

This document is published in:

IEEE Congress on Evolutionary Computation (CEC 2010). Barcelona, Spain, 18-23 July 2010.
IEEE, pp. 4314-4321. DOI: <http://dx.doi.org/10.1109/CEC.2010.5586265>

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Introducing a Robust and Efficient Stopping Criterion for MOEAs

José L. Guerrero, Luis Martí, Student Member IEEE, Antonio Berlanga, Member IEEE, Jesús García, Member IEEE and José M. Molina, Member IEEE

Abstract— Soft computing methods, and Multi-Objective Evolutionary Algorithms (MOEAs) in particular, lack a general convergence criterion which prevents these algorithms from detecting the generation where further evolution will provide little improvements (or none at all) over the current solution, making them waste computational resources. This paper presents the Least Squares Stopping Criterion (LSSC), an easily configurable and implementable, robust and efficient stopping criterion, based on simple statistical parameters and residue analysis, which tries to introduce as few setup parameters as possible, being them always related to the MOEAs research field rather than the techniques applied by the criterion.

I. INTRODUCTION

The applicability of soft computing methods is still arguable in some domains, especially those related to high computational demands for each running generation. Actual industrial improvements often involve several parameters (which are usually in conflict) which require to be optimized jointly. These situations are known as Multi-objective optimization problems (MOOPs) [1]. The application of evolutionary algorithms (EAs) has achieved important accomplishments resolving these problems, classifying these applications as Multi-objective evolutionary algorithms [2] (MOEAs).

There are a number of underlying factors regarding the applicability of such algorithms, where the lack of a convergence criterion was among the first ones which researchers had to deal with [3]. The traditional solution to this problem was the establishment of an *a priori* number of maximum generations for the algorithm [4], which, at least, managed to set a boundary for the algorithm's running time. This need for a stopping criterion has already been stated and approached in general for EAs [5], [6].

This handicap is even more important in the case of MOEAs, where setting that *a-priori* value can be a particularly difficult task to be performed accurately. This issue is a MOOP itself, where the objectives are to maximize the quality of the optimal Pareto front (OPF) approximation while minimizing the number of generations, or, in general, the number of function evaluations.

All the authors belong to the Group of Artificial Intelligence (GIAA), Department of Computer Science, University Carlos III of Madrid. Av. de la Universidad Carlos III, 22, Colmenarejo, 28270, Madrid, Spain. <http://www.giaa.inf.uc3m.es>. {jguerrer, lmarti}@inf.uc3m.es, aberlan@ia.uc3m.es, jgherrer@inf.uc3m.es, molina@ia.uc3m.es. This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02

In order to measure the quality of an approximation to its correspondent OPF several community accepted indicators have been introduced [7], even though it is also reported to be a particularly complex task [8]. These indicators have lead to algorithms which accomplish the first objective of the presented solution to the convergence criterion issue: maximizing the quality of the approximation to the OPF [9].

This first approach to stopping criteria is usually defined as local (or iteration-wise) criteria, as only the knowledge of the current generation is used to determine if it has reached the stopping generation. They exhibit several handicaps: they require knowing the OPF *a priori* (in order to compute the indicator's value) and they do not take into account the number of generations required to obtain the solution.

There are, on the other hand, stopping criteria based on the evolution of the algorithm across different generations, defined as global (or execution-wise) criteria. In relation to the proposed solution for the convergence criteria issue, these algorithms can maximize the quality of the OPF approximation by measuring the progress towards that solution in a relative way, and, at the same time, provide a solution which can satisfy restrictions over the number of generations, by analyzing the amount of progress achieved in every new generation and setting the appropriate thresholds.

Quality indicators, such as the hypervolume [10] or the epsilon indicator [11], can be used, as has been previously stated, as a mean to resolve convergence criteria in terms of approximation to the OPF (even though they were originally created as performance indicators). They can be, on the other hand, redefined as progress indicators [12] in order to be used as part of global criteria. With this reformulation, they will measure the improvement of the solution over a number of generations, instead of the quality of the Pareto front obtained (which eliminates the restriction of knowing the OPF beforehand, making them virtually applicable to any problem).

There have been, as well, progress indicators specially designed for this purpose (instead of derived from quality ones), such as the mutual domination rate [13] (MDR), along with its associated stopping criterion, MGBM [14]. The main advantage of these synthesized progress indicators, opposed to the derived ones, is their low computational complexity.

The balance between the quality of the solution of a MOEA and the number of the generations it has been running for can be found, as has been exposed, with the use of global stopping criteria, but there are, as in any MOOP,

some decisions which the user must take in order to determine which solution he wants to use (stating it simply, which grade of progress towards the solution per generation is worth to keep the algorithm running). It must be noted, as well, that the traditional approach is not always better, in terms of solution quality, as additional generations may degrade factors such as the population diversity by means of genetic drift [15]. This implies that, by choosing the right stopping generation, the user is not only saving computational resources, but may be also preserving the quality of the obtained solution.

Any stopping criterion following the previous considerations must try to get as close as possible to the optimal stopping generation, but that concept is, as well, a non-trivial issue. Figure 1 shows some examples regarding it.

