A Cumulative Evidential Stopping Criterion for Multiobjective Optimization Evolutionary Algorithms

Luis Martí, Jesús García, Antonio Berlanga and José M. Molina
GIAA, Dept. of Informatics, Universidad Carlos III de Madrid
Av. Universidad Carlos III 22, Colmenarejo 28270 — Madrid, Spain
{lmarti,jgherrer}@inf.uc3m.es; {aberlan,molina}@ia.uc3m.es

ABSTRACT

In this work we present a novel and efficient algorithm–independent stopping criterion, called the MGBM criterion, suitable for Multiobjective Optimization Evolutionary Algorithms (MOEAs).

The criterion, after each iteration of the optimization algorithm, gathers evidence of the improvement of the solutions obtained so far. A global (execution–wise) evidence accumulation process inspired by recursive Bayesian estimation decides when the optimization should be stopped. Evidence is collected using a novel relative improvement measure constructed on top of the Pareto dominance relations. The evidence gathered after each iteration is accumulated and updated following a rule based on a simplified version of a discrete Kalman filter.

Our criterion is particularly useful in complex and/or high-dimensional problems where the traditional procedure of stopping after a predefined amount of iterations cannot be used and the waste of computational resources can induce to a detriment of the quality of the results.

Although the criterion discussed here is meant for MOEAs, it can be easily adapted to other softcomputing or numerical methods by substituting the local improvement metric with a suitable one.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search; I.2.m [**Artificial Intelligence**]: Evolutionary Computing and Genetic Algorithms—*Multiobjective Evolutionary Algorithms*

General Terms

Algorithms, Experimentation, Performance

Keywords

Stopping Criterion, Multiobjective Optimization Evolutionary Algorithms (MOEAs)

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION

Softcomputing, heuristic, non–deterministic or numerical methods require a stopping criterion. This criterion, which usually is a heuristic itself, is responsible for minimizing the waste of computational resources in limit situations by detecting the scenarios where there is no sense in proceeding with the execution of the method.

In particular it should identify scenarios like:

- the solution yielded so far is satisfactory;
- the method is able to produce a solution but it is not satisfactory but a better one will not be produced, and;
- the method is unable to converge to any solution.

One particular subclass of the above mentioned methods, the Multiobjective Optimization Evolutionary Algorithms (MOEAs) [3], also require a stopping criterion. However, the formulation of an effective criterion is particularly complex in the case of the multiobjective optimization problem as judging the advance of the optimization can become as complex as the optimization itself and it can be very resource consuming. Furthermore, in this class of problems there is no "axis reference" in which base the heuristic, unlike other problems like function approximation, pattern recognition, etc.

Probably because of these issues, the formulation of an efficient stopping criterion for MOEAs has been left aside, although it has been repeatedly nominated as one of the fundamental topics that must be properly addressed in this research area [1, 2].

Stopping criteria can be grouped in local (iteration–wise) criteria, that work with data local to each iteration of the method, and global (execution–wise) criteria that keep track of the process advance in order to make decisions relying on the long–term behavior of the method.

For MOEAs, a local criterion must measure the distance between the current and optimal Pareto front and decide when they are close enough. This type of criterion has the obvious paradoxical shortcoming of requiring a known optimal Pareto front beforehand.

Global approaches can apply relative improvement metrics in order to assess the progress by analyzing the algorithm itself and without having to resort to an absolute comparison with an a priori established threshold. In the particular case of MOEAs this type of criteria compares the Pareto sets yielded by different iterations in order to determine how the optimization process is progressing.

In this work we introduce a global stopping criterion, called *MGBM criterion*, that gathers relative local measurements of the process advance to determine when the algorithm should be stopped. This combination of local and global processes makes the criterion particularly suitable for MOEAs. Our criterion accumulates evidence of algorithm progress or convergence in a process inspired by recursive Bayesian estimation. Evidence is collected using a novel relative improvement metric constructed on top of the dominance relations that is also discussed here. The evidence gathered after each iteration is accumulated and updated following a rule based on a discrete Kalman filter with a simple dynamic model.

It should be noted that, although the criterion discussed here is meant for MOEAs, it can be easily adapted to other softcomputing or numerical methods by substituting the local improvement metric with a suitable one.

