

Are Evolutionary Algorithm Competitions Characterizing Landscapes Appropriately?

Pilar Caamaño
Grupo Integrado de Ingeniería
Escuela Politécnica Superior
Universidade da Coruña,
Spain
pcsobrin@udc.es

Jose A. Becerra
Grupo Integrado de Ingeniería
Escuela Politécnica Superior
Universidade da Coruña,
Spain
ronin@udc.es

Francisco Bellas
Grupo Integrado de Ingeniería
Escuela Politécnica Superior
Universidade da Coruña,
Spain
fran@udc.es

Richard J. Duro
Grupo Integrado de Ingeniería
Escuela Politécnica Superior
Universidade da Coruña,
Spain
richard@udc.es

ABSTRACT

Currently, researchers in the field of Evolutionary Algorithms (EAs) are very interested in competitions where new algorithm implementations are evaluated and compared. Usually, EA users perform their algorithm selection by following the results published in these competitions, which are typically focused on average performance measures over benchmark sets. These sets are very complete but the functions within them are usually classified into binary classes according to their separability and modality. Here we show that this binary classification could produce misleading conclusions about the performance of the EAs and, consequently, it is necessary to consider finer grained features so that better conclusions can be obtained about what scenarios are adequate or inappropriate for an EA. In particular, new elements are presented to study separability and modality in more detail than is usually done in the literature. The need for such detail in order to understand why things happen the way they do is made evident over three different EAs.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Searches—*Heuristic methods*

General Terms

Algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '11, July 12–16, 2011, Dublin, Ireland.

Copyright 2011 ACM 978-1-4503-0690-4/11/07 ...\$10.00.

Keywords

Evolutionary computation, fitness landscapes, modality analysis, separability analysis

1. INTRODUCTION

Evolutionary Algorithm (EA) competitions take place, typically, during EA workshops and conferences, such as GECCO (EvoDOP-2007) or CEC [16, 9, 17]. Researchers from different disciplines use them as a reference to select an EA by following the results published in these competitions, which are typically focused on average performance measures over a benchmark set. Once the “winner” EA has been selected, the most common procedure users follow is to tune its configuration and apply it trying to solve their problem until a successful solution is obtained [19, 12]. If this initial selection fails, the next step is to attempt to address the problem with a different EA after consulting the literature of the topic. This is a highly time-consuming and frustrating trial and error process as these selections could prove to be completely wrong if the most acclaimed algorithms fail in a particular feature required for solving the specific problem the user faces. In order to try to minimize this problem there is no doubt that it is necessary to increase our knowledge about how an EA behaves. One way of doing this is by increasing the detail of the analysis of the benchmark functions used to analyze EA response.

Although in EA competitions the comparisons of the algorithms are performed fairly by specifying a common stopping criterion, problem size, initialization scheme, etc. and even though the benchmark sets are very complete, we have detected shortcomings in the way the functions in these sets are characterized. Dimensionality, modality and separability are the basic features that are taken into account but, typically, very superficial classifications are considered (separable/non-separable, unimodal/multimodal, etc.) and, consequently, it is often not clear why an EA performs better over some functions and worse over others leading to researchers focusing just on how many functions are solved and how fast the algorithms are. The success and the performance of an EA over a function are closely related to these

three features through the way in which its search strategy is adapted to the morphology of the fitness landscape. Thus, a finer grained analysis of the landscape features will allow the designer to obtain more reliable and usable conclusions about the behavior of their EAs.

This paper proposes a set of additional considerations related to separability and modality in order to achieve this finer grained view of benchmark functions and EA behavior. To determine which ones are the most appropriate and test their practical relevance, we have chosen an extensive benchmark function set of real-parameter optimization problems [5]. These functions were classified according their separability and modality in a binary way in EA competitions. The benchmark set contains 36 scalable and non-scalable functions that permit studying the differences between the fitness landscapes in depth so as to allow linking these differences to EA performance.

The remainder of the paper is structured as follows: Section 2 describes the experimental setup used in this work to analyze the relevance of the proposed landscape features. Sections 3 and 4 are devoted to the description of the specific landscape features in terms of separability and modality, and with the presentation of empirical results that confirm their relevance over three well-known EAs. Finally, the main conclusions of this work are presented in Section 5.

2. EXPERIMENTAL SETUP

To illustrate the practical relevance of the landscape features proposed in this work, three EAs were chosen. Two of them are the winners of most EA competitions in the last few years: Differential Evolution (DE) [15] and the Covariance Matrix Adaptation - Evolutionary Strategy (CMA-ES) [3]. Additionally, a Real-Coded Genetic Algorithm (RCGA) [8] is chosen as a typical EA reference. The particular configuration parameters of each algorithm are the ones recommended by their authors in [14], [3] and [8]. For each algorithm-function-dimension combination, 25 independent runs were executed. The scalable function set was analyzed considering 10, 30 and 50 dimensions. The stopping criterion of each run is based on the maximum number of function evaluations (FEs) and was set to $10000 \cdot n$, where n is the dimension of the problem.

To study the results provided by the algorithms we have proposed the *Combined Error and Performance Measure* (CPEM) [4]. It provides combined results on the error level and the performance of the algorithms at a glance. The CPEM is calculated as follows:

$$CPEM = \begin{cases} \varepsilon \cdot \frac{FEs}{FEs_{max}} & \text{if } FEs \leq FEs_{max} \\ \text{absolute error} & \text{otherwise} \end{cases} \quad (1)$$

Where, ε is the threshold to consider that a function is solved (the global optimum is found), in this work it has been set to 10^{-6} . All the results of tables 2 and 4 are provided in terms of average CPEM over the 25 independent runs. In these tables the lowest CPEM the best result.

3. SEPARABILITY

Separability refers to the dependencies among the parameters of a function and it can be related with the biological concept of epistasis [13]. According to separability, the competitions' benchmark sets are usually divided into Sepa-

rable and Non-Separable functions. Separable functions [11] are those where all the variables are independent and, as a consequence, the optimization process of an n -dimensional Separable function ($f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$) could be divided into n 1-D optimization processes over each parameter x_i , where $0 \leq i < n$. Non-Separable functions [3] are those where dependencies are present and all the variables should be optimized during the same process because of the relationships between them.

