

TOWARDS MEASURING EFFECTIVENESS IN DYNAMIC ENVIRONMENTS

Pavel Novoa-Hernández*, Eduardo Samaniego-Mena* and Jorge Murillo-Oviedo*
 Facultad de Ciencias de la Ingeniería,
 Universidad Técnica Estatal de Quevedo, Ecuador.

ABSTRACT

Many real-life scenarios can be modeled as Dynamic Optimization Problems (DOPs), which demand for finding optimal solution over time. From the viewpoint of metaheuristics methods, DOPs have been extensively addressed over the last two decades. One important issue in this context is how to assess the algorithm performance. Most of current proposals rely on single information from data, which limits the notion about the overall performance of the algorithm. So, in order to contribute to this issue, in this paper we propose a new performance measure for algorithm assessment in evolutionary dynamic optimization. We derived our proposal from what we considered as *effectiveness* in dynamic environments. Different from other existing measures, our proposal involve not only the accuracy, but also the time (efficiency) of the algorithm. In order to illustrate its usefulness and relationship with other literature measures an experimental analysis was conducted. Results show that the proposed measure can be suitable employed in typical experimentation scenarios and offers new information about the algorithms performance.

KEYWORDS: Evolutionary Dynamic Optimization, Performance measures, Effectiveness.

MSC: 68T20.

RESUMEN

Varios escenarios reales pueden ser modelizados como Problemas Dinámicos de Optimización (PDOs), los que exigen encontrar soluciones óptimas a lo largo del tiempo. Desde el punto de vista de métodos metaheurísticos, los PDOs han sido estudiados ampliamente durante las dos últimas décadas. En este contexto un aspecto importante es cómo evaluar el rendimiento del algoritmo. La mayoría de las propuestas actuales se basan en una información única a partir de los datos, lo cual limita la noción acerca del rendimiento global del algoritmo. Con el objetivo de contribuir a resolver esta problemática, en el presente artículo proponemos una nueva medida para evaluar el rendimiento de los algoritmos en ambientes dinámicos. Hemos derivado nuestra propuesta a partir de lo que consideramos como *efectividad* en ambientes dinámicos. Diferente a otras medidas, nuestra propuesta involucra no solo la precisión del algoritmo, sino también su eficiencia. Para ilustrar su utilidad y relación con otras medidas de la literatura hemos desarrollado un análisis experimental. Los resultados muestran que la medida propuesta puede ser empleada en escenarios típicos de experimentación y al mismo tiempo ofrece información diferente sobre el rendimiento del algoritmo.

1. INTRODUCTION

Evolutionary Dynamic Optimization (EDO) [12] is a popular research area in the field of Soft Computing [21]. Its main goal is to solve Dynamic Optimization Problems (DOPs) by evolutionary (metaheuristics) methods. Formally, a DOP is defined as:

$$DOP := \{\max_{x \in X} f^{(t)}(x)\} \quad (1)$$

where $X \subseteq \mathbb{R}^D$ is the set of feasible solutions (D -dimensional search space), and $f : \mathbb{R}^D \times \mathbb{N}_0 \rightarrow \mathbb{R}$ is the objective function to be maximized at every time step t .

Over the last two decades several approaches has been proposed for solving DOPs from the viewpoint of metaheuristics methods, as shown in [6, 7, 12]. In addition to the optimization task, in EDO the algorithm must implement special mechanisms for dealing with environment changes. Specially for change detection and for adapting the search when a new environment appears. Usually, for population-based algorithms (e.g. Evolutionary Algorithms and Swarm Intelligence Algorithms), adapting the search is achieved by maintaining the population diversity with aim of avoiding convergence issues.

*pnovoa@uteq.edu.ec

In this context, one important research topic in EDO is the design and selection of suitable performance measures for algorithm assessment. However, from a review of the related literature it is possible to observe that there is no measure for characterizing the algorithm in a wider perspective, that is, involving more than one performance criteria. For instance accuracy vs. efficiency, reactivity vs. stability [22]. Some examples of proposals involving single information measures are the offline error [4], the adaptability [20], the best error before the change [9], the collective fitness [11], the fitness-based statistics from [19], among others.

In order to contribute to this topic, the present work proposed a new performance measure for dynamic environments, which we have called *effectiveness*. Our proposal quantifies the accuracy and efficiency of the algorithm in a single magnitude. With aims of illustrating both, how *efficiency* can be used in dynamic environments and its relationship with other existing measures, we analyzed it through computational experiments. Results showed that the proposed measure can be suitable for typical experimentation scenarios and it offers new information about the algorithm performance.