As part of this work we will briefly present the background matters required for our discussion. Afterwards we will discuss the existing approaches to this issue. Then we will lay out our proposal thoroughly commenting it from a theoretical point of view. Subsequently, a set of synthetic experiments are performed in order to review the properties of the criterion from a practical scope. Finally some concluding comments and remarks are formulated.

2. THEORETICAL BACKGROUND

For the sake of completeness and to establish the notation we now briefly introduce the theoretical foundations of our work. However, we recommend the interested reader to consult the cited references for more comprehensive explanations.

2.1 Multiobjective Optimization Evolutionary Algorithms

The concept of multiobjective optimization refers to the process of finding one or more feasible solutions of a problem which corresponds to the extreme values (either maximum or minimum) of two or more objective functions subject to a set of restrictions. This problem can be expressed as

$$\begin{array}{rl} \min & f_1(\mathbf{x}),\ldots,f_M(\mathbf{x})\,, \\ \text{subject to} & g_1(\mathbf{x}),\ldots,g_L(\mathbf{x}) \leq \mathbf{0}\,, \\ & \text{with } \mathbf{x}\in\mathcal{S}\,, \end{array} \tag{1}$$

where $\mathcal{S}$ is known as the decision space.

For these problems does not exist a unique optimal solution. Instead, a set of compromise solutions must be yielded by the optimization algorithm applied. In order to define the set of optimal solutions a so called dominance relation has been defined.

Definition 1. *A solution $\mathbf{x}_1$ dominates $\mathbf{x}_2$ ($\mathbf{x}_1 \preceq \mathbf{x}_2$) iff $\forall f_j$, $f_j(\mathbf{x}_1) \leq f_j(\mathbf{x}_2)$ and $\exists f_i$ such as $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$.*

The set of non–dominated elements of $\mathcal{S}$ is the solution of the problem stated in (1) and is called the Pareto–optimal set. Correspondingly, the image of this set in objective space is called Pareto–optimal front, $\mathbf{P}^*$.

2.2 Kalman Filters

The Kalman filter [7, 10] provides an efficient computational means to estimate the state of a dynamic system from a series of incomplete and noisy measurements. It is very powerful since it supports estimations of past, current, and future states, and it can do so even when the nature of the modeled system is not completely known.

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:

- $\hat{x}_t$, 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 of making an *a priori estimation* of the future state of the system by relying on the current state and error covariance estimates. The update phase is responsible for the feeding back the (noisy) measurement of state of the system to obtain an improved *a posteriori estimate*.

The Kalman filter assumes a dynamic model given by

$$x_t = Ax_{t-1} + Bu_t + w_t\,, \tag{2}$$

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

Additionally the measurement process is modeled by

$$z_t = Hx_t + v_t\,, \tag{3}$$

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.

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

$$\hat{x}_t^- = A\hat{x}_{t-1}^- + Bu_t\,, \tag{4}$$

$$P_t^- = AP_{t-1}^- A^T + Q\,. \tag{5}$$

Then the update phase proceeds by computing the Kalman gain,

$$K_t = \frac{P_t^- H^T}{HP_t^- H^T + R}\,. \tag{6}$$

Having this, the a posteriori estimation is calculated as the feedback is injected to the filter as

$$\hat{x}_t = \hat{x}_t^- + K_t\left(z_t - H\hat{x}_t^-\right)\,. \tag{7}$$

The next and final step of this phase is to obtain an a posteriori error covariance estimate

$$P_t = (I - K_t H)P_t^- \tag{8}$$

3. APPROACHES TO THE ISSUE

As we already stated in the introduction of this paper the stopping criterion issue has been repeatedly left aside in the framework of MOEAs.

The approach used by almost all the works is simply to stop when the algorithm reaches a given number of iterations. This class of solution could be valid for simple, low–dimensional problems where a simple trial–and–error determination of the required amount of iterations is feasible. However, more complex problems cannot afford it since such procedure becomes computationally unfeasible.