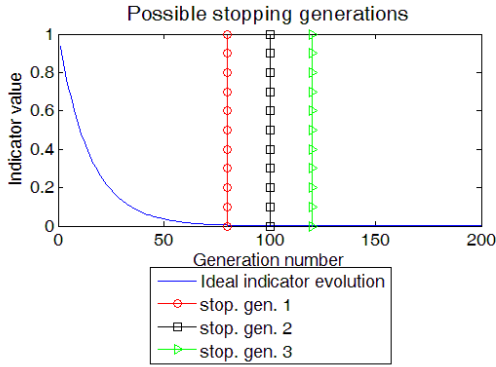


Fig. 1. Different possible stopping generations for a given indicator evolution.

There is no clear choice regarding the best stopping generation in Figure 1. In a general case (the one which the default configuration of a stopping criterion should be trying to achieve) the optimal stopping generation would be number two, but there may be additional considerations. If the running time of each individual generation is very high, the user might try to stop the algorithm following stopping generation number one (after which, even if there is some improvement in the solution quality, it is very low), whereas, on the other hand, if extreme accuracy in the solution is the key factor in the algorithm, the user might prefer stopping generation three. This shows that stopping criteria must be able to adapt to the user's needs.

The idea proposed in this paper is to obtain a simple and yet robust stopping criterion based on global criteria, also requiring as little configuration from the user as possible, in order to facilitate the inclusion of the criterion in any MOEA. This criterion can be based on different progress indicators, which we will show by testing its results with the three examples mentioned in this introduction. In order to test its performance under different situations, results of its applications to different standard algorithms and problems will be shown.

The second section of this paper will analyze the different situations that a convergence criterion must cover, being

followed by a section introducing different quality indicators and their adaptation to progress ones. The fourth section introduces the difficulties faced while designing global stopping criteria and summarizes some of the current approaches, setting the required basis for the fifth section, where our proposed criterion is presented. Finally, the sixth section presents the results over the different experimental sets, leading to the conclusions on the seventh section.

II. STOPPING CRITERIA IN MOEAS

In the introduction section of this paper we presented the idea that, in order to establish a stopping criterion over a MOEA, we require the information of an indicator in order to determine either the quality of our approximation to the OPF (in local criteria) or the progress towards the solution (in global criteria). An important consideration is that any of these indicators involves a dimensionality reduction, and this may lead to invalid conclusions [8].

There are four clear situations where a MOEA should be stopped [14], to which we may add a fifth one regarding the concepts we have already introduced:

1. The amount of computation is sufficient.
2. A solution obtained so far is satisfactory.
3. The solution is not satisfactory, but a better one is unlikely to be produced.
4. The method is not able to converge to a solution.
5. Additional computation will provide little or no improvements in the current solution.

The first situation was the one covered by traditional approaches, where the amount of computation was measured in number of generations. There have been modifications based on the number of function evaluations (FE) and the introduction of modeling methods, as an approach to more complete convergence criteria, but they still needed a number of fixed FE, which could be high (30000-500000) when the key parameter was the quality in the approximation [16] or low (130-250), in the case of model-assisted approaches [17]. One of the handicaps of this approach is that it may be dependent on several different parameters, such as the population size, the selection technique, the complexity of the fitness functions, etc.

The second situation is the one approached by quality indicators, which requires, as was pointed out in the introduction, to know the OPF *a priori* (in order to be applied automatically) or a decision maker which can estimate the quality of a given solution. The automatic application of this criterion involves the knowledge of the solution prior to the application of the algorithm, making it inapplicable to resolve new problems for which the OPF is not known (relegating this criterion to validation and comparison issues rather than using it as a general tool). On the other hand, a decision maker usually also requires a good knowledge over the expected solution in order to test the validity of a given Pareto front.

The differences between the following three situations are very subtle. The third situation presents a scenario where the

solution value has converged but is not satisfactory, in the fourth one the solution is no able to converge, while the fifth implies that the evolution in the improvement over different solutions at different generations has become not significant for the algorithm.

Quality indicators are required in order to automatically determine if the solution is unsatisfactory, but at the same time, some analysis of the progress towards the solution is required to determine if a better one is likely to be obtained or not. With these requirements, knowing the OPF would be required as well, but looking at the fifth situation, the reader may consider whether it is needed to know if the current solution is satisfactory or not if no further improvement can be gained by the application of the algorithm. This point of view allows us to summarize the three final scenarios into obtaining the best solution possible without considerations about its quality, at least as part of the stopping criteria.

The analysis of the previous stopping situations is not mutually exclusive. This means that, even if a global stopping criteria is applied, it is also recommended to set a maximum number of generations in order to limit the highest possible running time of the algorithm (even if better solutions could be obtained yet).

III. QUALITY AND PROGRESS INDICATORS

Quality indicators measure the quality of the current approximation to the OPF. This measuring involves three factors [7]:

1. Distance of the elements from the current Pareto front to the OPF (determining how close our solution is to the optima)
2. Distance from the OPF elements to their closest one in the current solution (determining how well the current solution covers the OPF)
3. Distribution of the elements in the current solution and the OPF (determining the spread of the elements on both sets)

From the factors above, we can determine that the main considerations in a quality indicator are distance and diversity. Among quality indicators, we will focus on binary ones [8], which compare two sets of solutions (in their original formulation, the current Pareto front and the OPF). The reason for this choice is the easy adaptation which can be performed over these quality indicators to convert them to progress ones.