The rest of the paper is organized as follows: in Section 2. we reviewed the previous works. Section 3. describes our proposal, which is analyzed in Section 4.. Finally we outlined the conclusion and future works in Section 5.

2. PERFORMANCE MEASURES IN DYNAMIC ENVIRONMENTS

As pointed out by [12, 22, 23] the selection of a performance measure is strongly related to what the researcher considered as the goal of the algorithm in dynamic environments. For instance, from the definition of DOPs given in Eq. (1) the reader can easily infer that the goal of the algorithm is to find the best solution, in terms of the objective function (fitness), at every time step. However, other goals are also possible leading to defining different measures.

For instance, one of the first performance measure in dynamic environments was *weighted accuracy* from [10]. This measure aims for characterizing the algorithm accuracy depending on the generation g (iteration):

$$perfAcc = \frac{1}{G} \sum_{g=1}^G \alpha_g \cdot accuracy^{(g)} \quad (2)$$

where $\alpha^{(g)} \in \mathbb{R}$ is the weight for generation g with $g = 1, \dots, G$, and *accuracy* is defined as:

$$accuracy^{(g)} = \frac{\hat{f}^{(g)} - f_{min}}{f_{max} - f_{min}} \quad (3)$$

where \hat{f} is the fitness (objective function value) corresponding to the best solution found by the algorithm in generation g . Besides, f_{max} y f_{min} are the best and the worst fitness in the search space for the generation g . As the reader can observe this measure quantify rely on the assumption that the algorithm accuracy in certain generations are more important than in others. For instance, first generations after a change vs. generations before a new change.

On the other hand, the *offline error* and the offline performance from [4], are among the most used measures in dynamic environments. These measures focus in the error and fitness of the best solution of the algorithm, during the execution:

$$offlineError = \frac{1}{T} \sum_{t=1}^T bestError^{(t)} \quad (4)$$

$$offlinePerf = \frac{1}{T} \sum_{t=1}^T \hat{f}^{(t)} \quad (5)$$

where $t = 1, \dots, T$ are function evaluations. In practice, each function evaluation is considered as a time unit in EDO. Besides, *bestError*^(t) is defined as:

$$bestError^{(t)} = |f_{optim}^{(t)} - \hat{f}^{(t)}| \quad (6)$$

Other fitness-based measures are *adaptability* and *stage accuracy* [20]. They are defined as:

$$adaptability = \frac{1}{K} \sum_{k=1}^K \left(\frac{1}{G_k} \sum_{g=1}^{G_k} bestError^{(g)} \right) \quad (7)$$

$$stageAccuracy = \frac{1}{K} \sum_{k=1}^K |f_{optim}^{(G_k)} - \hat{f}^{(G_k)}| \quad (8)$$

$$(9)$$

where K is the number of changes in the problem, while G_k is the number of generations within a stage k (environment).

In [22] the author proposed two interesting measures: *stability* and *reactivity*. These measure were defined bearing in mind different goals for an algorithm in dynamic environments:

$$stability^{(G)} = \max\{0, acc^{(g-1)} - acc^{(g)}\} \quad (10)$$

$$\varepsilon - reactivity^{(G)} = \min\{g' - g : g < g' \leq G, \frac{acc(g')}{acc(g)} \geq (1 - \varepsilon)\} \quad (11)$$

here, acc is an expression for computing the algorithm accuracy (e.g. Eq. 3). Besides, $\varepsilon \in \mathbb{R}_+$ y $g' \in \mathbb{N}$. Note that *stability* is defined in the range $[0, 1]$ if Eq. 3 is employed for acc . However, regardless the selected expression for acc , a value close to 0 of *stability* means that the algorithm is stable. On the other hand, $\varepsilon - reactivity$ is given by time units, that is, generations. The conceptual meaning of these the required time of the algorithm for achieving an accuracy close to the obtained in the previous stage (e.g. before a change). Particularly, a lower value indicates high algorithm reactivity.

Additionally, [22] explored extensions of *stability* and *reactivity*, by employing different expressions for acc . For instance, the author considered the *distance* between the best solution of the algorithm and the global optima of the problem. This measure is defined as:

$$bestDist^{(k)} = \sqrt{\sum_{d=1}^D (x_d^* - \hat{x}_d)^2} \quad (12)$$

where x_d^* and \hat{x}_d are the component d of the problem optima and algorithm best solution, respectively.