To the best of our knowledge, the stopping criterion issue has been only previously addressed by Rudenko and Schoenauer [11]. In their work the authors present a local stopping criterion that computes a stability measure of the spread of the non–dominated solutions after each iteration that seems to be bound to the NSGA–II algorithm.

In spite of novelty of the proposed stability measure, the local scope of the criterion hinders the overall comprehension of progress of the optimization algorithm.

4. THE MGBM CRITERION

As exposed, the MGBM criterion combines a local improvement metric and a global evidence accumulation criterion that decides when the evolution of values yielded by the local metric indicate that when the algorithm should be stopped.

4.1 Local Improvement Metric

For the evidence gathering we use a metric is based on the set of non–dominated solutions of two consecutive iterations, $\mathbf{ND}_t$ and $\mathbf{ND}_{t-1}$.

In order to simplify the explanation we introduce the function $\Delta\left(\mathbf{A},\mathbf{B}\right)$ that returns the set of elements of $\mathbf{A}$ that are dominated by at least one element of $\mathbf{B}$. Expressing it in a more formal fashion.

$$\mathbf{C} = \Delta\left(\mathbf{A},\mathbf{B}\right)\ , \tag{9}$$

such that

$$\forall x \in \mathbf{C}, x \in \mathbf{A},\ \text{and}\ \exists y \in \mathbf{B}\ \text{with}\ y \preceq x\,. \tag{10}$$

The progress indicator $s_t \in [-1, 1]$ contrasts how many non–dominated individuals of the iteration t dominate the non–dominated individuals of the previous one $(t-1)$ and vice versa,

$$s_t = \frac{\|\Delta\left(\mathbf{ND}_{t-1},\mathbf{ND}_t\right)\|}{\|\mathbf{ND}_{t-1}\|} - \frac{\|\Delta\left(\mathbf{ND}_t,\mathbf{ND}_{t-1}\right)\|}{\|\mathbf{ND}_t\|}\,. \tag{11}$$

The s_t indicator provides different types of information. If $s_t = 1$ it means that the population from iteration t is completely better than the precedent one. The case $s_t = 0$ implies that there has not been any substantial progress. The worst case, $s_t = -1$, indicates that iteration t does not improves any of the solutions of its predecessor.

As we mentioned in the introduction of this work a stopping criterion should be able to discover three situations where the algorithm execution should be stopped. From the stopping point of view all this situations can be interpreted as if there is or if there is not progress of the algorithm.

The determination of the non–dominated individuals after each iteration can be computationally expensive. However, as most MOEAs extract them for their own purpose it would be reasonable to embed this part of the criterion into the MOEAs being used.

4.2 Accumulating Evidence

Our approach is based on the recursive estimation prediction/update framework proposed by Kalman filters. Because of this we will be assuming that noise process present in the measured progress indicator is uncorrelated between consecutive iterations. Furthermore, the estimated value of the progress indicator and its associated covariance follows a Markov process and therefore the outcome of each iteration is only dependant on the previous one.

In our case we keep track of the algorithm progress indicator at iteration t, s_t.

After each iteration, we compute the *a priori* estimated indicator $\hat{s}_t^-$ by using a simple version of the dynamic model (12) with $A = 1$ and $B = 0$. This implies that we are taking a positivist stance and predicting that the indicator will remain the constant across iterations and therefore equal to the *a posteriori* estimation, $\hat{s}_{t-1}$,

$$\hat{s}_t^- = \hat{s}_{t-1}, \tag{12}$$

eliminating the control input, u, as there is no external interaction.

As this is purely computational process we can disregard the prediction error in our dynamic model $(Q = 0)$. Correspondingly, the a priori error covariance becomes

$$P_t^- = P_{t-1}\,. \tag{13}$$

We then rewrite (3) as

$$z_t = s_t. \tag{14}$$

where s_t is calculated following (11). Here we assume that s_t contains both the measurement and measurement error components of (3) and that the measurement error has a Gaussian nature.

The correction step of the process results as

$$K_t = \frac{P_t^-}{P_t^- + R}\,. \tag{15}$$

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

$$\hat{s}_t = \hat{s}_t^- + K_t(z_t - \hat{s}_t^-)\,. \tag{16}$$

The stopping criterion will fire if the a posteriori estimation $\hat{s}_t$ falls bellow a certain threshold

$$\hat{s}_t < \hat{s}_{\min}\,. \tag{17}$$

In particular, we would be interested in stopping when no further progress is predicted, situation that is represented by $\hat{s}_{\min=0}$.

