Real-Valued Multimodal Fitness Landscape Characterization for Evolution

P. Caamaño, A. Prieto, J.A. Becerra, F. Bellas, and R.J. Duro

Integrated Group of Engineering Research,
Escuela Politécnica Superior, Campus de Esteiro, 15403 Ferrol, Spain
{pcsobrino,abprieto,ronin,fran,richard}@udc.es
http://www.gii.udc.es

Abstract. This paper deals with the characterization of the fitness landscape of multimodal functions and how it can be used to choose the most appropriate evolutionary algorithm for a given problem. An algorithm that obtains a general description of real valued multimodal fitness landscapes in terms of the relative number of optima, their sparseness, the size of their attraction basins and the evolution of this size when moving away from the global optimum is presented and used to characterize a set of well-known multimodal benchmark functions. To illustrate the relevance of the information obtained and its relationship to the performance of evolutionary algorithms over different fitness landscapes, two evolutionary algorithms, Differential Evolution and Covariance Matrix Adaptation, are compared over the same benchmark set showing their behavior depending on the multimodal features of each landscape.

1 Introduction

The application of evolutionary algorithms (EA) to real-valued problems is one of the most prolific topics in evolutionary computation due to the successful results obtained, mainly in high dimensional cases [1][2], where other approaches fail. The main drawback detected by the users is the uncertainty about which would be the most appropriate EA for a particular problem or fitness landscape. This usually implies either a trial and error stage until a proper algorithm is found or the use of suboptimal algorithms which may waste precious resources and time. Due to the usually high computational cost of EAs, much attention has been paid to solving this problem through the study and analysis of the features that characterize a landscape to determine its complexity prior to EA selection [3][4].

The multimodality of a fitness landscape is one of the main features that contribute to the increase in the difficulty of an optimization problem. It seems clear that a large number of local optima is a characteristic that complicates the search, even more as dimensionality grows. However, this feature is not so relevant on its own, and must be combined with the sizes of the attraction basins of the local optima or the distances between these local optima and the global optimum for the analysis to be meaningful. Here, we propose a real-valued fitness landscape analysis algorithm that provides such detailed information and discuss its implications using two very successful EAs, Differential Evolution and Covariance Matrix Adaptation.

The rest of the paper is structured as follows: section 2 is devoted to the formal presentation of the algorithm and its application to the characterization of a multimodal benchmark function set. Section 3 presents the results obtained when applying the algorithm to the problem of selecting between Differential Evolution (DE) and Covariance Matrix Adaptation (CMA), showing how this detailed information is crucial in some cases. Finally, the main conclusions of this work are commented in Section 4.

2 Multimodal fitness landscape characterization

In [5], the authors present a method to analyze multimodal fitness landscapes based on the attraction basins theory for binary valued functions. They suggest an algorithm that, given a landscape and a local search operator selected for the particular features of the EA under study, estimates the number of local optima and the attraction basins size with the objective of determining the complexity of such a landscape for the particular operator. Starting from the background ideas of this work, we have developed a more general analysis algorithm for real valued problems that uses a generic local search operator to estimate the local optima distribution with the objective of comparing different EAs in these terms.

First of all and for the sake of formalization, a fitness landscape is defined as a composition of three elements [6]:

- A set S of possible solutions, also known as the search space of the problem.
- A neighborhood function $\mathcal{V} : S \longrightarrow 2^{|S|}$, which assigns to each solution $s \in S$ a subset of neighbors $\mathcal{V}(\int) \subset S$. The concept of neighborhood is defined from a local search operator, i. e., if we have an operator μ, the neighborhood of a solution s is the set $\mathcal{V}(s) = y \in S | y = \mu(s)$.
- A fitness function $f : S \longrightarrow \mathbb{R}$, that assigns the fitness value to each solution.