Another interesting proposal was given in [11], in which the author proposed a generation-based version of the offline performance [4]. The author called this measure *collective fitness* and he defined as:

$$collectiveFitness = \frac{1}{G} \sum_{g=1}^G \hat{f}^{(g)} \quad (13)$$

In some cases where is known that the problem is multimodal (several solution locally optima), then it would be interesting to analyze whether the algorithm is capable of locating these multiple solutions. Performance measures motivated from this goal are the proposed in [2, 5]. Specifically, the first one takes into account the error of locating the true global optima of the problem, while the second one says how many *peaks* (local optima solutions) are covered by the algorithm.

In recent years, some authors have proposed more sophisticated measures. This is the case of *fitness degradation* from [1], which applied linear regression to estimate the algorithm overall accuracy. Another sophisticated approach is adopted in [19], which relied on multiple hypothesis testing for time series.

As the reader may note, one common feature of performance measures in dynamic environments is that they rely on a single source of information. So, they characterize the algorithm from a limited viewpoint. The next section describe our proposal for overcome this issue.

3. PROPOSED MEASURE

The new measure we propose is motivated by the fact that in some real-world scenarios (even for non-dynamic environments), the final user may demand the set of best possible solutions at the minimum waste of resources

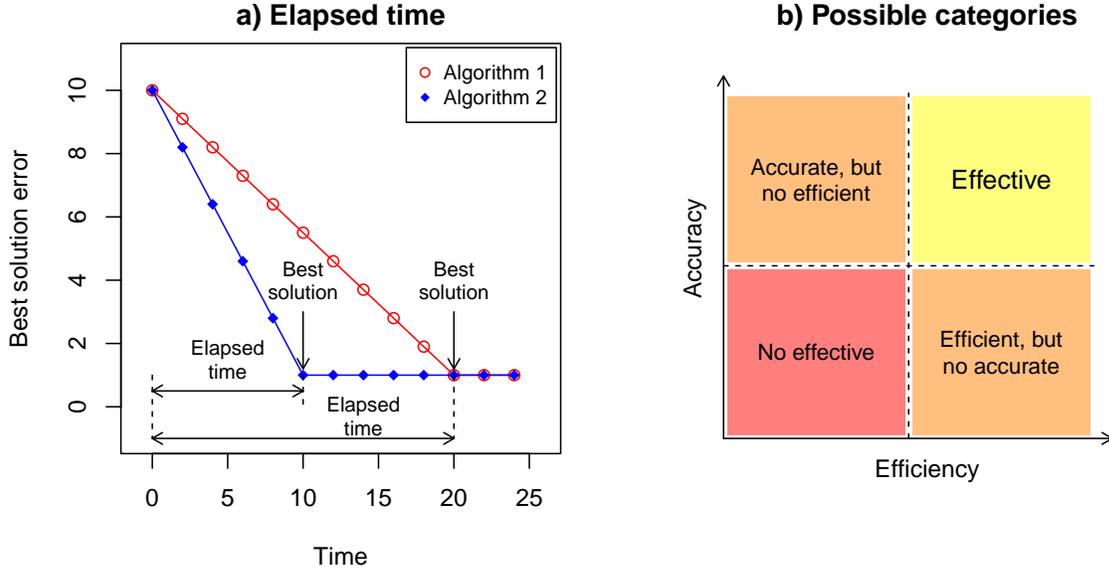


Figure 1: Illustrative example of *elapsed time* in two algorithms with different performance (a), and the four possible categories from the concepts of *efficiency*, *accuracy* and *effectiveness* (b).

(e.g. time). As one may note, this requirement is not new in the context of evolutionary optimization. In fact, its accomplishment in complex environments is the main cause of the success of metaheuristics over classical (exact) optimization methods.

What we considered as “best solution” is, of course, a relative concept that depends on the user. Mathematically it can be modeled by assuming that it depends on a satisfaction degree ε , over of which the obtained solution satisfies the user. Regardless the selected criteria, it is clear that the consumed time in achieving the desired solution is an important criteria for evaluating the algorithm *efficiency*. Of course, other computational resources are also important, like RAM memory or CPU time. However, defining efficiency in dynamic environments is difficult since the algorithm is running regardless the solution quality. In other words, the execution time is theoretically infinite [18]. One possible solution for dealing with this dilemma is to defined efficiency in terms of the time wasted by the algorithm in order to obtain the “best solution” in every environment, but from the viewpoint of the algorithm, that is, regardless the user. With aim of illustrate this idea, Fig. 1-a) shows the behavior, in terms of the best solution error over time, of two different algorithms in a single environment. Note that both algorithms obtain solutions with identical quality, however, *Algorithm 1* is more efficient than *Algorithm 2*, since the former converges more quickly than the second one.