The basic idea behind the conversion of a binary quality indicator to its respective progress one is based on their ability to compare two sets of solutions. Instead of comparing the current solution to the OPF we will compare two different solutions provided at different generations of our MOEA, thus measuring the progress we are obtaining. This simple idea allows us to use well established quality indicators as part of a stopping criterion framework.

We will describe next, even if only in a summarized way, the indicators which will be used in the experimental section of this paper. These indicators include two quality ones,

hypervolume [10] and additive epsilon [7]), and an especially designed progress indicator, MDR [13]. All of them are based on the objective space, even though the applicability of global stopping criteria could be, in theory, based on decision variable space indicators as well.

A. Hypervolume indicator

The hypervolume indicator $I_h(A)$ [10] computes the volume of the region H , delimited by a given set of points A , and a set of reference or nadir points N (points which dominate no other)

$$I_h(A) = \text{volume}(\cup_{\forall a \in A, \forall n \in N} \text{hypercube}(a, n)) \quad (1)$$

corresponding larger values of the indicator to better solutions. The binary indicator derived from the hypervolume concept [18] can be calculated with (2)

$$I_{HD}(A, B) = \begin{cases} I_h(B) - I_h(A) & \text{if } \forall x^2 \in B \exists x^1 \in A: x^1 \succ x^2 \\ I_h(A + B) - I_h(A) & \text{in any other case} \end{cases} \quad (2)$$

B. Epsilon indicator

The epsilon indicators [7], [11], are a set of performance indicators which, relying on the dominance concept, measure how close our current Pareto front is to the global optimal one. They introduce the epsilon dominance concept, which can be defined in additive or multiplicative terms. According to additive terms, it is defined as follows:

Having x_1 and $x_2 \in D$ (decision variable space), x_1 is said to ϵ -dominate x_2 (expressed as $x_1 \preceq_{\epsilon+} x_2$) if $f_j(x_1) \leq \epsilon + f_j(x_2)$

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

$$I_{\epsilon+} = \inf_{\epsilon \in \mathbb{R}} \{ \forall y \in B, \exists x \in A \text{ such that } x \preceq_{\epsilon+} y \} \quad (3)$$

C. Mutual domination rate indicator

The two previous indicators were, as we pointed out, reformulated to compare the solutions of two consecutive iterations. The main handicap of this approach is the high complexity of these indicators. The MDR indicator [13] is a specially created indicator to deal with this weakness. To simplify its definition, we will introduce a function $\Delta(A, B)$ that returns the set of elements of A that are dominated by at least one element of B .

The progress indicator $I_{\text{mdr}}(t) \in [-1, 1]$ contrasts how many non-dominated individuals of iteration t dominate the non-dominated individuals of the previous one and viceversa:

$$I_{\text{mdr}}(P_t^*, P_{t-1}^*) = \frac{\|\Delta(P^*[t-1], P^*[t])\|}{\|P^*[t-1]\|} - \frac{\|\Delta(P^*[t], P^*[t-1])\|}{\|P^*[t]\|} \quad (4)$$

If $I_{\text{mdr}}=1$, it means that the population of iteration t is completely better than the precedent one. If its value is 0, it implies that there has not been any substantial progress. If $I_{\text{mdr}}=-1$, it indicates the worst possible case, where the iteration t deteriorates its predecessor's solutions

IV. GLOBAL STOPPING CRITERIA

There is a recent concern about obtaining general stopping criteria which can be applied to a wide range of algorithms and problems [3], [12], [13], [14], [19], dealing especially with industrial applications [20]. There are, as well, different non-general approaches: design of special algorithms which can guarantee local optimality of solutions [21], using as well the gradient of the hypervolume to guarantee diversity and spread [22] (which may be considered a transformation of the hypervolume into a progress indicator), or the design of algorithm specific stopping criterion, based on values used by the selection criterion [23] (in this reference the authors use the crowding distance for an NSGA-II [24] based stopping criterion). While lacking the possibility of application to other algorithms, these criteria are very interesting due to the fact that the indicator value has already been calculated in the MOEA cycle, reducing the computational cost of the application of the stopping criterion.

Basically, the most common approach for general solutions is to identify when the evolution of the indicator becomes linear, where the tendency of that evolution (or, in other words, the amount of improvement over the solution per generation) falls below a certain threshold. This threshold is used as an application of the stopping scenarios presented, and the linear check determines the validity of the obtained tendency.