4.3 Algorithmics of the Criterion

Relying on the equations introduced above we can formulate the algorithmic schema of our stopping criterion. This algorithm is sketched in Algorithm 1.

Along with the positivist stance expressed before when formulating (12) will use as the initial a posteriori progress estimation, $\hat{s}_0$, equal to 1. This means that we will be assuming a full progress from the start and let this indicator decay as the process advances.

On the other hand, we have not yet demonstrated the convergence of our criterion and, therefore, there is no theoretical guaranty of the stopping of the optimization process. This implies that a maximum limit to the amount of iterations, $t_{\max}$, must be used as a safety measure.

The remaining issue is the choice of the process noise covariance R, that in our case represents degree of inertia of the system. As this is the only free parameter of the criterion to choose it incorrectly could lead to an undesired behavior. In the next section we show how the performance and robustness of the criterion under different values of R.

Algorithm 1 Algorithmic description of the stopping criterion.

```
Initialize $t = 0$ and $\hat{s}_0 = 1$.
Set $R$,
Set $t_{\max}$, the maximum amount of iterations.
Set $\hat{s}_{\min}$, the minimum accepted value of the a posteriori estimation.
while $\hat{s}_t \geq \hat{s}_{\min}$ and $t < t_{\max}$ do
    Execute one iteration of the MOEA.
    $t = t + 1$.
    Compute the a priori progress estimation, $\hat{s}_t^-$, following (12).
    Calculate measured rate of improvement, $z_t$, as specified in (11) and (14).
    Determine the a posteriori estimation $\hat{s}_t$ from equations (13)–(16).
end while
```

5. EXPERIMENTS

We will now experimentally illustrate the accuracy of the MGBM criterion by analyzing the performance of the estimations in a set of experiments.

In particular we will present the results of applying two well established MOEAs: the Nondominated Sorting Genetic Algorithm II (NSGA–II) [4] and the improved Strength Pareto Evolutionary Algorithm (SPEA2) [15] when solving two scalable multiobjective test problems: the DTLZ3 and DTLZ7 [5, 6] problems under different initial conditions.

The initial conditions intend to a priori bias the algorithm so we can test whether our criterion can resolve all possible end situations and while we also study the impact of the criterion parameters.

5.1 Test Problems

The DTLZ3 and DTLZ7 problems are part of a family of scalable multiobjective test problems originally introduced to study and compare the performance of different MOEAs in problems in high–dimensional problems.

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

The DTLZ3 problem is a M–objective problem with a n–dimensional decision vector.

The Pareto–optimal front lies on the first orthant of a unit hypersphere (see Figure 1(a) for a 3-D representation). This problem was introduced to test the ability of a MOEA to converge to the global Pareto–optimal front, since there are $3^{n-M+1} - 1$ suboptimal fronts parallel to the optimal one.

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

A 3-D graphical representation of the Pareto–optimal front of DTLZ7 is presented on Figure 1(b).

5.2 Algorithms

An exhaustive description of NSGA–II and SPEA2 is out of the scope of this work and, on the other hand, impossible due to the length restrictions. Therefore, we will now briefly describe them and state how they were configured in our experiments.

Both algorithms exploit elitism by explicitly keeping a population with non–dominated solutions, so the selection operators are based on comparisons with them. Both SPEA2 and NSGA-II use density information in order to guide the search together with the Pareto-dominance sorting, but have different approaches when implementing elitism and density estimation.

The Strength Pareto Evolutionary Algorithm (SPEA) [16] implements elitism by preserving an *external population*. This population stores a fixed amount of non–dominated individuals discovered since the beginning of the simulation. After every iteration of the algorithm, if a new non–dominated solution is found it is compared with the ones present in the external population to preserve the best solutions.

SPEA goes beyond than just keeping an elite set of solutions. It uses the solutions stored along with the dominated solutions in all genetic operations with the hope of inducing a better performance of the search in the solution space.