More formally, we can define efficiency as follows:

Definition 1 (Efficiency). *Be the environment k defined in the time interval $[T_k^{min}, T_k^{max}]$, and $\hat{\mathbf{x}}_k$ the best solution found by the algorithm A in k , then the **efficiency** of A in the environment k is:*

$$efficiency_A^{(k)} = 1 - \frac{\hat{t}}{T_k^{max} - T_k^{min}} \quad (14)$$

where \hat{t} is the elapsed time by A such that: $\hat{t} = \min \{t : \hat{\mathbf{x}}^{(t)} = \hat{\mathbf{x}}_k\}$.

As one may note the main drawback of defining efficiency like above, is the impossibility of known how accurate the algorithm is. Fortunately, we have several options. For instance, the absolute fitness (error) of the best solution [16, 17], the distance to the true optimum of the problem [22], among others. In general, we say that one of these measures assesses the algorithm *accuracy*.

So, in our opinion the performance of one algorithm in dynamic environments depends on these two criteria *efficiency* and *accuracy*. Intuitively, if the algorithm is efficient and accurate, then we say that it is *effective*. Fig. 1-b) shows the four possible categories for an algorithm from the concepts of efficiency, accuracy and

effectiveness. Note for example that algorithms with low levels of accuracy and efficiency are termed as *no effective*, while *effective* on the contrary.

Assuming that both, efficiency and accuracy are measures in the range $[0, 1]$, then we can define *effectiveness* as:

Definition 2 (Effectiveness). *Be $efficiency_A^{(k)}$ and $accuracy_A^{(k)}$ the efficiency and accuracy of the algorithm A in the environment k , respectively, then the **effectiveness** of A in k is:*

$$effectiveness_A^{(k)} = 1 - \frac{\|\mathbf{v}_A - \mathbf{v}^*\|}{\|\mathbf{v}^*\|} \quad (15)$$

where $\|\cdot\|$ is the euclidean norm, $\mathbf{v}_A = (efficiency_A^{(k)}, accuracy_A^{(k)})^T$, and $\mathbf{v}^* = (1, 1)^T$.

The meaning of these measure is quite simple. For instance, note that vector \mathbf{v}^* is composed by ideal values for the efficiency and accuracy. So, it also represents the ideal *effectiveness*. On the other hand, the norm $\|\mathbf{v}_A - \mathbf{v}^*\|$ says how different is the algorithm effectiveness regarding the ideal value, which in turn is normalized by $\|\mathbf{v}^*\|$. Finally, by subtracting this normalized value from one, we give an intuitive meaning to the measure. For example, a value closer to 0 means a low effectiveness, while a value closer to 1 the opposite.

Nevertheless, like occurring for other measures, *effectiveness* presents some difficulties:

1. the selected measure for *accuracy* requires to be normalized, so in the case we do not know which are the best and the worst values of the measure, computing that normalization could be difficult, and
2. both criteria (e.g. *efficiency* and *accuracy*) have the same importance, so we can not use Def. (2) for expressing the case of different importance degrees for them.

The first issue is much difficult to solve than the second one. One possible solution for this is to approach it as a Pareto efficiency problem, that is, by applying the concept of *dominance*. In this way we have not rely on a normalize value for accuracy. Alternatively, we can also normalize the accuracy by using the results obtained by the algorithms involved in the experiments, that is, by employing the worst and the best accuracy values found by the algorithms. Although this is an easy solution for the normalization issue, it is worth noting that the experiment results obtained by this approach cannot be used across different studies. So, this issue will be the subject of our future research.

Regarding to the second difficulty, an intuitive solution is to use weights for the criteria. For instance, we can weight the component-wise difference of the vector $\mathbf{v}_A - \mathbf{v}^*$. Formally we have:

$$\|\mathbf{v}_A - \mathbf{v}^*\|_{\mathbf{w}} = \|\mathbf{w}^T \cdot (\mathbf{v}_A - \mathbf{v}^*)\| \quad (16)$$

where $\mathbf{w} = (\omega_e, \omega_a)$ is the weight vector corresponding to the efficiency and accuracy, respectively. As a consequence, we obtain a more general definition for *effectiveness*. In what follows, we refer it as *weighted effectiveness*.