The degree of difficulty of a fitness landscape is linked, among other features, to the existence of local optima. Thus:

- A fitness landscape is *multimodal* when it has more than one local optimum.
- A solution $m \in S$ is a local optimum if, for every solution of its neighborhood, there is no solution with a better fitness value.
- Each local optimum of the fitness landscape is associated with an attraction basin $E(m)$.
- A solution s_0 belongs to the attraction basin of the local optimum m if, by applying operator μ a finite number of steps, the search process starting at s_0 ends up at m.

The distribution of local optima and the size of every attraction basin are important factors in the performance of any metaheuristics [7] and, therefore, of evolutionary algorithms.

The proposed algorithm applies a local search to several random points $x_i \in S$, so that by starting at each x_i the algorithm ends up near one of the local optima. Therefore, to apply the algorithm, we assume that a landscape L, defined by a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ with n parameters, may be divided into M partitions or subspaces, each one of them is an attraction basin for one of the local optima of the landscape. The algorithm is described in the following pseudocode:

Multimodality Analysis Algorithm

```
for all  x_i   N do
    do
      Record, for all  μ-neighbors of  x_i,  f(μ_j(x_i)).
      Assign x_i = μ_j(x_i) where f(μ_j(x_i)) reach the minimum value.
    while there is strictly positive improvement in the
     μ -neighbors
end for
Where:
    -  μ (x_i) is the application of the  μ  operator on  x_i.
    - The  μ  operator is applied P times in each loop for each  x_i.
```

As shown, it is run for N random points of the search space. The μ operator generates each random point within an area of radius r around point x_i. After the execution of the local search, we store the number of random points belonging to each attraction basin. As we are working in a continuous domain, we have considered that a solution s_i belongs to an attraction basin if the Euclidean distance between s_i and the local optimum of this basin is less than δ. The point with the best fitness value in each attraction basin is taken as its center and corresponds to a local optimum. Finally, the best value of all the local optima is taken as the global optimum.

The number of random points (N) could be set to a fixed number but we have implemented it dependent on the search process. That is, we start with only 1 point and add one more each step until a stopping criterion is met. In particular, we stop the search process when the statistical frequencies of all the local optima do not change more than a threshold, $1.00E{-3}$ in this case, from one execution to the next.

The information provided by the algorithm has been summarized into the following measures:

- Number of attraction basins: an estimation of the number of attraction basins, and thus, of local optima, is produced.
- Size of the attraction basins: it can be calculated as the statistical frequency with which the local search algorithm reaches its corresponding local optimum.
- Distance between attraction basins: with this measure we can estimate the distribution of the local optima. In particular, in our experiments we have measured:
  - Maximum distance between attraction basins.
  - Distance between the global optimum and the local optimum with the largest attraction basin.

Table 1. Analysis of the set of multimodal benchmark functions.