Two facts lead us to analyze the function separability in depth. On one hand, after analyzing the results provided by the three EAs considered here using the typical binary classification of the benchmark functions into Separable and Non-separable, the situation is very confusing. For instance, Fig. 1(a) shows the results provided by the RCGA over the whole benchmark set (extracted from Tables 2 and 4). This figure displays the average CPEM error obtained for each function organized into 4 classes: functions with less than 10 dimensions, functions with 10, 30 and 50 dimensions. Square points correspond to Non-Separable landscapes while plus symbols correspond to Separable ones. The horizontal line at 10^{-6} corresponds to the threshold used to consider that a run has been solved. As shown, it is difficult to decide on the behavior of the RCGA over the Non-Separable function set due to the fact that solved and unsolved cases appear for all dimensions (separable functions are all solved but 3). On the other hand, sometimes it is difficult to classify a function in terms of separability because there is no consensus in the literature on how to do it. For example, Ackley's function [1] is sometimes classified as Separable [18] and in some other instances as Non-Separable [2].

Thus, after analyzing the results of the RCGA over *Non-Separable* functions, we realized that there are two different types of functions within this subset. It includes functions that are separable, but not linearly, and other functions that are Non-separable. Consequently, we propose a different classification in terms of separability that organizes functions into linearly separable functions (L-Separable), non-linearly separable functions (NL-Separable) [20] and Non-Separable functions.

To characterize an objective function in terms of these categories is not easy. Here we have made use of an empirical method although others are possible. The basic idea is to fix the values of $n-1$ parameters of an n parameter function and iterate the remaining parameter between the low and the high bounds for different values of the fixed parameters. To better understand how the algorithm works, Fig. 2 displays 2-D plots of the three types of separability considered here. Thus, if a function is linearly separable (*L-separable*), for different values of these $n-1$ parameters, it preserves the same graph shape shifted in the y-axis (see Fig. 2(a)). Following this reasoning, non-linearly separable functions (*NL-separable* hereafter) are those functions with different shapes (different search space) but with the optimum always in the same point (see Fig. 2(b)); and *non-separable* functions are those with different shape and different optimum (see Fig. 2(c)).

This method has allowed us to classify the functions of the benchmark set into the aforementioned three categories. After the name of each function in tables 2 and 4 we have added an L, NL or N to indicate if the function is L-Separable, NL-Separable and Non-Separable, respectively. The case of the Griewank function must be highlighted. Its separability

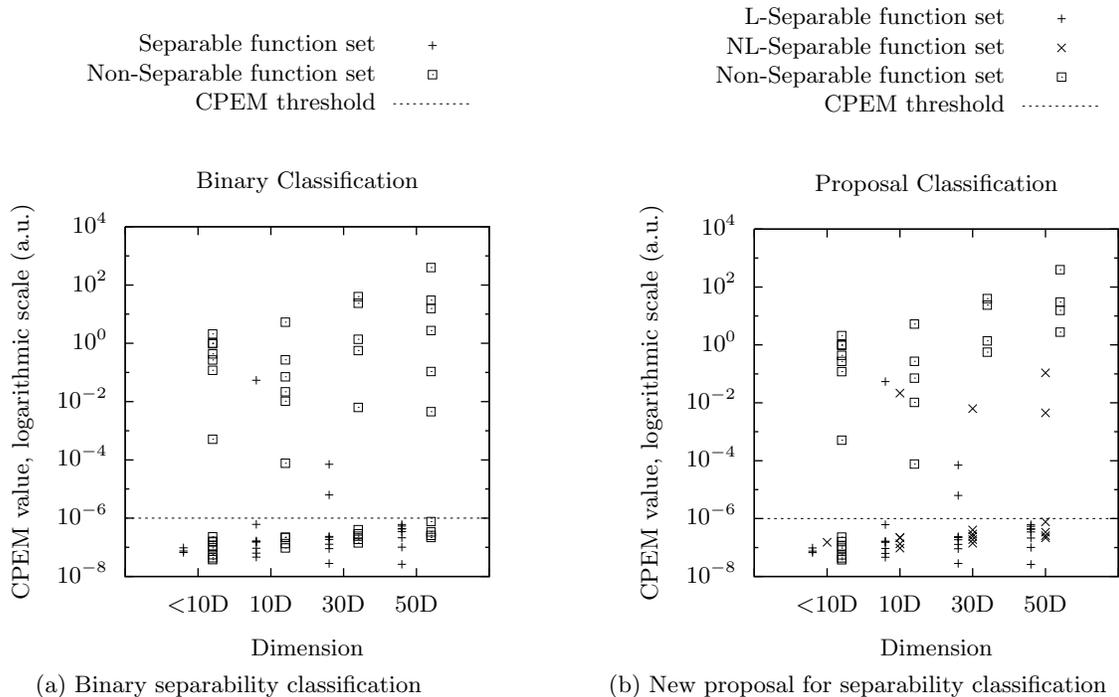


Figure 1: Two different proposals for separability classification.

changes as the dimensionality of the problem increases [10], going from Non-Separable when we deal with 10 dimensions to NL-Separable in the other two cases.

The behavior and success of an EA is closely related to the type of separability of the functions it acts on. Thus, the search process to optimize an L-Separable function could be carried out as independent processes over each parameter, in other words, one parameter at a time. That is also the case for NL-Separable functions where the parameters present non-linear dependencies among them, but the optimal value of each one may be obtained using independent optimization processes [20]. This does not happen for Non-Separable functions, where the goodness of a parameter depends on the value of some other parameters of the function. What must be highlighted here is that NL-separable and Non-separable functions, typically classified together, do not require the same type of search method.