Obviously, the complete evolution of the indicator never follows that criterion (if it did, we would never reach an acceptable solution) so the name “global criteria” may be a misleading one, due to the fact that the algorithm will only be looking at a local portion of the indicator evolution each generation. This analysis window usually covers the value of the indicator for a certain amount of previous generations (which may change its value dynamically). Figure 2 shows this process:

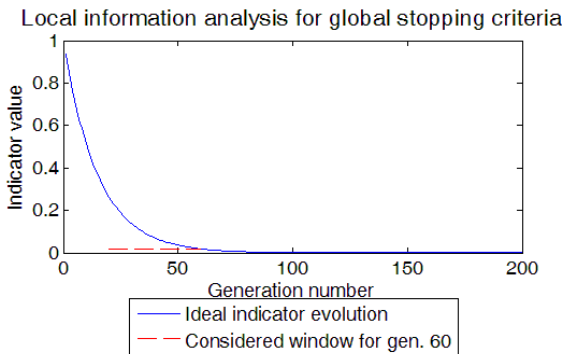


Fig. 2. Example of the local information analysis performed by global stopping criteria in order to determine the stopping generation.

In the example in Figure 2, the evolution of the indicator is clearly non-uniform in the considered window, and thus the criterion would probably determine that the MOEA must continue running. That idea can also lead us to some considerations about the window size these algorithms must

have in order to prevent inaccurate stops.

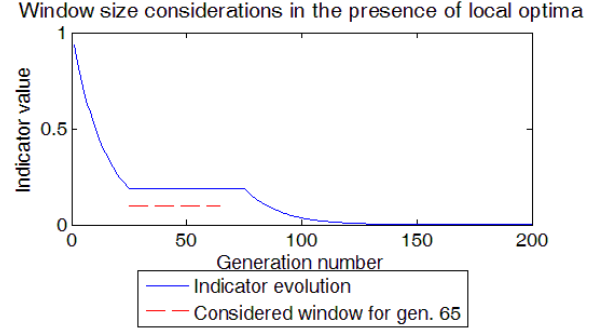


Fig. 3. Example of problematic situations for global criteria in the presence of local optima

The example in Figure 3 considers a window of the same size as Figure 2 (40 generations). With this size, regardless of the concrete technique applied, probably any generation from 65 to 75 would be considered to stop, even though, seeing the whole indicator evolution, we may notice that there are improvement chances after that stagnation of the indicator value. A bigger window size (depending on the technique applied, probably for 55 or more generations) would have been able to determine that the indicator, even though it had become very stable, was improving again. This situation shows that, if the problem is known or suspected to have local optima or different situations (such as neutral drift scenarios) where the indicator may reach a stagnation situation, the considered window size must be considerably larger, in order to be able to determine correctly if the improvement over the indicator value has stopped permanently.

The different current general online convergence criteria accomplish (or try to accomplish) this objective in different ways. In [13] a new progress indicator was introduced (MDR), along with a specific stopping criteria (MGBM) based on the analysis of the residue of a Kalman Filter [25]. To deal with the Gaussian noise sources (which, in fact, determine the distance to the measured values, and thus, the residue value) they chose fixed *a priori* values for the Kalman noise matrices (basically eliminating these sources).

In [12] Kalman filtering is used again, but, instead of analyzing the residue value, they focus on the corrections made by the filter in its update phase. This means that, assuming a uniform model, they set a maximum threshold for the corrections made by the model and track the value of those corrections, determining the stopping generation when they fall below the chosen boundary. The idea of using several progress indicators is also introduced, along with the possible application of data fusion architectures [26] for this purpose.

Focused on statistical testing, [3], [19], [20] determine the stopping criterion as a linear combination of the values of different indicators, using statistical tests for the analysis of the variance and the trend of their considered window over the different indicators' values.

An interesting parameter the commented general convergence criteria is the analysis of the computational complexity added by the stopping criterion used. The complexity of the indicators used is usually high enough to clearly exceed (and thus eliminate in the typical complexity order analysis) the one added by the stopping criterion. This is especially true when dealing with quality indicators.

V. LSSC: INTRODUCING A SIMPLE INTUITIVE STOPPING CRITERION

The reasons for the stop of a MOEA have already been discussed thoroughly, along with some state-of-the-art approaches to this task. The objective of the Least Squares Stopping Criterion (LSSC) proposed in this paper is to introduce a stopping criterion which can be easily implemented in any programming language (to facilitate the task of incorporating it to any MOEA) and easily configured by parameters related to MOEA's research (instead of those related to the particular techniques applied by the criterion).

The underlying idea is to determine when our progress indicator has reached a stagnation situation. To achieve this, we will base our criterion on two different considerations: the adjustment to a uniform model (by means of a least squares approximation, which gives its name to our criterion) and the value of the slope of that uniform model.

Simple Least Squares [27] is a basic linear regression method which approximates a variable according to the following model:

$$y = a + bx \quad (5)$$

It introduces some key assumptions, among which some of the most important are zero mean error and constant variance in the indicator value. These assumptions, which make this approach inapplicable to some real problems, fit our stopping needs (these are the circumstances under which we would like our algorithm to stop). Representing y the indicator value and x the generation number, a and b can be calculated with the following matrix-based formula:

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & w_l \end{bmatrix}^{-1} * \begin{bmatrix} \sum x_i * y_i \\ \sum y_i \end{bmatrix} \quad (6)$$

where w_l is the length of the chosen window. Once we have determined the values of the parameters for our linear regression, we need to define the normalized residue value:

$$res = \frac{\sum (y_i - (a + bx_i))^2}{w_l} \sim \frac{\chi^2}{w_l} \quad (7)$$