4. EXPERIMENTAL ANALYSIS

In order to analyze the proposed measure, we have designed two experiments. The first one is devoted to illustrate how the efficacy can be employed in typical scenarios, while the second one is oriented to show the statistical relationship between the proposed measure and others from literature. To this end we have considered four algorithms and 36 problem instances from the Moving Peaks Benchmark [4]. Table 1 shows the default parameter setting for MPB's scenario 2, which was employed in the experiments. The MPB instances were derived from the levels combination of the following factors:

- Peak function (f_p):
 $\{Sphere, Schwefel, Rastrigin, Ackley\}$
- Change frequency (Δe):
 $\{1000, 5000, 10000\}$.

Parameter	Setting
Dimension (D)	5
Search space (Ω, Γ)	$[0, 100]^5$
Number of peaks	10
Peak heights (H_i)	$\in [30, 70]$
Peak widths (W_i)	$\in [1, 12]$
Peak function (f_p)	$f_{cone}(X) = \sqrt{\sum_{d=1}^D X_d^2}$
Shift severity (sev)	1.0
Change frequency (Δe)	5000
Correlation coefficient (λ)	1.0

Table 1: Default parameter setting for Scenario 2 from the Moving Peak Benchmark.

Parameter	mQSO	mQSOE	mPSOD	mPSODE
Number of populations	10	10	10	10
Number of conventional individuals	5	5	5	5
Number of quantum individuals	5	5	5	5
Cloud / diversity radii (r_{cloud} / r_{div})	1.0	1.0	1.0	1.0
Exclusion radii (r_{excl})	30.0	30.0	30.0	30.0
Diversity strategy	Quantum individuals.	Quantum individuals. + Control rule for converged populations.	Diversity after the change.	Diversity after the change + Control rule for converged populations.

Table 2: Parameter settings for the algorithms employed in the experiments.

- Shift severity (sev):
 $\{1.0, 5.0, 10.0\}$.

Regarding to the algorithms, we selected: the mQSO from [3], and those proposed in [13], that is, mPSOD, mPSODE, and mQSOE. All the algorithms include several populations and detect the environment changes by reevaluating the current best solution. Their main differences are the optimization paradigm, and the diversity strategy for coping with the environment changes. More details can be found in the Table 2, and in the references given below.

In general, we performed 20 runs for each pair problem-algorithm and each problem instance changed 50 times. Regarding to the *accuracy* definition, we considered the fitness of the best solution of the algorithm, which is in turn normalized by the difference between the worst and the best fitness values taking into account the three algorithms. Finally, it is important to remark that for every run we have averaged the values of effectiveness over 50 changes, leading to a single value of this measure.