Fig. 1(b) displays the response of the RCGA over the whole benchmark set again, but now classifying the functions into these three categories (crosses correspond to NL-separable functions). It can be observed that the algorithm performs successfully in most L and NL-Separable functions, failing only in 7 cases (and this is due to other issues related to modality). However, in the case of Non-separable functions its behavior degrades. Now it can be clearly seen that there are no squares in the region corresponding to solved functions for dimensions higher than 10. Consequently, we can now conclude that the RCGA has problems with Non-separable landscapes, that become worse as dimensionality increases. This is due to the fact that this RCGA implementation does not take into account the dependencies between parameters during the optimization process.

Using this classification of the benchmark functions, the behavior of EAs can now be analyzed with confidence. For

example, Fig. 3 displays the results obtained by the three algorithms considered here for the Non-separable functions of the benchmark set. It can be observed that the RCGA performs clearly worse than the other two. The CMA-ES performance [3] is successful due to the fact that it takes into account the dependencies among variables it performs (see results in Table 2 and 4). This algorithm approximates a covariance matrix of the function parameters representing the relationships between them. The Fig. shows that the behavior of DE in Non-Separable landscapes is very similar to that of CMA-ES, although CMA-ES shows better performance as it is less affected by dimensionality (DE fails 5 of 6 cases for dimensions higher than 10, as displayed in Fig. 3 looking at the square points). It is interesting to note that even though DE does not provide a mechanism to use the dependencies among variables in the generation of new individuals, its search process can be adjusted to deal with different types of separability. One of the parameters that regulates the behavior of DE, the cross-over rate (CR), permits determining how much of the genotype is changed each generation. Low values of CR cause few parameter changes, which favors solving L and NL-Separable functions. The opposite behavior takes place when CR is set to a high value, favoring the solution of Non-Separable functions. In this work the CR parameter is set to 0.1 for L and NL-Separable functions and to 0.9 for Non-Separable function, as the authors recommend in [14].

To sum up, separability is a highly relevant feature in the performance of EAs. It is closely related to the way the search process is carried out by the evolutionary operators in each EA. However, the classification of benchmark functions in terms of separability presented in many papers in the literature and used in all the EA competitions is sometimes confusing, mixing different types of functions, such

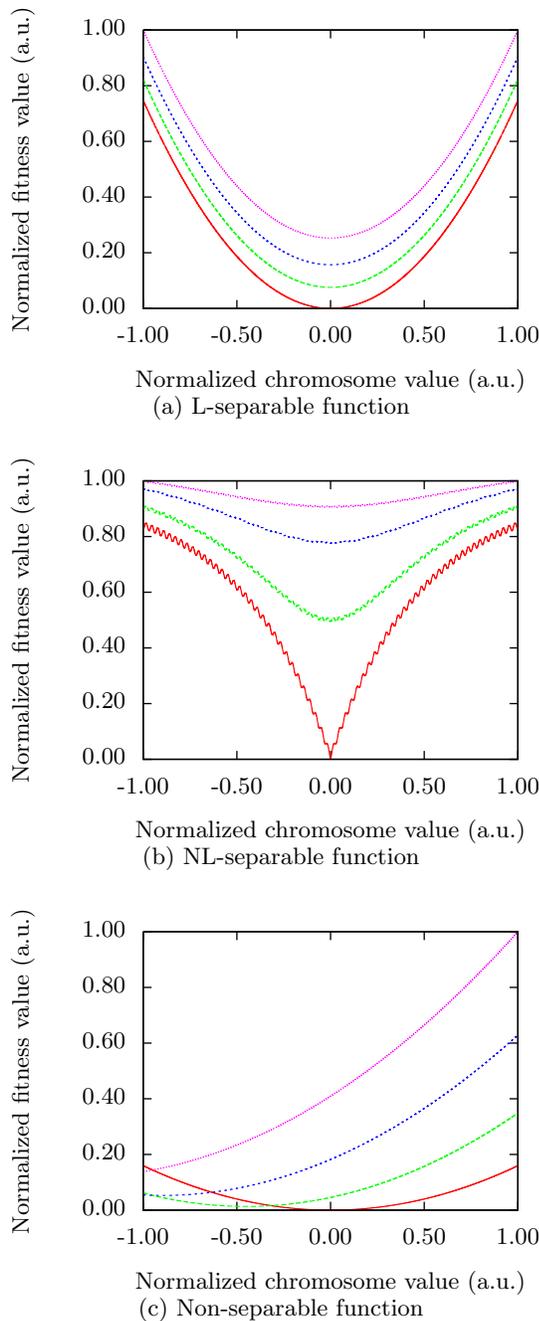


Figure 2: 2-D representation of the three types of separability considered

as NL-Separable and Non-Separable functions, in the same class when the search strategy required to solve them is very different. Here we propose using a finer grained classification of benchmark functions into three groups: L-separable, NL-separable and Non-separable in order to facilitate explaining the behavior of the different algorithms.

4. MODALITY

As commented in the introduction, the second landscape

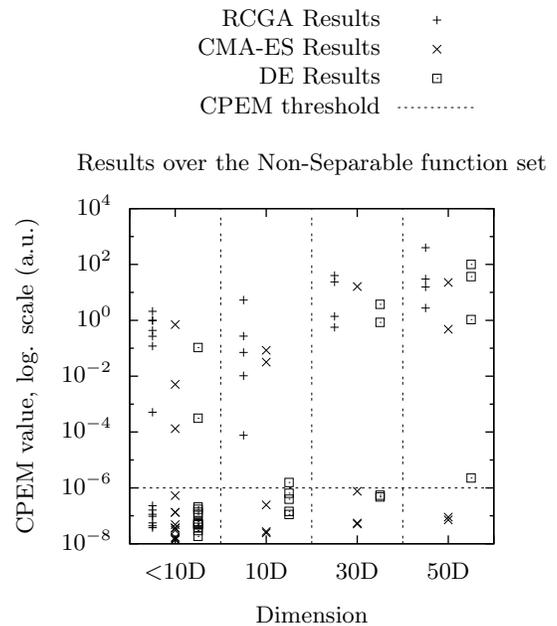


Figure 3: Results of the three algorithms over the Non-Separable function set.

feature typically considered in EA competitions is modality. Functions are usually divided into unimodal and multimodal and the behavior of the EAs is analyzed considering these two categories. A function is called *unimodal* if it displays a single global optimum and no local optima. On the other hand, *multimodal* functions present several local and / or global optima. Unimodal functions are, in general, easy to solve by a population based algorithm, and thus are mainly used to test the convergence speed of the algorithms.