As shown in (7) the obtained residue follows a chi-square distribution which has $w_l - n$ degrees of freedom, where $n=2$ in our case (as we are imposing two linear restrictions, a and b parameters). This distribution has known mean and variance (presented normalized):

$$\mu = 1 - \frac{2}{w_l} \quad (8)$$

$$\sigma^2 = \frac{2}{w_l} - \frac{4}{w_l^2} \quad (9)$$

With the values in (8), (9), we may use Chebichev's inequality [26] to determine a threshold leaving bellow the 99% residue values coming from a uniform distributions.

$$thres = \mu + 3\sigma = 1 - \frac{2}{w_l} + 3 * \sqrt{\frac{2}{w_l} - \frac{4}{w_l^2}} \quad (10)$$

When the value of the residue shown in (7) falls below the threshold exposed in (10) we can consider that the evolution of the indicator has started to be uniform. This is a needed restriction to stop the algorithm's evolution, but not sufficient, as shown in Figures 4 and 5.

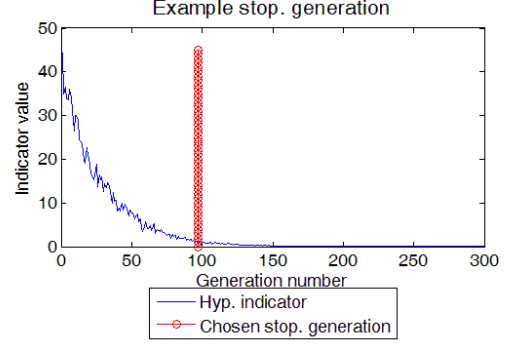


Fig. 4. Example of stopping generation vs indicator value based only on residue analysis (window size = 30)

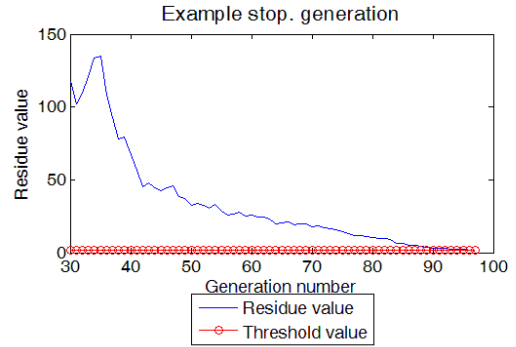


Fig. 5. Example of stopping generation vs residue value based only on residue analysis (window size = 30)

The complimentary restriction is the slope value, which will allow us to stop our evolution not only when it has become uniform, but also when the amount of that evolution per generation has become insufficient. Thus, the complete stopping criterion is defined with the following formula:

$$stop = \frac{\sum (y_i - (a + bx_i))^2}{w_l} < thres \wedge b < min_{prog} \quad (11)$$

The process to choose the min_{prog} value is rather simple: the researcher chooses a number of examples, determines which would be the ideal stopping generation for them and tunes the value of the min_{prog} parameter to guide the stopping criterion to stop at the chosen generation. This parameter value can then be applied to new problems and/or algorithms without requiring any modification in it (as it will be shown in the experimental section).

For the given example in Figures 4 and 5, which shows hypervolume indicator applied over a NSGA-II MOEA to the DTLZ3 problem, we have chosen an absolute value for the min_{prog} parameter of 0.002 (in fact this will be the used value for the parameter in the experimental section).

whenever we are using the hypervolume indicator, regardless of the problem or the algorithm). Figures 6 and 7 show the behavior of the stopping criterion in (11)

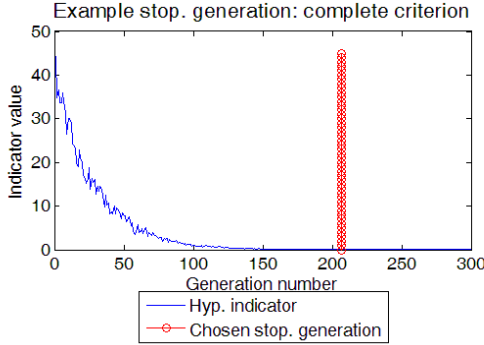


Fig. 6. Full stopping criterion example (stopping generation vs indicator value, window size = 30)

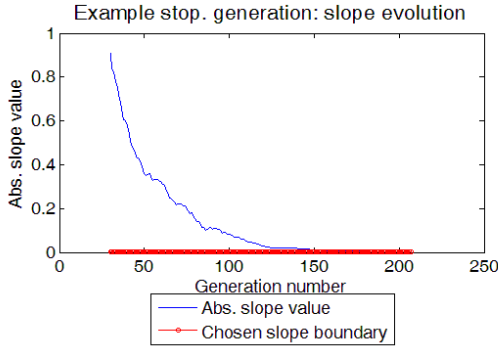


Fig. 7. Full stopping criterion example, absolute slope value evolution

The chosen value for the slope is fairly conservative, trying to stop only when future improvements would be marginal ones. Some users may prefer higher values, sacrificing small indicator improvements in favor of faster results. The dependency with the window size parameter, in absence of problematic local situations such as the ones presented in the global stopping criteria section, is not strong, only increasing, according to its value, the final stopping generation once the evolution of the indicator's value starts to satisfy the criterion conditions

It is important to remember that this stopping criterion is not necessarily applied on its own for a given algorithm: very usually, along with it, the user will add a different one regarding the maximum computation allowable for the algorithm (either in time, number of generations or number of function evaluations).