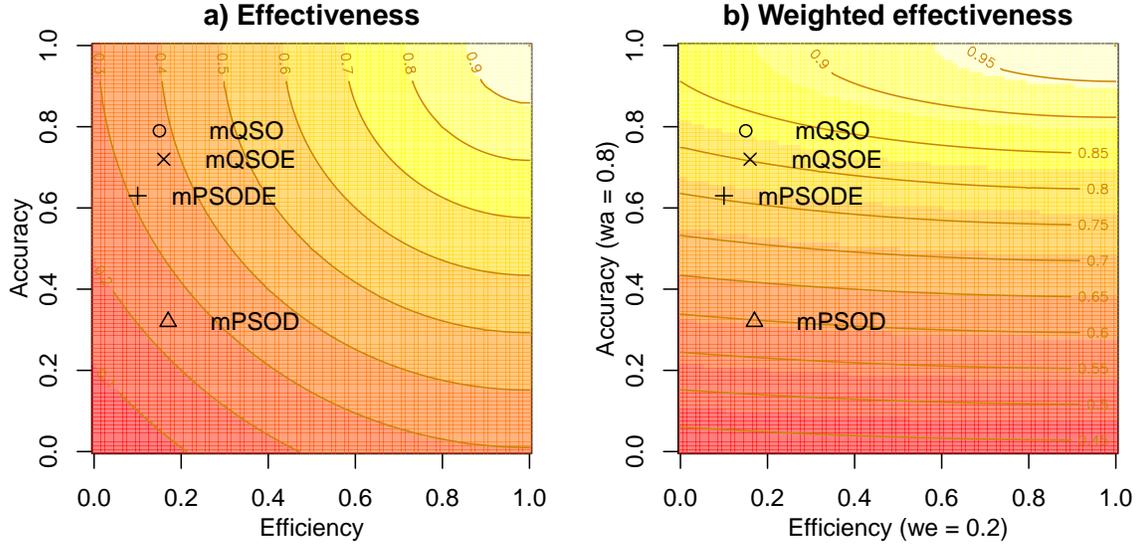


Figure 2: Contour maps of the *effectiveness* (a) and *weighted effectiveness* (b) for the algorithms mQSO, mPSOD, mPSODE and mQSOE. The results are obtained from the average of the effectiveness in all problem instances.

4.1. Using the effectiveness measure

To see how the four algorithms can be evaluated using the effectiveness measure, in Fig. 2-a) we plotted through contour maps, their corresponding effectiveness levels, that is, according to their levels of efficiency and accuracy. Particularly, Fig. 2-b) shows a weighted variant of the effectiveness. In this case, our intention was to illustrate the case when the user consider that the accuracy is more important than the efficiency. Note that since the employed weights are different, then the effectiveness function is clearly affected (see the contours of Fig. 2 -b).

The first conclusion that we can derive from the results of Fig. 2 is that all algorithms have a low level of efficiency, which means that they obtain the best solution too close of the occurrence of a change in the environment. Regarding the accuracy, algorithms mQSO, mQSOE and mPSODE are better than mPSOD. Finally, if we consider the effectiveness, algorithms mQSO and mQSOE are the best ones in Fig. 2-a), while the mQSO is the best in the case of the weighted case Fig. 2-b).

4.2. Relationship with other measures

On the other hand, with aims of discovering the relationship between our measures and others from literature, we proceed with a correlation analysis. In that sense, we employed the Pearson's coefficient for expressing not only the strength, but also the direction of the correlation. The measures considered from literature were: the distance to the optimum before the change [22], the best error before the change [8, 15], the stability [22], reaction [22], and the offline error [4]. Besides, we consider interesting to include the *efficiency* measure (Eq. 14) in our correlation analysis. See Table 3.

In Fig. 3 we have drawn the Pearson's correlation coefficients for the two variants of effectiveness. Note that we have distinguished through different colors the significant and no significant correlation. As we can see in Fig. 3-a) the effectiveness positively correlates with efficiency and its weighted variant. Of course, the correlation is stronger in the latter case. Additionally it negatively correlates with the best error before the change and reactivity, being its correlation with the rest no significant (at least for the employed data). On the other hand, it is worth noting that the weighted effectiveness (Fig. 3-b) has a quite strong correlation with respect to the distance to the optimum. It is obvious that the employed weights have an important impact in the measure. However, compared with the others, our measures have not correlation coefficients greater(resp. lower) than 0.5(resp. -0.5). This indicates that our proposals offer new information for assessing the algorithm

Performance Measure	Formula
Distance to the optimum before the change [22]	$avgBestDist^{(K)} = \frac{1}{K} \sum_{k=1}^K bestDist^{(k)}$
Best error before the change [8]	$avgBestErrorBef^{(K)} = \frac{1}{K} \sum_{k=1}^K bestError^{(k)}$
Stability [22]	$avgStability^{(K)} = \frac{1}{K} \sum_{k=1}^K stability^{(k)}$
Reactivity ε – reactivity [22]	$avgReactivity_{\varepsilon}^{(K)} = \frac{1}{K} \sum_{k=1}^K reactivity^{(k)}$
Offline error [4]	$offlineError^{(K)} = \frac{1}{T} \sum_{t=1}^T bestError^{(t)}$
Efficiency	$avgEfficiency^{(K)} = \frac{1}{K} \sum_{k=1}^K efficiency^{(k)}$
Effectiveness	$avgEffectiveness^{(K)} = \frac{1}{K} \sum_{k=1}^K effectiveness^{(k)}$

Table 3: Selected performance measures for experimental analysis.