Although from the previous statement it may seem that every unimodal function presents the same difficulty to every EA, here we will show that this assertion must be revised. In fact, it is well-known that needle-in-a-haystack (NiAH) [7] or a long path problem represent a challenge to EAs. Thus, it was necessary to establish what landscape feature could be used to account for the differences of behaviors of EAs over different unimodal landscapes. It turned out to be the longest path to the optimum, which is a property that is easy to measure for the different functions of the benchmark set used here and constitutes a difficulty indicator. To compute it, a local search method was used: the length of the path from the corners of the landscape to the optimum value were measured as the mean number of steps after 25 runs starting at each corner ($x_i = (x_i^1, \dots, x_i^n), x_i^j \in [-1.0, 1.0]$). The results of its application are shown in Table 1 for the unimodal functions of the benchmark set.

Fig. 4 displays the CPEM results of the three EAs considered here over the unimodal subset as a function of the longest path to the optimum (extracted from Table 1). As shown, the performance of the three EAs degrades as the path length increases, without any success for lengths above 1000. The RCGA seems to be worse than the other two algorithms, because it starts to fail with lengths of 400. Both the DE and CMA-ES show a successful overall performance over unimodal landscapes. They fail only in 6 and 3 out

of 30 cases respectively, as shown in Table 2. The DE implementation used here can adjust the mutation step size during the optimization process, adapting it to the landscape. The case of the CMA-ES is similar. The mutation step size is adapted depending on previous step sizes during evolution, that is, when mutations are carried out consistently in the same direction, the step size is increased to cover the same distance in less steps. Obviously, this also increases the convergence speed. This strategy makes this algorithm the fastest one, as can be seen in Table 2 where the CMA-ES outperforms the other two algorithms. There are three exceptions: the *Easom*, *Perm* and *Schwefel 2.21* functions. These three functions share the feature of presenting broad plateaus with very little information. Without this information it is very difficult for the CMA-ES to generate the covariance matrix, so the performance of the algorithm becomes worse.

Thus, the longest path to the optimum is a relevant landscape feature to explain EA performance in unimodal landscapes and it should be considered for characterizing the algorithms properly.

Multimodal functions are those that contain many local and global optima. The complexity of these functions is usually measured only by the number of local optima they have, and this is not always enough to explain the behavior of EAs over them. For example, as shown in Fig. 5(a), it is hard to explain the behavior of the CMA-ES algorithm over the multimodal functions of the benchmark set using only the number of local optima. Thus, we propose the consideration of more detailed morphological features. These features are three: size of the optimum and the largest attraction basins (being this size the frequency with which a point of the search space ends a local search at an optima) and maximum distance between attraction basins.

An algorithm has been developed [4] based on the distribution of attraction basins in multimodal landscapes to estimate the values for these three features as an extension of the work of [6]. The results provided by this algorithm over the multimodal function subset are displayed in Table 3. Due to the computational complexity of this method, these results for the scalable function set are an estimation obtained by analyzing 5-D fitness landscapes. In the non-scalable function set the results are obtained using their specific dimensionality.

As expected, different EAs are affected by different features of the distribution of the attraction basins of a fitness

Table 1: Longest path length to the optimum.

Function	Path length
Perm	1060.46
Schwefel 2.21	573.48
Colville	508.30
Schwefel 1.2	453.02
Schwefel 2.22	265.15
Axis Parallel	224.97
Zakharov	220.41
Sphere Model	204.12
Matyas	188.31
SumOf	142.14
Step	126.60
Easom	24.40

Table 2: Results provided by the three analyzed algorithms over the unimodal functions subset