Function	m	Attr. Basin Size		Distance	
		Largest	Optimum	lar.-opt.	Maxima
$\mathbf{f_1}$ **Cosine**	$\approx 5^n$	$2.60E{-}2$	$2.60E{-}2$	$0.00E{+}0$	$2.15E{-}1$
$\mathbf{f_2}$ **Rastrigin**	$\approx 11^n$	$8.00E{-}3$	$8.00E{-}3$	$0.00E{+}0$	$4.86E{-}1$
$\mathbf{f_3}$ **Schwefel**	$\approx 8^n$	$2.36E{-}1$	$2.96E{-}5$	$9.20E{-}1$	$9.20E{-}1$
$\mathbf{f_4}$ **Ackleys**	$\approx 65^n$	$1.00E{-}3$	$1.00E{-}3$	$0.00E{+}0$	$5.00E{-}1$
$\mathbf{f_5}$ **Griewank**	$\approx 4^n$	$1.00E{-}3$	$2.00E{-}4$	$3.97E{-}1$	$4.55E{-}1$
$\mathbf{f_6}$ **Levy**	$\approx 5^n$	$5.77E{-}1$	$5.77E{-}1$	$0.00E{+}0$	$2.61E{-}1$
$\mathbf{f_7}$ **Penalized1**	$\approx 5^n$	$1.28E{-}1$	$1.28E{-}1$	$0.00E{+}0$	$1.13E{-}1$
$\mathbf{f_8}$ **Penalized2**	$\approx 5^n$	$1.74E{-}1$	$1.74E{-}1$	$0.00E{+}0$	$5.00E{-}2$
$\mathbf{f_9}$ **Rosenbrock**	2	$3.76E{-}1$	$0.376E{-}1$	$0.00E{+}0$	$2.20E{-}1$
$\mathbf{f_{10}}$ **Deckers**	3	$8.79E{-}1$	$8.79E{-}1$	$0.00E{+}0$	$2.91E{-}1$
$\mathbf{f_{11}}$ **Bohac1**	13	$7.54E{-}1$	$7.54E{-}1$	$0.00E{+}0$	$9.00E{-}3$
$\mathbf{f_{12}}$ **Bohac2**	10	$5.35E{-}1$	$5.35E{-}1$	$0.00E{+}0$	$9.00E{-}3$
$\mathbf{f_{13}}$ **Foxholes**	25	$6.70E{-}2$	$6.70E{-}2$	$0.00E{+}0$	$4.87E{-}1$
$\mathbf{f_{14}}$ **Beale**	4	$4.69E{-}1$	$4.69E{-}1$	$0.00E{+}0$	$5.91E{-}1$
$\mathbf{f_{15}}$ **Aluffi**	10	$8.60E{-}1$	$8.60E{-}1$	$0.00E{+}0$	$7.00E{-}2$
$\mathbf{f_{16}}$ **Becker**	4	$1.00E{+}0$	$1.00E{+}0$	$0.00E{+}0$	$0.00E{+}0$
$\mathbf{f_{17}}$ **Goldstein**	6	$5.03E{-}1$	$5.03E{-}1$	$0.00E{+}0$	$4.24E{-}1$
$\mathbf{f_{18}}$ **Hartman3**	3	$6.22E{-}1$	$6.22E{-}1$	$0.00E{+}0$	$4.66E{-}1$
$\mathbf{f_{19}}$ **Hartman6**	2	$6.25E{-}1$	$2.74E{-}1$	$4.50E{-}1$	$4.50E{-}1$
$\mathbf{f_{20}}$ **Shekel5**	5	$3.12E{-}1$	$2.61E{-}1$	$2.02E{-}1$	$3.89E{-}1$
$\mathbf{f_{21}}$ **Shekel7**	7	$2.95E{-}1$	$2.95E{-}1$	$0.00E{+}0$	$3.77E{-}1$
$\mathbf{f_{22}}$ **Shekel10**	10	$2.40E{-}1$	$2.40E{-}1$	$0.00E{+}0$	$3.81E{-}1$
$\mathbf{f_{23}}$ **SixHump**	6	$2.65E{-}1$	$2.32E{-}1$	$1.14E{-}1$	$1.40E{-}1$

3 Application

In this section we apply the previous algorithm to a real valued multimodal function benchmark set to characterize it using the measures stated above and then run two very relevant EAs over the benchmark set trying to understand the influence of these parameters on the performance of the algorithms. The benchmark set is made up of 23 multimodal functions with different topological features (see Table 1). Functions $f_1 - f_9$ [8] are scalable (in this work the results are obtained for dimension 5) and functions $f_{10} - f_{23}$ [9] are non-scalable with low dimensions (from 2 to 6).

The results obtained by applying the characterization algorithm to the selected benchmark function set are presented in Table 1. The columns of the table display, from left to right, the measures commented above: an estimation of the number of local minima (m), the size of the largest basin and the optimum attraction basin (largest and optimum respectively), the distance between the largest and the optimum attraction basins and the maximum distance between attraction basins for the functions that make up the selected benchmark set.

Table 2. Success Performance (SP) and Success Rate (SR) for multimodal functions.