A. Computational complexity

At every generation (excluding the first $w_l - 1$) we have to obtain the following values: the linear regression parameters (a , b), the threshold value ($thres$) and the residue value (res). It is useful to invert the order in which the conditions are checked in (11), in order to obtain the res value only if necessary. For a fixed window size, the threshold value only has to be computed once, applying the same value for all the comparisons. Using standard libraries, both the regression

parameters and the residue value can be obtained in $O(w_l)$, being this the complexity order of the criterion.

In (6) we intentionally presented the required formula for the calculation of the two required linear regression parameters, due to the fact that, being composed of different summations which only differ in one term for consecutive generations, once the initial parameters have been calculated, the rest can be obtained with a constant order complexity:

$$\begin{bmatrix} b_i | b_{i-1} \\ a_i | a_{i-1} \end{bmatrix} = \begin{bmatrix} (\sum x_{i-1}^2) - x_{i-w_l}^2 + x_i^2 & (\sum x_{i-1}) - x_{i-w_l} + x_i \\ (\sum x_{i-1}) - x_{i-w_l} + x_i & w_l \end{bmatrix}^{-1} * \begin{bmatrix} (\sum x_i * y_i) - x_{i-w_l} * y_{i-w_l} + x_i * y_i \\ (\sum y_{i-1}) - y_{i-w_l} + y_i \end{bmatrix} \quad (12)$$

This does not change the worst case complexity order of the stopping criterion (the res value still has to be calculated in $O(w_l)$) but, by checking the min_{prog} condition first, it allows the criterion to run in $O(1)$ most generations, without requiring a complex implementation, and becomes specially interesting and advisable when the progress indicator is also required by the selection criterion (and thus integrated in the MOEA's usual cycle) or computationally inexpensive to calculate (such as MDR).

VI. EXPERIMENTAL VALIDATION

To ease the comparison process, the proposed stopping criterion is going to be tested with the algorithms and problems used in [12], [14]. The algorithm's set will include: NSGA-II [24], SPEA2 [28] and PESA [29], whereas the problems' set includes DTLZ3, DTLZ6 and DTLZ7 [30]. Both sets are considered to be representative enough for the validation purposes of this section. The details of the configuration of these algorithms and problems can be found in [14], whereas the actual data used are available online¹.

A. Criterion configuration

For the optimum configuration of the stopping generation factors such as the cost of running additional generations or the required accuracy in the final solution may be considered, but only the progress indicator used is required to be analyzed in order to determine the right slope value. Results regarding the three different presented progress indicators will be shown in this section, so three different slope values need to be configured. The chosen values are 0.002 (hypervolume) 0.0004 (epsilon) and 0.00002 (MDR).

These values have been chosen according to the process explained in the method's presentation section. Intuitively, these values can be related to the different ranges which the indicator exhibit. The window size will be constant regardless of the indicator used, and fixed at 30 generations. To determine the quality of the stopping generation obtained, we will compute the hypervolume difference

¹http://www.giaa.inf.uc3m.es/miembros/lmarti/_media/papers%3Bmarti-et-al-stop-crit-data.zip

compared to an *a priori* fixed generation (which will be chosen based on the problem's difficulty).

B. Results presentation

Each experiment has been run thirty times. We provide the statistical values of mean, standard deviation, minimum and maximum obtained (both for the stopping generation and the hypervolume difference with the *a priori* stopping generation), to verify the criterion performance.

The chosen values for the slopes were quite conservative, in order to obtain very accurate results. This can be observed in that the maximum hypervolume difference in mean value is 0,14, whereas in maximum value is 0,331 (both obtained for the hardest problem, DTLZ7). Even so, we have managed to obtain a stopping generation whose value is about 2/3 of its respective *a priori* one, with significantly similar performance results over the different indicators.

The comparison with the different alternatives proposed in the literature is difficult, due to the different use of algorithms, test problems and the lack of a clear optimal choice for the stopping generation. That is the reason why this results presentation is centered in the consistency and quality of the criterion presented in this paper.

1) DTLZ3

TABLE I
STOPPING GENERATION FOR DTLZ3 PROBLEM

Alg. / Indicator	Stopping generation				
	Mean	Min	Max	Std. D.	<i>A priori</i>
NSGA-II - epsilon	204,733	193	215	5,9186	300
NSGA-II - hypervolume	196	184	207	5,79536	300
NSGA-II- MDR	219,233	202	233	7,93371	300
PESA - epsilon	203,733	188	213	6,10219	300
PESA - hypervolume	210,533	194	223	7,394	300
PESA - MDR	200,367	185	212	6,58359	300
SPEA2 - epsilon	178,267	165	188	4,63073	300
SPEA2 - hypervolume	174,033	166	182	4,27892	300
SPEA2 - MDR	174,567	167	184	3,92765	300