Although SPEA have produced a number of relevant results it has been pointed out that it has some potential weaknesses [15]. SPEA2 [14, 15] was proposed as an attempt to overcome the limitations of SPEA. It keeps the overall scheme of its predecessor but, in contrast to SPEA, SPEA2 uses a fine–grained fitness assignment strategy which incorporates density information. Furthermore, the external population has a fixed size; 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 which has similar features but does not miss boundary points.

The NSGA-II algorithm is an improvement over the original NSGA [12,13]. There are two key concepts in the NSGA family: fast non-dominated sorting of the population and a crowding distance calculation for maintain diversity in the population.

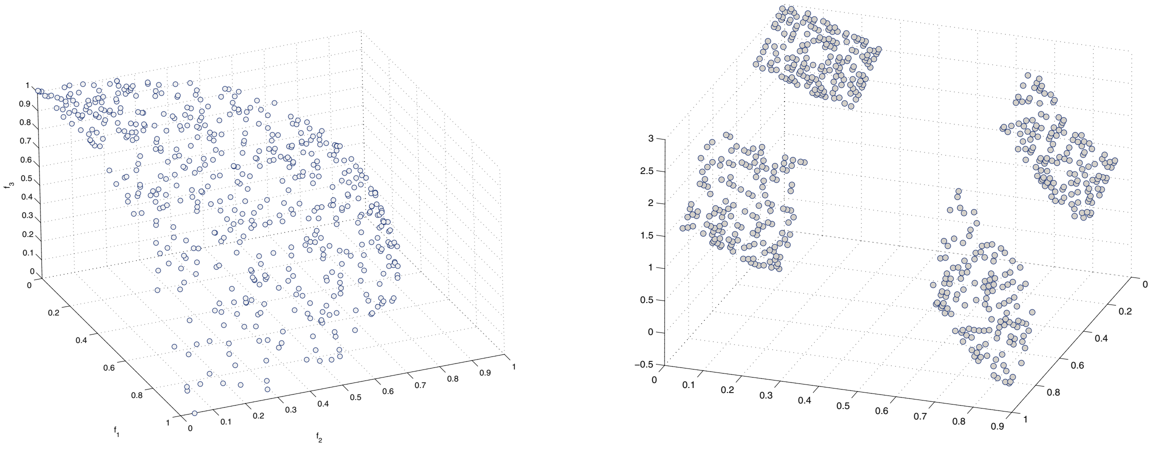
NSGA–II introduces a faster algorithm to sort the population that takes $O(Mn^2)$ computations, instead of the original $O(Mn^3)$ of NSGA, where M is the number of objectives and n is the number of population members. A crowding distance considers the size of the largest cuboid enclosing each individual without including any other of the population. This feature is used to keep diversity in the population and points belonging to the same front and with higher crowding distance are assigned a better fitness than those with lower crowding distance, avoiding the use of the fitness sharing factor.

For the sake of reproducibility and to compare results we used in this study the values of the internal parameters as the ones described in [9] except for the population size that was set to 100 elements.

5.3 Experiments Results

As we already stated we are interested in showing that our criterion can resolve positive and negative end conditions. With that in mind we have devised a set of experiments that test our criterion in a controlled environment.

We have grouped these experiments as expected successes



(a) DTLZ3 test problem.

(b) DTLZ7 test problem.

Figure 1: Sample of the Pareto–optimal front of DTLZ3 and DTLZ7 test problems with three objectives ($M = 3$).

and expected failures. The results of each set of experiments are detailed in the rest of this section.

It should be noted that all the results presented here are the average over 30 run of the algorithms under the same initial conditions.

5.3.1 *Expected Successes*

In these tests we will plot the mean over all simulations of the distance of the set of non–dominated solutions yielded by an iteration t, $\mathbf{ND}_t$, to the Pareto–optimal front, $\mathbf{P}^*$.

On these plots we will then point out the moments at which the criterion suggested an algorithm stop. To test the robustness of the criterion we used in this analysis different values of R. In particular $R = 0.05$, $R = 0.1$ and $R = 0.15$.