Function	D	CPEM		
		RCGA	CMA-ES	DE
Axis Parallel-L	10	$1.55 \cdot 10^{-7}$	$2.11 \cdot 10^{-8}$	$7.34 \cdot 10^{-8}$
	30	$2.21 \cdot 10^{-7}$	$2.58 \cdot 10^{-8}$	$8.98 \cdot 10^{-8}$
	50	$4.27 \cdot 10^{-7}$	$3.03 \cdot 10^{-8}$	$9.88 \cdot 10^{-8}$
Schwefel 2.22-L	10	$1.66 \cdot 10^{-7}$	$3.27 \cdot 10^{-8}$	$9.19 \cdot 10^{-8}$
	30	$2.36 \cdot 10^{-7}$	$5.83 \cdot 10^{-8}$	$1.01 \cdot 10^{-7}$
	50	$6.10 \cdot 10^{-7}$	$1.09 \cdot 10^{-7}$	$1.08 \cdot 10^{-7}$
Sphere Model-L	10	$1.53 \cdot 10^{-7}$	$1.82 \cdot 10^{-8}$	$6.90 \cdot 10^{-8}$
	30	$1.85 \cdot 10^{-7}$	$1.76 \cdot 10^{-8}$	$8.08 \cdot 10^{-8}$
	50	$3.45 \cdot 10^{-7}$	$1.74 \cdot 10^{-8}$	$8.92 \cdot 10^{-8}$
Step-L	10	$6.29 \cdot 10^{-8}$	$9.92 \cdot 10^{-9}$	$1.47 \cdot 10^{-8}$
	30	$9.02 \cdot 10^{-8}$	$2.44 \cdot 10^{-8}$	$3.51 \cdot 10^{-8}$
	50	$1.01 \cdot 10^{-7}$	$4.99 \cdot 10^{-8}$	$3.88 \cdot 10^{-8}$
SumOf-L	10	$4.60 \cdot 10^{-8}$	$1.24 \cdot 10^{-7}$	$1.69 \cdot 10^{-8}$
	30	$3.10 \cdot 10^{-8}$	$2.89 \cdot 10^{-8}$	$2.13 \cdot 10^{-8}$
	50	$2.62 \cdot 10^{-8}$	$4.46 \cdot 10^{-8}$	$1.97 \cdot 10^{-8}$
Easom-NL	2	$1.52 \cdot 10^{-7}$	$1.95 \cdot 10^{-1}$	$2.72 \cdot 10^{-7}$
Schwefel 2.21-NL	10	$2.16 \cdot 10^{-2}$	$1.99 \cdot 10^{-7}$	$3.14 \cdot 10^{-7}$
	30	$6.29 \cdot 10^{-3}$	$9.88 \cdot 10^1$	$5.82 \cdot 10^{-7}$
	50	$1.08 \cdot 10^{-1}$	$1.00 \cdot 10^2$	$7.57 \cdot 10^{-7}$
Colville-N	4	$4.36 \cdot 10^{-1}$	$4.81 \cdot 10^{-8}$	$1.13 \cdot 10^{-7}$
Matyas-N	2	$1.60 \cdot 10^{-7}$	$1.40 \cdot 10^{-8}$	$3.86 \cdot 10^{-8}$
Perm-N	10	$7.80 \cdot 10^{-2}$	$8.40 \cdot 10^{-2}$	$1.56 \cdot 10^{-6}$
	30	$1.38 \cdot 10^0$	$1.61 \cdot 10^1$	$3.74 \cdot 10^0$
	50	$2.75 \cdot 10^0$	$2.26 \cdot 10^1$	$5.29 \cdot 10^{99}$
Schwefel 1.2-N	10	$2.71 \cdot 10^{-1}$	$2.74 \cdot 10^{-8}$	$1.44 \cdot 10^{-7}$
	30	$3.99 \cdot 10^1$	$5.08 \cdot 10^{-8}$	$4.73 \cdot 10^{-7}$
	50	$3.98 \cdot 10^2$	$7.19 \cdot 10^{-8}$	$2.26 \cdot 10^{-6}$
Zakharov (N)	10	$7.66 \cdot 10^{-5}$	$2.46 \cdot 10^{-8}$	$1.12 \cdot 10^{-7}$
	30	$5.63 \cdot 10^{-1}$	$5.52 \cdot 10^{-8}$	$5.47 \cdot 10^{-7}$
	50	$1.56 \cdot 10^1$	$8.99 \cdot 10^{-8}$	$1.06 \cdot 10^0$

landscape. For example, Fig. 5(b) exemplifies the fact that the performance of the CMA-ES algorithm is closely related to the maximum distance between the optimum attraction basin and all the attraction basins of the function, getting worse as this distance increases. This behavior is due to the highly exploitative strategy of the CMA-ES. In functions where the attraction basins are spread out over the fitness landscape, the algorithm should explore it in order to reach the attraction basin containing the optimum and then exploit it to reach the optimum. A second conclusion that can be extracted from this feature is that the DE algorithm outperforms the results of the CMA-ES on those functions where the attraction basins are spread out over the whole search space (Table 3 displays the maximum distance between attraction basins).

Another interesting feature to analyze in multimodal functions is the relationship between the optimum and the largest attraction basins, i.e., if those attraction basins are the same or not. According to the size of the attraction basin containing the optimum in the functions of the benchmark set used here, when it is smaller than a 10% of the search space (see Table 3), an exploratory behavior is needed to reach this attraction basin, and after this, exploit it until the optimum solution is reached. This is the reason why the

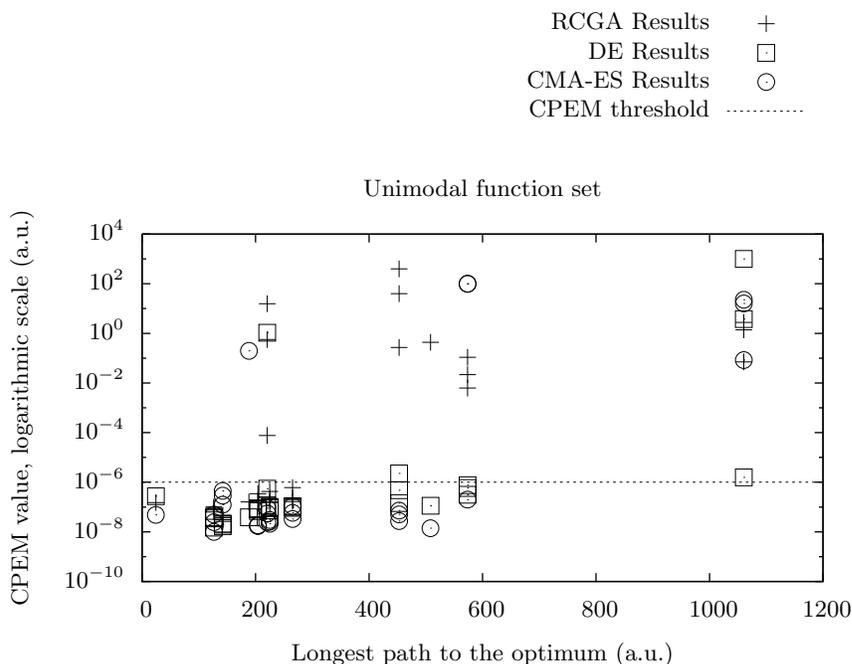


Figure 4: Results provided by the three algorithms over the unimodal functions subset characterized by the length of the longest path to the optimum.

