500
PESA	125	129	138	500	500
			DTLZ6		
NSGA-II	104	106	140	500	500
SPEA2	71	78	123	500	500
PESA	95	103	151	500	500
			DTLZ7		
NSGA-II	237	259	275	200	N/A
SPEA2	269	305	330	200	N/A
PESA	279	298	326	200	N/A



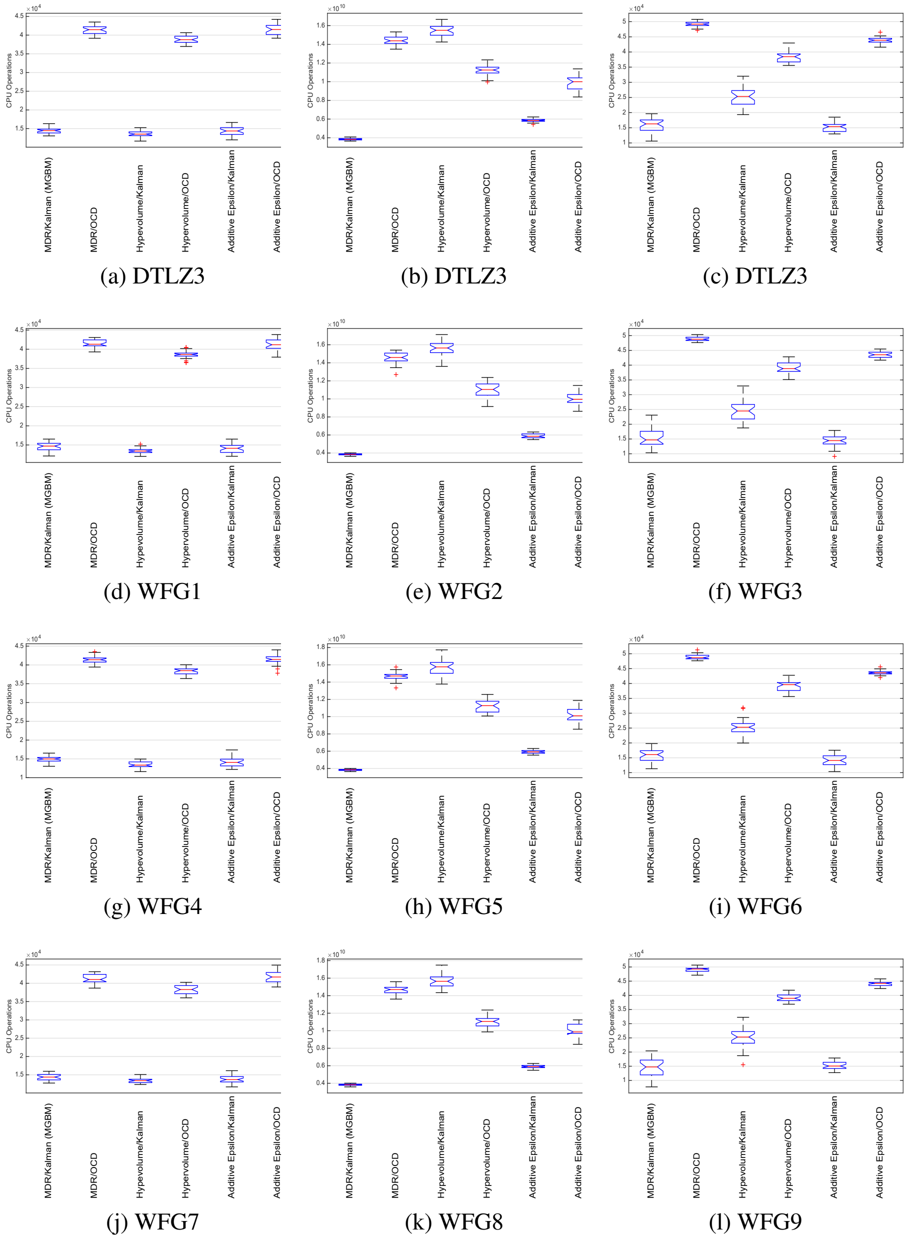
(a) DTLZ3 (b) DTLZ3 (c) DTLZ3

(d) WFG1 (e) WFG2 (f) WFG3

(g) WFG4 (h) WFG5 (i) WFG6

(j) WFG7 (k) WFG8 (l) WFG9

Figure 11: Mean CPU operations per iteration performed by each stopping criterion for success-biased experiments.



(a) DTLZ3

(b) DTLZ3

(c) DTLZ3

(d) WFG1

(e) WFG2

(f) WFG3

(g) WFG4

(h) WFG5

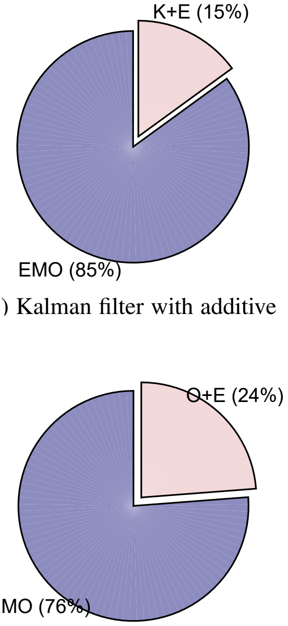
(i) WFG6

(j) WFG7

(k) WFG8

(l) WFG9

Figure 12: Mean CPU operations per iteration performed by each stopping criterion for failure-biased experiments.



(c) Kalman filter with additive

(f) OCD with additive epsilon (O+E)

Figure 13: Relationship between the amount of CPU resources dedicated to stopping criterion and the MOEA algorithms that hosts it.