TABLE II
HYPERVOLUME DIFFERENCE FOR DTLZ3 PROBLEM

Alg. / Indicator	Hypervolume difference			
	Mean	Min	Max	Std. D.
NSGA-II - epsilon	0,027	0,011	0,057	0,012
NSGA-II - hypervolume	0,040	0,016	0,073	0,015
NSGA-II- MDR	0,015	0,004	0,038	0,008
PESA - epsilon	0,049	0,016	0,108	0,021
PESA - hypervolume	0,046	0,015	0,096	0,020
PESA - MDR	0,055	0,027	0,110	0,022
SPEA2 - epsilon	0,024	0,007	0,056	0,013
SPEA2 - hypervolume	0,032	0,015	0,059	0,012
SPEA2 - MDR	0,027	0,011	0,060	0,014

2) DTLZ6

TABLE III
STOPPING GENERATION FOR DTLZ6 PROBLEM

Alg. / Indicator	Stopping generation				
	Mean	Min	Max	Std. D.	<i>A priori</i>
NSGA-II - epsilon	208,233	192	220	6,1346	300
NSGA-II - hypervolume	199,567	183	211	6,22389	300
NSGA-II- MDR	225,033	205	239	9,7432	300
PESA - epsilon	207,333	194	222	6,61937	300
PESA - hypervolume	213,467	203	231	5,74596	300
PESA - MDR	205,633	198	215	4,25468	300
SPEA2 - epsilon	183,767	176	193	4,11627	300
SPEA2 - hypervolume	179,367	174	186	3,36804	300
SPEA2 - MDR	177,1	166	185	4,24548	300

TABLE IV
HYPERVOLUME DIFFERENCE FOR DTLZ6 PROBLEM

Alg. / Indicator	Hypervolume difference			
	Mean	Min	Max	Std. D.
NSGA-II - epsilon	0,026	0,010	0,054	0,012
NSGA-II - hypervolume	0,045	0,017	0,081	0,015
NSGA-II- MDR	0,017	0,004	0,048	0,011
PESA - epsilon	0,053	0,020	0,101	0,023
PESA - hypervolume	0,044	0,018	0,087	0,017
PESA - MDR	0,057	0,024	0,094	0,020
SPEA2 - epsilon	0,020	0,006	0,044	0,009
SPEA2 - hypervolume	0,032	0,014	0,053	0,009
SPEA2 - MDR	0,030	0,014	0,049	0,010

3) DTLZ7

TABLE V
STOPPING GENERATION FOR DTLZ7 PROBLEM

Alg. / Indicator	Stopping generation				
	Mean	Min	Max	Std. D.	<i>A priori</i>
NSGA-II - epsilon	302,733	279	322	12,4123	425
NSGA-II - hypervolume	296,8	263	316	11,583	425
NSGA-II- MDR	332,233	300	368	16,3532	425
PESA - epsilon	301,067	264	322	12,956	425
PESA - hypervolume	313,867	288	337	12,125	425
PESA - MDR	310,433	276	335	12,7135	425
SPEA2 - epsilon	262,6	243	275	8,47145	425
SPEA2 - hypervolume	265,567	245	294	10,8522	425
SPEA2 - MDR	268,5	246	279	8,88916	425

TABLE VI
HYPERVOLUME DIFFERENCE FOR DTLZ7 PROBLEM

Alg. / Indicator	Hypervolume difference			
	Mean	Min	Max	Std. D.
NSGA-II - epsilon	0,094	0,033	0,181	0,041
NSGA-II - hypervolume	0,113	0,049	0,215	0,041
NSGA-II- MDR	0,047	0,004	0,109	0,027
PESA - epsilon	0,140	0,052	0,331	0,065
PESA - hypervolume	0,134	0,073	0,244	0,039
PESA - MDR	0,113	0,045	0,257	0,051
SPEA2 - epsilon	0,082	0,039	0,153	0,034
SPEA2 - hypervolume	0,093	0,025	0,170	0,035
SPEA2 - MDR	0,069	0,030	0,107	0,023

VII. CONCLUSIONS

The Least Squares Stopping Criterion is presented in this paper with the following attributes: easily configurable, (with only two required parameters, the acceptable amount of indicator variation per generation and the considered window), easily implementable (involving, basically, calculating least squares parameters and an Euclidean distance), robust (the same configuration of a progress indicator can be applied, as seen in the experimental section, to a wide range of different MOEAs and problems) and efficient (it can be implemented to run, for most generations, in constant time, or $O(n)$ in the worst generations), which we believe to be covered. Future lines involve the consideration of several progress indicators jointly, the application to different sets of problems and application to indicators based on the decision variables space.