When calculating the distance, if the Pareto–optimal front is not available directly from the problem statement, as is the case of problem DTLZ7, it must be substituted by a set $\check{\mathbf{P}}^*$ sampled elements of $\mathbf{P}^*$. In this case the distance is measured from element i of $\mathbf{ND}_t$ and the closest element in $\check{\mathbf{P}}^*$,

$$d_i = \min_{j=1}^{|\check{\mathbf{P}}^*|} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(i) - f_k(j)}{f_k^{\max} - f_k^{\min}} \right)^2}\,, \tag{18}$$

where $f_k^{\max}$ and $f_k^{\min}$ are the maximum and minimum values of the k–th objective function.

For the case of the DTLZ3 problem this distance can be simplified as it is the distance from a point i of $\mathbf{ND}_t$ to the first orthant of a hypersphere of radius 1. Therefore the distance can be stated as the norm of the vector of objective functions minus 1:

$$d_i = \|\{f_1(i), \dots f_M(i)\}\| - 1\,. \tag{19}$$

Having computed the distance of each element of $\mathbf{ND}_t$ they are combined as the combined measure

$$D_t = \frac{\sum_{i=1}^{|\mathbf{ND}_t|} d_i}{|\mathbf{ND}_t|} \tag{20}$$

that is a convergence metric that represent how close a $\mathbf{ND}_t$ is to the optimum.

In order to assure the success of the algorithms three dimensional functions were used. The algorithms were left to run for 500 iterations, what was found sufficient in the initial trials.

The results when solving the DTLZ3 problem using NSGA–II and SPEA2 can be observed in Figures 2(a) and 2(b) respectively. Correspondingly, the results for DTLZ7 are shown on Figures 2(c) and 2(d). The plots show the distance after each iteration from the obtained set of non–dominated solutions to the Pareto-optimal set. The points in time where the MGBM criterion suggested to stop (for different values of R) are marked with a circle.

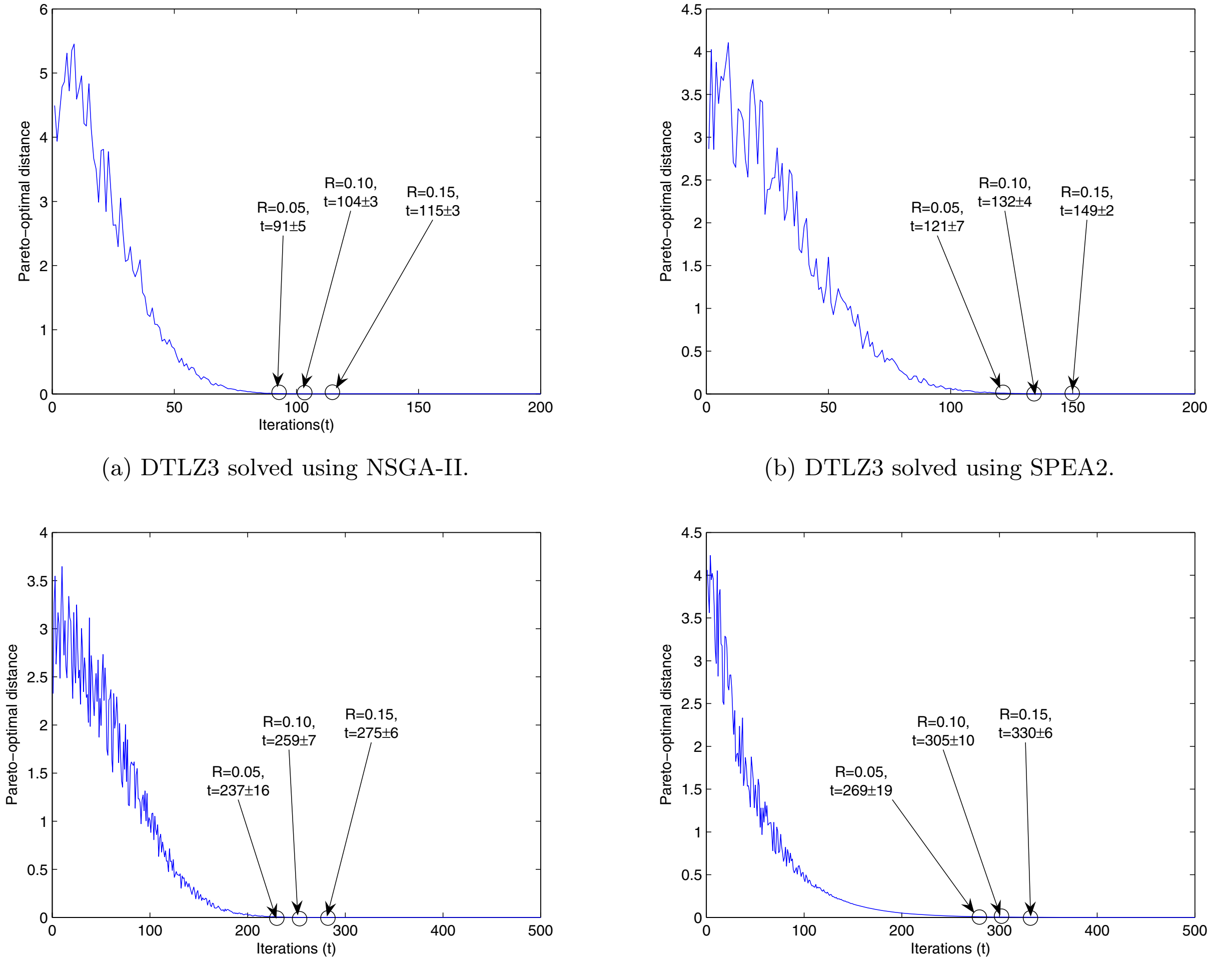
In both cases the stopping criterion fired when the algorithms have become stable and close to the optimum. Depending on the value of R the criterion had a quicker or more inertial response.