Function	Dim.	CMA		DE	
		SR	SP	SR	SP
$\mathbf{f_1}$ **Cosine**	5	1.00	2636.60	**1.00**	**2232.84**
$\mathbf{f_2}$ **Rastrigin**	5	0.08	621949.00	**1.00**	**5142.76**
$\mathbf{f_3}$ **Schwefel**	5	0.08	608619.00	**1.00**	**7483.64**
$\mathbf{f_4}$ **Ackleys**	5	**1.00**	**1864.60**	1.00	5171.72
$\mathbf{f_5}$ **Griewank**	5	0.04	1232999.00	**1.00**	**19992.92**
$\mathbf{f_6}$ **Levy**	5	**1.00**	**815.00**	1.00	2210.80
$\mathbf{f_7}$ **Penalized1**	5	**1.00**	**1484.20**	1.00	2759.96
$\mathbf{f_8}$ **Penalized2**	5	**1.00**	**2074.60**	1.00	2869.88
$\mathbf{f_9}$ **Rosenbrock**	5	**1.00**	**3683.80**	1.00	16258.24
$\mathbf{f_{10}}$ **Deckers**	2	**1.00**	**282.60**	1.00	370.92
$\mathbf{f_{11}}$ **Bohac1**	2	**1.00**	**711.00**	1.00	1252.64
$\mathbf{f_{12}}$ **Bohac2**	2	**1.00**	**704.60**	1.00	1247.80
$\mathbf{f_{13}}$ **Foxholes**	2	1.00	10612.20	**1.00**	**2334.20**
$\mathbf{f_{14}}$ **Beale**	2	**1.00**	**753.40**	1.00	939.44
$\mathbf{f_{15}}$ **Aluffi**	2	**1.00**	**513.40**	1.00	835.48
$\mathbf{f_{16}}$ **Becker**	2	**1.00**	**523.80**	1.00	526.88
$\mathbf{f_{17}}$ **Goldstein**	2	**1.00**	**602.60**	1.00	1050.36
$\mathbf{f_{18}}$ **Hartman3**	3	**1.00**	**514.20**	1.00	1798.92
$\mathbf{f_{19}}$ **Hartman6**	6	**0.96**	**6034.00**	0.28	156288.86
$\mathbf{f_{20}}$ **Shekel5**	4	0.88	24175.36	**1.00**	**8261.76**
$\mathbf{f_{21}}$ **Shekel7**	4	**1.00**	**5315.00**	1.00	5406.04
$\mathbf{f_{22}}$ **Shekel10**	4	**1.00**	**5375.00**	1.00	6885.12
$\mathbf{f_{23}}$ **SixHump**	2	**1.00**	**317.40**	1.00	1898.46

To illustrate the practical relevance of the characterization algorithm, we have run two well-known evolutionary algorithms, Differential Evolution (DE) [10] and Covariance Matrix Adaptation (CMA) [11], over the same benchmark function set. These two algorithms operate quite differently, the DE is a better explorer and the CMA a really good exploiter, although its exploration capabilities are limited. We have implemented the canonical version of the DE algorithm, *DE/rand/1/bin* with the parameters recommended in [12] and the CMA version and parameters as presented in [13]. For each benchmark function, 25 independent runs were performed. The stopping criterion is based on the number of function evaluations as in [2], with a maximum number set to $10000n$, being n the dimensionality of the problem. The results obtained in terms of Success Performance (SP) and Success Rate (SR) [2] are displayed in Table 2. The SP indicates the average number of function evaluations the algorithm needs to reach the optima and is an indication of the convergence speed. The SR is the number of success runs measured as a fraction of unity, it is 1.00 if all the runs are solved.

The table clearly shows that both algorithms obtain very successful results over the benchmark set. The CMA solves all the runs in 18 of the 23 functions while the DE solves all the runs for 22 of the 23. This is basically due to the fact that we are using low dimensions in the functions and these two algorithms are very powerful. Consequently, in order to determine how the features of the fitness landscape produced by the algorithm can be associated to the performance of each one of the EAs, the analysis must be carried out in terms of the functions where they fail and in terms of their relative convergence speed.