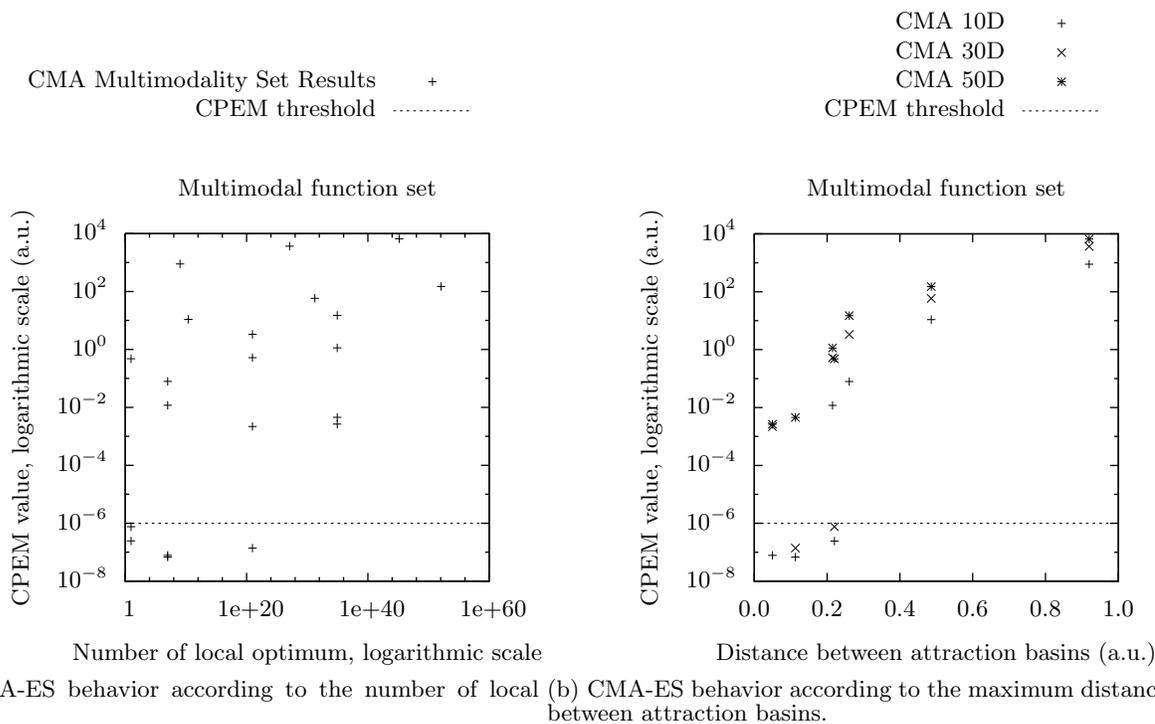


Figure 5: Results provided by the CMA-ES algorithm in the multimodal function subset.

DE outperforms the behavior of the CMA-ES over those functions with small optimum attraction basins (see results on Table 4). On the other hand, when the optimum attraction basin is large enough to be easily reached, the CMA-ES results improve on those of the DE because of its exploita-

tion capabilities. According to the relationship between the largest and the optimum attraction basins, when the optimum attraction basin is not the largest one, a more explorative behavior is required. In this type of functions, algorithms like the CMA-ES tend towards the largest attraction

Table 3: Proposed features of the multimodal functions.

Function	Attr. Basin Size		Max.
	Largest	Optima	Distance
Aluffi-Pentini's	$8.60 \cdot 10^{-1}$	$8.60 \cdot 10^{-1}$	$7.00 \cdot 10^{-2}$
Becker and Lago	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$0.00 \cdot 10^0$
Bohachevsky 1	$7.54 \cdot 10^{-1}$	$7.54 \cdot 10^{-1}$	$9.00 \cdot 10^{-3}$
Cosine Mixture	$2.60 \cdot 10^{-2}$	$2.60 \cdot 10^{-2}$	$2.15 \cdot 10^{-1}$
Rastrigin	$8.00 \cdot 10^{-3}$	$8.00 \cdot 10^{-3}$	$4.86 \cdot 10^{-1}$
Schwefel	$2.36 \cdot 10^{-1}$	$2.96 \cdot 10^{-5}$	$9.20 \cdot 10^{-1}$
Ackley's	$1.00 \cdot 10^{-3}$	$1.00 \cdot 10^{-3}$	$5.00 \cdot 10^{-1}$
Griewank	$1.00 \cdot 10^{-3}$	$2.00 \cdot 10^{-4}$	$4.55 \cdot 10^{-1}$
Levy	$5.77 \cdot 10^{-1}$	$5.77 \cdot 10^{-1}$	$2.61 \cdot 10^{-1}$
Penalized 1	$1.28 \cdot 10^{-1}$	$1.28 \cdot 10^{-1}$	$1.13 \cdot 10^{-1}$
Penalized 2	$1.74 \cdot 10^{-1}$	$1.74 \cdot 10^{-1}$	$5.00 \cdot 10^{-2}$
Beale	$4.69 \cdot 10^{-1}$	$4.69 \cdot 10^{-1}$	$5.91 \cdot 10^{-1}$
Bohachevsky 2	$5.35 \cdot 10^{-1}$	$5.35 \cdot 10^{-1}$	$9.00 \cdot 10^{-3}$
Dekkers & Aarts	$8.79 \cdot 10^{-1}$	$8.79 \cdot 10^{-1}$	$2.91 \cdot 10^{-1}$
Goldstein Price	$5.03 \cdot 10^{-1}$	$5.03 \cdot 10^{-1}$	$4.24 \cdot 10^{-1}$
Hartman 3	$6.22 \cdot 10^{-1}$	$6.22 \cdot 10^{-1}$	$4.66 \cdot 10^{-1}$
Hartman 6	$6.25 \cdot 10^{-1}$	$2.74 \cdot 10^{-1}$	$4.50 \cdot 10^{-1}$
Kowalik	$1.28 \cdot 10^{-1}$	$2.87 \cdot 10^{-4}$	$4.27 \cdot 10^{-1}$
Rosenbrock	$3.76 \cdot 10^{-1}$	$3.76 \cdot 10^{-4}$	$2.20 \cdot 10^{-1}$
Shekel Fam. 5	$3.12 \cdot 10^{-1}$	$2.61 \cdot 10^{-1}$	$3.89 \cdot 10^{-1}$
Shekel Fam. 7	$2.95 \cdot 10^{-1}$	$2.95 \cdot 10^{-1}$	$3.77 \cdot 10^{-1}$
Shekel Fam. 10	$2.40 \cdot 10^{-1}$	$2.40 \cdot 10^{-1}$	$3.81 \cdot 10^{-1}$
Shekel's Foxholes	$6.70 \cdot 10^{-2}$	$6.70 \cdot 10^{-2}$	$4.87 \cdot 10^{-1}$
Six Hump	$2.65 \cdot 10^{-1}$	$2.32 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$