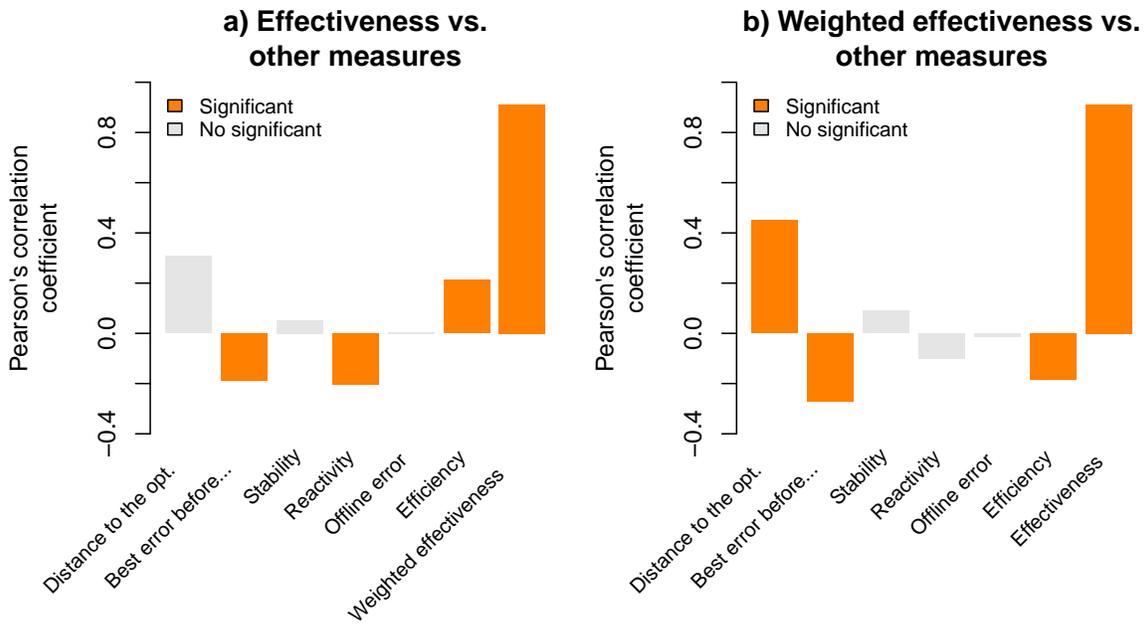


Figure 3: Pearson's correlation coefficient ($\alpha = 0.05$) for the effectiveness (a) and weighted effectiveness (b) vs. other measures.

performance.

5. CONCLUSION AND FUTURE WORKS

In this paper we proposed a new measure for assessing algorithm effectiveness in dynamic environments. Our proposal is based in the concepts of efficiency and accuracy, which are two important criteria in optimization. From the conducted experimental analysis it is possible to conclude that the proposed measure results informative enough for algorithm comparison, and also it offers a new perspective to assess algorithms in dynamic environments. However, we found as major drawback the normalization of the accuracy, since it depends on the best and worst value of the problem.

Our future research will be oriented to: the exploration of other accuracy criteria and the examination of the discrimination power of the proposed measure when statistical tests are used for algorithm comparison. Besides, we plan to include our performance measures in DynOptLab [14], a software for experimentation in dynamic

environments, and to explore discrete optimization problems.