A closer look at the results of Table 1, shows that there are 18 functions where the attraction basin of the optimum is the largest one (simply by comparing the values in columns *largest* and *optimum*) making the exploration phase easy, that is, it is easy to have samples in a large attraction basin, and if the largest one corresponds to the optimum and is large in terms of percentage of the search space, the exploration power of the algorithms is not determinant, being the discriminant element their exploitation capabilities. This is what is extracted from Table 2, which shows that the DE outperforms the CMA only for the Cosine, Rastrigin and Foxholes functions, where the optimum attraction basin is very small, in particular, less than 10% of the whole landscape.

When the size of the optimum attraction basin is larger than 10% of the landscape (functions Penalized1, Penalized2, Levy, Rosenbrock, Deckers, Boha1, Boha2, Beale, Aluffi, Becker, Goldstein, Hartman3, Shekel7 and Shekel10), the behavior of the CMA improves as the exploration phase becomes less significant. Once the CMA locates the optimum attraction basin, its exploitation behavior is faster than that of the DE, obtaining better convergence speed results.

The remaining functions (Schwefel, Griewank, Hartman6, Shekel5 and Six-Hump) share the feature that the optimum attraction basin is not the largest one and there are several attraction basins in the landscape that take up more space. The Hartman6 function must be analyzed in detail, as it is the only case where the DE clearly fails in some runs. This function presents two attraction basins and the size of the optimum one is less than half the size of the other one. The convergence speed of the CMA is 25 times faster than the DE. In this case, the optimum attraction basin is large enough to be easily reached by the CMA and the DE. The CMA quickly exploits it and reaches the optimum. However, for the DE, this function is quite ambiguous. Its exploitation characteristics are not fast and precise enough not to be misled in some of the runs towards the local optimum with largest attraction basin confirming the trend indicated above: when the attraction basin of the optimum is large enough for basically any exploration strategy to place samples there, the better exploiter is the algorithm of choice, in this case, the CMA algorithm.

With the aim of performing a better analysis of the remaining two parameters, that is, variation of size of the attraction basins with distance from the optimum and distance between attraction basins, we consider the remaining four functions. Figure 1 displays their attraction basin distribution. The x-coordinate of these graphs represent the distance between the attraction basins and the optimum attraction basin whereas the y-coordinate represents the size of the attraction basins in a logarithmic scale (measured as the number of points of the search space that belong to each attraction basin given as a fraction of unity). In the first two graphs, corresponding to the Schwefel and Griewank functions,

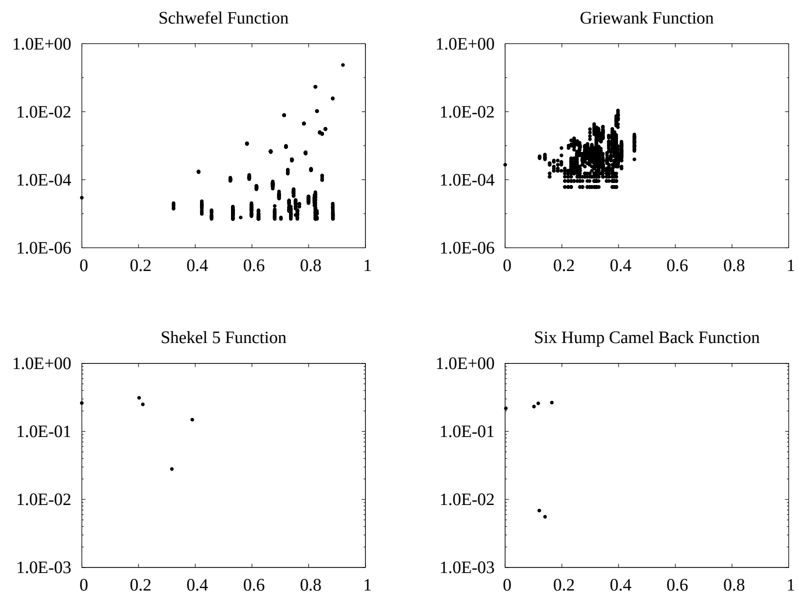

Fig. 1. Graphs of the distribution of the attraction basins for the scalable functions set. The X-axis represents the normalized distance to the optimum and the Y-axis represents the size of the attraction basins.

the size of the attraction basins increases when moving away from the optimum. This makes it more difficult to place samples in the optimum attraction basin and thus favors algorithms with better exploration characteristics. This is confirmed by the results obtained for the EAs, where the CMA was more than 70 times slower than the DE. The exploration strategy of the CMA tends towards large attraction basins, and in these cases, these attraction basins are far from the optimum.