REFERENCES

- [1] Ehrgott, M. "Multicriteria optimization". In Lecture notes in Economics and Mathematical Systems. Vol 491. Springer. 2005
- [2] Coello Coello, C. A., Lamont, G. B., Van Veldhuizen, D. A. "Evolutionary Algorithms for Solving Multi-Objective Problems" 2nd edition. Springer. 2007
- [3] Wagner T., Trautmann, H. and Naujoks, B. "OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing". Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science. Vol 5497. 198-215. Springer. 2009
- [4] Coello Coello, C.A. "An Updated Survey of GA-Based Multiobjective Optimization Techniques". ACM Computing Surveys 32(2):109-143. 2000
- [5] Safe, M., Carballido, J., Ponzoni, I., and Brignole, N., "On stopping criteria for genetic algorithms," in Advances in Artificial Intelligence. SBIA 2004. Berlin/Heidelberg: Springer, pp. 405-413. 2004,
- [6] Hernandez, G., Wilder, K. Nino, F. and Garcia, J. "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, pp. 615-620. 2005
- [7] Knowles, J, Thiele, L. and Zitzler, E. "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Computer Engineering and Networks Laboratory (TIK), ETH. 2006.
- [8] Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C. M., and Grunert da Fonseca V., "Why Quality Assessment of Multiobjective Optimizers Is Difficult," in GECCO'2002, Morgan Kaufmann Publishers, pp. 666-673. 2002.

- [9] Wagner, T., Beume, N., Naujoks, B.: "Pareto-, aggregation-, and indicator-based methods in many-objective optimization". In EMO 2007. LNCS, vol. 4403, pp. 742-756. Springer. 2007.
- [10] Zitzler, E., Brockhoff, D., and Thiele, L. "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighed integration". In EMO 2007. LNCS vol. 4403 pp 862-876. 2007.
- [11] Zitzler, E., Thiele, L., Laumanns, M., Fonseca C.M., da Fonseca, V.G. "Performance Assessment of multiobjective optimizers: An analysis and review". In Transactions on Evolutionary Computation. IEEE. 7(2). Pages 117-132. 2003
- [12] Guerrero, J. L., García, J., Martí, L., Molina, J. M., and Berlanga, A. "A Stopping Criterion Based on Kalman Estimation Techniques with Several Progress Indicators". In GECCO 2009, pp 587-594. 2009.
- [13] Martí, L., García, J., Berlanga, A., & Molina, J. M. "A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms (extended version)". In GECCO '07: pp 2845-2842. 2007.
- [14] Martí, L., García, J., Berlanga, A., and Molina, J. M. "An Approach to Stopping Criteria for Multi-objective Optimization Evolutionary Algorithms: The MGBM Criterion". In CEC 2009, pp 1263-1270, 2009.
- [15] Rudolph, G., Naujoks, B., Preuss, M.: "Capabilities of EMOA to detect and preserve equivalent pareto subsets". In EMO 2007. LNCS, vol. 4403, pp. 36-50. Springer 2007.
- [16] Deb, K., Mohan, M., Mishra, S.: "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions". KanGAL report 2003002, Indian Institute of Technology, Kanpur, India. 2003
- [17] Knowles, J.: "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems". IEEE Trans. on Evolutionary Computation 10(1), pp 50-66. 2006
- [18] Zitzler, E., Thiele, L. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach". IEEE Transactions on Evolutionary Computation 3(4):257-271. 1999
- [19] Trautmann, H., Wagner, T., Naujoks, B., Preuss, M., Mehnen, J. "Statistical Methods for Convergence Detection of Multi-Objective Evolutionary Algorithms". Evolutionary computation, Vol. 17, n. 4, pp 493-509. 2009.
- [20] Naujoks, B. and Trautmann, H. "Online Convergence Detection for Multiobjective Aerodynamic Applications". In 2009 IEEE Conference on Evolutionary Computation (CEC), pp 332-339, 2009.
- [21] Deb, K., Lele, S., Datta, R.: "A hybrid evolutionary multi-objective and SQP based procedure for constrained optimization". ISICA 2007. LNCS, vol. 4683, pp. 36-45. Springer, 2007.
- [22] Rudenko, O. and Schoenauer, M. "A steady performance stopping criterion for pareto-based evolutionary algorithms," Multi-Objective Programming and Goal Programming 2004
- [23] Emmerich, M., Deutz, A., Beume, N.: "Gradient-based/Evolutionary relay hybrid for computing pareto front approximations maximizing the S-metric". In HCI/ICCV 2007. LNCS, vol. 4771, pp. 140-156. Springer, 2007
- [24] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". In IEEE Transactions on Evolutionary Computation. 6(2), pp 182-197. 2002
- [25] Welch, G. and Bishop, G. "An Introduction to the Kalman Filter". TR 95-041. Department of Computer Science. University of North Carolina at Chapel Hill. 2006.
- [26] Groves, P. D. "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems". Artech House, 2008.
- [27] Meyer, P. L. "Introductory Probability and Statistical Applications". Addison Wesley, 2nd Edition, 1970.
- [28] Zitzler, E., Laumanns, M., Thiele, L. "SPEA2: Improving the Strength Pareto Evolutionary Approach". In EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems", pp 95-100. 2002
- [29] Come, D.W., Knowles, J.D., Oates, M.J. "The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization". In "Proceedings of the Parallel Solving from Nature VI Conference". Lecture Notes in Computer Science. No. 1917, pp 839-848. 2000
- [30] Deb, K., Thiele, L., Laumanns, M., Zitzler, E. "Scalable test problems for evolutionary multiobjective optimization". In "Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Advanced Information and Knowledge Processing" pp 105-145. Springer. 2005