RECEIVED: *****
REVISED: *****

REFERENCIAS

- [1] ALBA, E. AND SARASOLA, B. [2010]: Measuring fitness degradation in dynamic optimization problems In **EvoApplications 2010**, volume I, pages 572–581. Springer-Verlag Berlin Heidelberg.
- [2] BIRD, S. AND LI, X. [2007]: Informative performance metrics for dynamic optimization problems In **9th Conf. on Genetic and Evolutionary Computation**, pages 18–25.
- [3] BLACKWELL, T. M. AND BRANKE, J. [2006]: Multiswarms, exclusion, and anti-convergence in dynamic environments **IEEE Transactions on Evolutionary Computation**, 10, 4,:459–472.
- [4] BRANKE, J. [1999]: Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzal, A., editors, **Proceedings of the Congress on Evolutionary Computation**, volume 3, pages 1875–1882, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- [5] BRANKE, J. [2002]: **Evolutionary Optimization in Dynamic Environments**, volume 3 Springer US, Boston, MA.
- [6] CRUZ, C., GONZÁLEZ, J. R., AND PELTA, D. [2011]: Optimization in dynamic environments: a survey on problems, methods and measures **Soft Computing**, 15, 7,:1427–1448.
- [7] JIN, Y. AND BRANKE, J. [2005]: Evolutionary optimization in uncertain environments—a survey **Evolutionary Computation, IEEE Transactions on**, 9, 3,:303–317.
- [8] LI, C. AND YANG, S. [2008]: A Generalized Approach to Construct Benchmark Problems for Dynamic Optimization In **Simulated Evolution and Learning**, volume 5361 of **Lecture Notes in Computer Science**, pages 391–400. Springer Berlin / Heidelberg.
- [9] Li, C., Yang, S., Nguyen, T. T., Yu, E. L., Yao, X., Jin, Y., Beyer, H.-G., and Suganthan, P. N. [2008]: Benchmark Generator for CEC’2009 Competition on Dynamic Optimization Technical report, Department of Computer Science, University of Leicester, U.K.
- [10] MORI, N., KITA, H., AND NISHIKAWA, Y. [1996]: Adaptation to a changing environment by means of the thermodynamical genetic algorithm In Voigt, H.-M., editor, **Parallel Problem Solving from Nature**, number 1141 in LNCS, pages 513–522.
- [11] MORRISON, R. [2003]: Performance Measurement in Dynamic Environments In Branke, J., editor, **GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems**.
- [12] NGUYEN, T. T., YANG, S., AND BRANKE, J. [2012]: Evolutionary dynamic optimization: A survey of the state of the art **Swarm and Evolutionary Computation**, 6, 0,:1–24.
- [13] NOVOA-HERNÁNDEZ, P., CORONA, C., AND PELTA, D. [2011]: Efficient multi-swarm PSO algorithms for dynamic environments **Memetic Computing**, 3, 3,:163–174.
- [14] NOVOA-HERNÁNDEZ, P., CRUZ CORONA, C., AND PELTA, D. [2015]: A Software Tool for Assisting Experimentation in Dynamic Environments **Applied Computational Intelligence and Soft Computing**, 2015, Article ID 302172,:12p.
- [15] NOVOA-HERNÁNDEZ, P., CRUZ CORONA, C., AND PELTA, D. [2016]: Self-adaptation in dynamic environments - a survey and open issues **International Journal of Bio-inspired Computation**, 8, 1,:1–13.
- [16] NOVOA-HERNÁNDEZ, P., CRUZ CORONA, C., AND PELTA, D. A. [2013]: Self-adaptive, multipopulation differential evolution in dynamic environments **Soft Computing**, 17, 10,:1861–1881.

- [17] NOVOA-HERNÁNDEZ, P., PELTA, D., AND CORONA, C. [2010]: Improvement Strategies for Multi-swarm PSO in Dynamic Environments In González, J., Pelta, D., Cruz, C., Terrazas, G., and Krasnogor, N., editors, **Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)**, volume 284 of **Studies in Computational Intelligence**, pages 371–383. Springer Berlin / Heidelberg.
- [18] SCHÖNEMANN, L. [2007]: Evolution Strategies in Dynamic Environments In Yang, S., Ong, Y.-S., and Jin, Y., editors, **Evolutionary Computation in Dynamic and Uncertain Environments**, volume 51 of **Studies in Computational Intelligence**, pages 51–77. Springer Berlin Heidelberg.
- [19] SHILANE, D., MARTIKAINEN, J., AND OVASKA, S. J. [2009]: Time-dependent performance comparison of evolutionary algorithms In et al., M. K., editor, **ICANNGA 2009**, number 5495 in **Lecture Notes On Computer Science**, pages 223–232. Springer-Verlag Berlin Heidelberg.
- [20] TROJANOWSKI, K. AND MICHALEWICZ, Z. [1999]: Searching for optima in non-stationary environments In **Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999)**, volume 3, page (2348).
- [21] VERDEGAY, J. L., YAGER, R. R., AND BONISSONE, P. P. [2008]: On heuristics as a fundamental constituent of soft computing **Fuzzy Sets Syst.**, 159, 7,:846–855.
- [22] WEICKER, K. [2002]: Performance Measures for Dynamic Environments In Guervós, J. J. M., Adamidis, P., Beyer, H.-G., Schwefel, H.-P., and Fernández-Villacañas, J.-L., editors, **Parallel Problem Solving from Nature PPSN VII**, volume 2439 of **Lecture Notes in Computer Science**, pages 64–73. Springer Berlin Heidelberg.
- [23] Weicker, K. [2003]: **Evolutionary Algorithms and Dynamic Optimization Problems** PhD thesis, Institut für Formale Methoden der Informatik der Universität Stuttgart.