basins. The results of the CMA-ES confirm the fact that an exploitative behavior is not recommended. With this type of strategy, it is easier to reach the largest attraction basin than the optimum one. There is, however, an exception that also reinforces the importance of considering the distance between attraction basins. In the *SixHump* function the optimum attraction basin is smaller than the largest one but the distance between them is short enough to permit "jumping" from one to the other allowing the CMA-ES to obtain more accurate results than the DE. The opposite behavior occurs on the other functions where the distance is large and the DE outperforms the CMA-ES.

Summarizing, as we have shown in this section, more information than is usually provided about the modality of the benchmark functions used to characterize EAs is needed in order to characterize their behavior appropriately. This extra information is required both in unimodal and multimodal functions. The path length in unimodal functions is useful to determine the difficulty of the functions: the longest the path the harder to find the solution for an EA without a strategy that adjusts the mutation step automatically, like the RCGA implementation used here. Regarding multimodal functions, the number of optima does not provide enough information. Functions with a high number of local optima that are not spread out over the search space are easier to solve than functions with few local optima but widely spread out over the landscape. Moreover, the behavior of the EAs is highly dependent on the size of the attraction basins. Functions where the optimum attraction basin is not the largest one are more difficult than those with very large optimum attraction basins, without taking into account the number of them.

Table 4: Results provided by the three analyzed algorithms over the multimodal functions subset

Function	D	CPEM		
		RCGA	CMA-ES	DE
Aluffi-Pentini's-L	2	$6.64 \cdot 10^{-8}$	$2.57 \cdot 10^{-8}$	$3.94 \cdot 10^{-8}$
Becker and Lago-L	2	$7.51 \cdot 10^{-8}$	$1.61 \cdot 10^{-8}$	$5.06 \cdot 10^{-8}$
Bohach.1-L	2	$9.62 \cdot 10^{-8}$	$3.56 \cdot 10^{-8}$	$5.66 \cdot 10^{-8}$
Cosine Mixture-L	10	$9.28 \cdot 10^{-8}$	$1.18 \cdot 10^{-2}$	$4.98 \cdot 10^{-8}$
	30	$1.28 \cdot 10^{-7}$	$5.20 \cdot 10^{-1}$	$5.90 \cdot 10^{-8}$
	50	$2.04 \cdot 10^{-7}$	$1.41 \cdot 10^0$	$6.47 \cdot 10^{-8}$
Rastrigin-L	10	$6.15 \cdot 10^{-7}$	$1.09 \cdot 10^1$	$1.18 \cdot 10^{-7}$
	30	$6.29 \cdot 10^{-6}$	$5.86 \cdot 10^1$	$1.59 \cdot 10^{-7}$
	50	$5.38 \cdot 10^{-7}$	$1.50 \cdot 10^2$	$3.78 \cdot 10^{-7}$
Schwefel-L	10	$5.40 \cdot 10^{-2}$	$8.98 \cdot 10^2$	$1.66 \cdot 10^{-7}$
	30	$7.06 \cdot 10^{-5}$	$3.69 \cdot 10^3$	$3.27 \cdot 10^{-7}$
	50	$4.39 \cdot 10^{-7}$	$6.63 \cdot 10^3$	$5.70 \cdot 10^{-7}$
Ackleys-L	10	$2.21 \cdot 10^{-7}$	$5.31 \cdot 10^{-8}$	$1.10 \cdot 10^{-7}$
	30	$2.93 \cdot 10^{-7}$	$1.01 \cdot 10^{-7}$	$1.01 \cdot 10^{-7}$
	50	$7.67 \cdot 10^{-7}$	$1.49 \cdot 10^{-7}$	$1.49 \cdot 10^{-7}$
Griewank NL	30	$4.05 \cdot 10^{-7}$	$9.86 \cdot 10^{-4}$	$6.18 \cdot 10^{-8}$
	50	$4.50 \cdot 10^{-3}$	$1.91 \cdot 10^{-7}$	$9.29 \cdot 10^{-8}$
Levy-NL	10	$9.54 \cdot 10^{-8}$	$7.92 \cdot 10^{-2}$	$5.03 \cdot 10^{-8}$
	30	$1.41 \cdot 10^{-7}$	$3.33 \cdot 10^0$	$6.18 \cdot 10^{-8}$
	50	$2.18 \cdot 10^{-7}$	$1.50 \cdot 10^1$	$6.95 \cdot 10^{-8}$
Penalized1 NL	10	$1.34 \cdot 10^{-7}$	$6.81 \cdot 10^{-8}$	$6.00 \cdot 10^{-8}$
	30	$1.87 \cdot 10^{-7}$	$1.40 \cdot 10^{-7}$	$6.98 \cdot 10^{-8}$
	50	$2.58 \cdot 10^{-7}$	$4.52 \cdot 10^{-3}$	$7.76 \cdot 10^{-8}$
Penalized2 NL	10	$2.18 \cdot 10^{-7}$	$7.81 \cdot 10^{-8}$	$6.27 \cdot 10^{-8}$
	30	$2.42 \cdot 10^{-7}$	$2.20 \cdot 10^{-3}$	$1.21 \cdot 10^{-7}$
	50	$3.39 \cdot 10^{-7}$	$2.64 \cdot 10^{-3}$	$8.51 \cdot 10^{-8}$
Beale-N	2	$2.30 \cdot 10^{-7}$	$3.77 \cdot 10^{-8}$	$4.70 \cdot 10^{-8}$
Bohach.2-N	2	$9.85 \cdot 10^{-8}$	$3.52 \cdot 10^{-8}$	$6.26 \cdot 10^{-8}$
Dekkers & Aarts-N	2	$4.38 \cdot 10^{-8}$	$1.41 \cdot 10^{-8}$	$1.85 \cdot 10^{-8}$
Goldstein Price-N	2	$1.13 \cdot 10^{-7}$	$3.01 \cdot 10^{-8}$	$5.25 \cdot 10^{-8}$
Griewank-N	10	$1.03 \cdot 10^{-2}$	$3.20 \cdot 10^{-2}$	$3.93 \cdot 10^{-7}$
Hart3-N	3	$3.77 \cdot 10^{-8}$	$1.71 \cdot 10^{-8}$	$2.94 \cdot 10^{-8}$
Hart6-N	6	$1.01 \cdot 10^{-2}$	$5.30 \cdot 10^{-3}$	$1.06 \cdot 10^{-1}$
Rosenbrock (N)	10	$5.28 \cdot 10^0$	$2.45 \cdot 10^{-7}$	$6.47 \cdot 10^{-7}$
	30	$2.38 \cdot 10^1$	$7.60 \cdot 10^{-7}$	$8.48 \cdot 10^{-1}$
	50	$3.03 \cdot 10^1$	$4.78 \cdot 10^{-1}$	$3.67 \cdot 10^1$
Kowalik-N	4	$5.13 \cdot 10^{-4}$	$1.32 \cdot 10^{-4}$	$3.14 \cdot 10^{-4}$
ShekF5-N	4	$2.09 \cdot 10^0$	$7.03 \cdot 10^{-1}$	$2.07 \cdot 10^{-7}$
ShekF7-N	4	$9.66 \cdot 10^{-1}$	$1.33 \cdot 10^{-7}$	$1.35 \cdot 10^{-7}$
ShekF10 N	2	$2.68 \cdot 10^{-1}$	$1.34 \cdot 10^{-7}$	$1.72 \cdot 10^{-7}$
Shekel's Foxholes-N	2	$1.19 \cdot 10^{-1}$	$5.31 \cdot 10^{-7}$	$1.17 \cdot 10^{-7}$
Six Hump-N	2	$5.57 \cdot 10^{-8}$	$1.59 \cdot 10^{-8}$	$5.02 \cdot 10^{-8}$