The distance between attraction basins is also a relevant parameter as can be seen in the results for the Shekel5 and Sixhump functions which show in Fig. 1 a very similar size for all the attraction basins. However, in the Shekel5 function the distance between them is large, making it harder to "jump" from one to the next. This requires better exploration strategies (DE). On the other hand, the SixHump attraction basins are close, and even an underpowered exploration strategy allows jumping from one to the next, being again the exploitation of solutions the differentiating element in the performance of the algorithms, which favors the CMA due to its faster convergence.

4 Conclusions

In this paper we have presented an extension for real valued problems of the method proposed in [5] that allows us to characterize the topology of the fitness landscapes for optimization problems in terms of attraction basins. By considering the number of optima, the size of their attraction basins and how their size evolves when moving away from the global optimum, we have been able to understand in detail the behavior of two very well known evolutionary algorithms, Differential Evolution and Covariance Matrix Adaptation, in multimodal terms. This analysis, through the consideration of the attraction basins of the optima, in particular their sparsity and size evolution, has helped to determine some relevant characteristics that make functions harder or easier to solve by each evolutionary algorithm and provides a way to estimate which would be more adequate for each type of function.

Acknowledgments. This work was partially funded by the Xunta de Galicia through the research project 09DPI012166PR.

References

1. Ashlock, D.: Evol. Computation for Modeling and Optimization, Springer, (2006).
2. Suganthan, P. N. Hansen, N.. Liang, J. J Deb, K. Chen, Y. P. Auger, A. and Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization, Nanyang Technological University, Tech. Rep., (2005).
3. Jones, T. and Forrest, S. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Proc. 6th Int. Conf. Genetic Algorithms. L. J. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 184 – 192, (1995).
4. Manderick, B., de Weger, M. and Spiessens, P. The genetic algorithm and the structure of the fitness landscape. Proc. 4th Int. Conf. Genetic Algorithms. R. K. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 143–150, (1991).
5. Garnier, J. and Kallel, L. How to detect all maxima of a function? Proc. of the Second EVONET Summer School on Theorical Aspects of Evolutionary Computing (Anvers, 1999), Springer, Berlin, 343–370, (2001).
6. Stadler, Peter F.: Towards a theory of landscapes. In R. Lopez-Pena, R. Capovilla, R. Garcia-Pelayo, H. Waelbroeck and F. Zeturche, editors, Complex Systems and Binary Networks, volume 461, pages 77–163, Springer-Verlag, (1995).
7. Garnier, J., Kallel, L.: Statistical distribution of the convergence time of evolutionary algorithms for long-path problems, IEEE-EC, 4(1):16, (2000).
8. Yao, X. Liu, Y. and Lin, G. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, vol 3, no. 2, 82–102, (1999).
9. Dixon, L.C.W., Szeg, G.P.: The Global Optimization Problem: An Introduction, Towards Global Optimisation, North-Holland, (1978)
10. Storn, R. and Price, K. V. Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, vol. 11, 341–359, (1997).
11. Hansen, N. and Ostermeier, A. Completely derandomized self- adaptation in evolution strategies. Evolutionary Computation, vol. 9, 159–195, (2001).
12. Ronkkonen, J. Kukkonen, S. and Price, K.V., Real-parameter optimization with differential evolution Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol.1, 5, 506–513, (2005).
13. Auger, A. and Hansen, N. A restart CMA evolution strategy with increasing population size. Congress on Evolutionary Computation 2005: 1769–1776, (2005).