It should be noted that in similar DTLZ3 tests performed by Khare [8] the algorithms were left to run for more iterations than the ones suggested by the criterion. A set of analogous DTLZ3 and DTLZ7 tests were performed by Deb et. al. [6] with the same population size as ours but with unspecified internal parameters. For DTLZ3 our criterion also suggested to halt the optimization with less iterations than the ones used. However, in the case of DTLZ7 the criterion suggested to keep running the processes for a longer term than the one used; indicating that further processing was need to reach the optimum. These results are summarized in Table 1.

5.3.2 *Expected Failures*

In these tests we force the failure of the algorithms by attempting to solve problems whose complexity exceeds their capacity as they are configured.

For these tests, to show the progress of the algorithms by measuring the distance of the $\mathbf{ND}_t$'s to $\mathbf{P}^*$ lacks sense since it would not show any usefull information.



(a) DTLZ3 solved using NSGA-II.

(b) DTLZ3 solved using SPEA2.

(c) DTLZ7 solved using NSGA-II.

(d) DTLZ7 solved using SPEA2.

Figure 2: Testing the MGBM criterion when solving the DTLZ3 and DTLZ7 problems with combination of initial parameters that assured the algorithms success. The plots show the distance after each iteration from the obtained set of non–dominated solutions to the Pareto-optimal set. The points in time where the criterion suggested to stop (for different values of R) are marked with a circle.

Instead it is more interesting to plot the variation of the a posteriori progress estimation, $\hat{s}_t$ as it will show how it moves towards zero.

Again, different values of R (0.05, 0.1 and 0.15) were used. To induce the failure a 10–objective DTLZ3 test problem is used. The algorithms are left to run for 200 iterations as it allows to have a clear perspective of the evolution of the values of $\hat{s}_t$.

The results of these tests can be observed in Figure 3. There it can be noticed how depending on R the estimation of progress descend with a bigger o lesser rate until it points out that there is no progress and the algorithm execution can be stopped.

6. FINAL REMARKS

On this work we have presented a novel stopping criterion to be used in multiobjective optimization problems. The criterion after each iteration of the optimization algorithm gathers evidence of the improvement of the solutions obtained so far. A global (execution–wise) evidence accumulation process decides when the optimization should be stopped.