5. CONCLUSIONS

This paper emphasizes the need for a more in depth analysis of the fitness landscapes in the functions typically used for EA competitions with the objective of obtaining more

useful conclusions about the behavior of the EAs over them. This more in depth analysis is performed in this paper and information such as the type of separability, the path length to the optimum and the size and distance between attraction basins is provided for a benchmark set. This information allows a higher degree of knowledge regarding separability and modality, which are the most commonly used features for the characterization of benchmark sets in EA competitions. In order to exemplify their usefulness, they have been applied to study the behavior of three well-known EAs showing that this deeper analysis permits obtaining reliable conclusions and relationships between the search strategy of the EAs and their performance over different types of functions.

6. ACKNOWLEDGMENTS

This work was partially funded by the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

7. REFERENCES

- [1] D. H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer, Boston, 1987.
- [2] E. Alba, B. Dorronsoro, M. Giacobini, and M. Tomassini. Decentralized cellular evolutionary algorithms. *Handbook of Bioinspired Algorithms and Applications*, 7:103–120, 2005.
- [3] A. Auger and N. Hansen. A Restart CMA Evolution Strategy with Increasing Population Size. In *In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, 2005.*, volume 2, pages 1769–1776, Apr 2005.
- [4] P. Caamaño, A. Prieto, J. A. Becerra, F. Bellas, and R. J. Duro. Real-valued multimodal fitness landscape characterization for evolution. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I, ICONIP'10*, pages 567–574, Berlin, Heidelberg, 2010. Springer-Verlag.
- [5] P. Caamano, F. Bellas, J. A. Becerra, and R. J. Duro. Application domain study of evolutionary algorithms in optimization problems. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO '08*, pages 495–502, New York, NY, USA, 2008. ACM.
- [6] J. Garnier and L. Kallel. How to detect all maxima of a function? In *Proceedings of the Second EVONET Summer School on Theoretical Aspects of Evolutionary Computing*, pages 343–370, 1999.
- [7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [8] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, 12:265–319, 1998.
- [9] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. Technical report, Nanyang Technological University, 2006.
- [10] M. Locatelli. A note on the Griewank test function. *Journal of Global Optimization*, 25(2):169–174, 2003.
- [11] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 113–122, New York, NY, USA, 2006. ACM.
- [12] A. Nimitawat and P. Nanakorn. Automated layout design of beam-slab floors using a genetic algorithm. *Computers and Structures*, 87(21-22):1308 – 1330, 2009.
- [13] G. Rawlins. *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.
- [14] J. Rönkkönen, S. Kukkonen, and K. V. Price. Real-Parameter Optimization with Differential Evolution. In *The 2005 IEEE International Congress on Evolutionary Computation (CEC'05)*, page 12, Apr 2005.
- [15] R. Storn and K. V. Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341 – 359, Nov 1997.
- [16] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real Parameter Optimization. Technical report, Nanyang Technological University, 2005.
- [17] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise. Benchmark Functions for the CEC 2010 Special Session and Competition on Large-Scale Global Optimization. Technical report, University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Héfěi, Ānhuī, China, Jan. 8, 2010.
- [18] K. Tang, X. Yao, P. Suganthan, C. Macnish, Y. Chen, C. Chen, and Z. Yang. Benchmark Functions for the CEC 2008 Special Session and Competition on Large Scale Global Optimization. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, November 2007.
- [19] M.-H. Tseng and H.-C. Liao. The genetic algorithm for breast tumor diagnosis—The case of DNA viruses. *Applied Soft Computing*, 9(2):703 – 710, 2009.
- [20] L. D. Whitley, K. E. Mathias, S. B. Rana, and J. Dzubera. Building Better Test Functions. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 239–247, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.