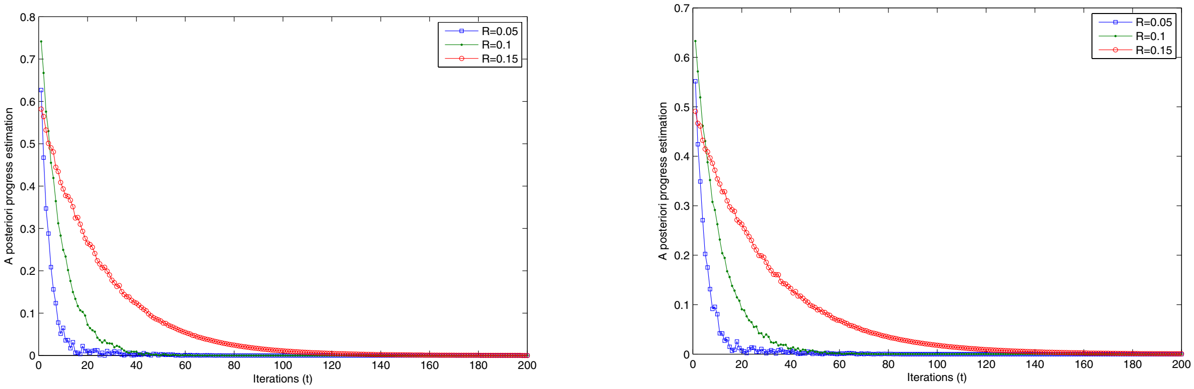
Our criterion is particularly useful in complex and/or high-dimensional problems where the traditional procedure of stopping after a predefined amount of iterations cannot be used and the waste of computational resources can induce to a detriment of the quality of the results.

As part of this paper we have described the criterion theoretically and have examined its performance in some test problems. However, since this is an initial approach to this issue some questions remain yet to be properly handled. For example, a better understanding of the impact of the R parameter must be achieved and a proper study on the complexity of the evidence accumulation algorithm must be made.

One salient issue is the interpretation of the final state of

Experiment	MGBM stop	Deb et. al.	Khare
DTLZ3 using NSGA–II	91,104,115	500	500
DTLZ3 using SPEA2	121,132,149	500	500
DTLZ7 using NSGA–II	237,259,275	200	—
DTLZ7 using SPEA2	269,305,330	200	—

Table 1: Stop iterations suggested by the MGBM criterion with different values of R and the amount of iterations used by Deb et. al. [6] and Khare [9] when solving DTLZ3 and DTLZ7 with similar configurations of NSGA–II and SPEA2.



(a) DTLZ3 solved using NSGA-II.

(b) DTLZ3 solved using SPEA2.

Figure 3: Expected failure tests applying NSGA-II and SPEA2 to the DTLZ3 problem with 10 objectives. The plots show the evolution of the a posteriori progress estimation, $\hat{s}_t$, after each iteration.

the algorithm in order to establish the reason of the process halt. Evidences obtained during the production of this work that indicate that, by analyzing amount of dominated and non–dominated individuals in the population, some conclusions can be extracted. If a large part (or in the worst case, all) of the population is non–dominated then the algorithm might had stopped because it was unable to reach satisfactory solutions. On the other hand, if there is a "healthy" relation between dominated and non–dominated individuals it is probable that the algorithm stopped with success.

Acknowledgements

This work is funded by projects CICYT TSI2005-07344, AUTOPIA (IMSERSO) and CAM MADRINET S-0505/TIC/0255.

7. REFERENCES

[1] C. A. Coello Coello. An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys*, 32(2):109–143, June 2000.

[2] C. A. Coello Coello. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter Recent Trends in Evolutionary Multiobjective Optimization, pages 7–32. Advanced Information and Knowledge Processing. Springer Verlag, 2004.

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

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter Scalable Test Problems for Evolutionary Multiobjective Optimization, pages 105–145. Advanced Information and Knowledge Processing. Springer Verlag, 2004.

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

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

[9] V. Khare, X. Yao, and K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 376–390, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[10] P. S. Maybeck. *Stochastic models, estimation, and*

control, volume 141 of *Mathematics in Science and Engineering*. Academic Press, 1979.

[11] O. Rudenko and 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), 2004.

[12] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1993.

[13] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[14] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[15] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[16] E. Zitzler and 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, May 1998.