ASSESSMENT OF OPTIMIZATION METHODS FOR AUTOMATIC TUNING OF SEGMENTATION PARAMETERS

P. N. Happ *, R. Q. Feitosa, A. Street

Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente 225, Gávea, CEP 22451-900, Rio de Janeiro, RJ, Brazil
(patrick, raul, street)@ele.puc-rio.br

KEY WORDS: Image Segmentation, Parameter Adjustment, Optimization, Genetic Algorithm, Derivative-Free Optimization.

ABSTRACT:

The image segmentation is a key step in the image classification process since its quality will directly affects the classification result. The quality measure of image segmentation has been widely discussed in image analysis leading to the development of different metrics in order to try to automate the process and replace the subjective analysis of a specialist. These metrics are also known as similarity metrics (or functions) and evaluate the segmentation outcome comparing it with a given image containing some reference objects and returning a numerical value that express the similarity between the result and the expected references. As the quality can be expressed by a metric, the problem lies in achieving a small similarity value. This task is related to the input segmentation parameters that vary according to the image features and the classes of objects of interest. Given that the relation between the parameters and the segmentation quality can not be formulated, this procedure is generally done by a trial and error process. To avoid misleading and time consuming, automatic parameter tuning are proposed using genetic algorithms. However, this solution tends to have a high computational cost and another several parameters to tune. This work compares this solution with some derivative-free optimization methods to present some alternatives that have smaller computational cost.

1. INTRODUCTION

In geographic object-based image analysis (GEOBIA), segmentation quality is a crucial issue. For good classification accuracy the parameters of the segmentation algorithm must be properly tuned to the object classes of each target application.

However, the relation between the input parameters and the segmentation result is generally unclear. In consequence, in most applications segmentation parameters are tuned through a time consuming trial and error process.

Supervised approaches to automate this procedure have been proposed, in which the optimum setting is given by the parameter values that minimize the degree of dissimilarity between the segmentation outcome and a set of reference segments provided manually by an operator.

Conventional optimization techniques are not applicable to this task since the dissimilarity is generally not expressed by differentiable functions. Most approaches proposed to date for segmentation parameter tuning rely on stochastic techniques, mostly on genetic algorithms (GA) (Bhanu et al., 1995, Pignalberi et al., 2003, Feitosa et al., 2006). However, GAs have a number of inconveniences: are non-deterministic, involve a bunch of parameters and options whose selection is not obvious, and have a high computational cost.

Derivative-free methods have been in focus in the last few years for the optimization of black-box functions, i.e., functions for which there is no analytical formulation (Conn et al., 2009). Such methods have been successful in many applications, including the search of optimal parameter settings of different algorithms (Audet and Orban, 2006).

The objective of this work is to assess derivate-free optimization methods for the automatic adaptation of segmentation parameters in terms of accuracy and computational load. Specifically, the following methods are evaluated:

a) Generalized Pattern Search (GPS),
b) Mesh Adaptive Direct Search (MADS),
c) The Nelder-Mead (ND),

using Genetic Algorithms as benchmark. These three methods are iterative algorithms, which, after the evaluation of the objective function on a finite number of points, consider only those values to settle on the next necessary actions.

GPS (Torczon, 1997) and MADS (Audet and Dennis Jr., 2006) are said as directional direct-search methods moving towards the best point guided by sets of directions with particular features. ND (Nelder and Mead, 1965) is a direct-search algorithm based on simplex, which shifts to the opposite side from the worst point through some operations.

The remainder of this paper is organized as follows. The next section provides a description of genetic algorithms and the derivative-free methods cited above. Also, the objective function is explained. In section 3, the methodology is presented and the results of experimental analysis are discussed in section 4. Lastly, section 5 ends the work with the main conclusions and directions for future work.

2. OPTIMIZATION TECHNIQUES

This section describes succinctly the referred techniques as well as the objective function used on the optimization process.

* Corresponding author.

2.1 Genetic Algorithms

Genetic algorithms are stochastic methods for finding approximate solutions to optimization problems. Inspired by the theory of evolution by Charles Darwin (Darwin, 1859), these algorithms consist of an evolutionary process that tries out to maximize or minimize a given fitness function.

This process occurs in an iterative way through generations of individuals, where each individual can be described as a possible solution to the optimization problem. The individuals are composed by a number of genes, which represent the variables that have to be set by the optimization process.

In each generation, the existing set of individuals is called population and each individual is evaluated by calculating a given fitness function. This function is responsible for representing numerically the capability of a particular individual to solve a given problem (Michalewicz, 1996). The worst individuals are then discarded, while new ones are generated through the reproduction of the fittest individuals with the help of genetic operator.

The classical genetic operators are crossover and mutation. Crossover operators create new individuals that inherit characteristics from their parents. This occurs by mixing genes between two individuals that generally are well fitted. Mutation modifies gene values randomly so as to keep away from convergence to local minima.

In this particular case, the goal is to find the optimal values for segmentation parameters. Thus, each individual represents a set of parameter values and the fitness function is defined by a function that expresses the degree of disparity between the results produced by segmentation and a set of segments defined as a reference image.

2.2 Derivative-Free Optimization

Derivative-free optimization (DFO) methods do not require any knowledge about derivatives of the objective function, nor estimates for these derivates. Therefore, they can be used when the objective function is unknown, when it is not smooth or when it is impossible to obtain its derivatives.

These methods include the following classes of algorithms (Vaz, 2009): *Directional Direct Search*, *Simplicial Direct Search*, *Line-Search* and *Trust Region*. In this paper, the interest lies on both direct search methods: directional and simplicial. So, these methods are presented below.

2.2.1 Directional Direct Search

These methods are known for determining possible optimal points using directions (whether fixed or not) through an iterative process.

Starting from an iteration k, the next iteration will be found looking for a pattern or mesh points in some directions **d** at a distance α_k called step size. The objective is to find a new point where the objective function decreases.

The step size α_0 is a parameter that is set at the beginning of the method. If at the current iteration a better function value is found, α_k remains unchanged or is increased. Otherwise, α_k is contracted.

The directions **d** are generally given by a positive basis. For a better understanding it is important to define some concepts. A positive combination of the set of vectors $\{v_i \epsilon \mathbf{R}^n : j=1,...,r\}$ is a linear combination $a_1v_1 +\ldots+ a_rv_r$ with $a_j \geq 0$. Also, a set of vectors $\{v_i \epsilon \mathbf{R}^n : j=1,...,r\}$ is said positively independent if none of them is a positive combination of the others. Lastly, a positive basis for R^n is such that every vector in R^n can be written as a positive combination of the positive basis vectors but no member of the positive basis is expressible as a positive combination of the remaining members of the basis (Coope and Price, 2000). The minimal positive basis contains n+1 elements and the maximal contains 2n, being n the dimension of the points.

Figure 1 illustrates an example of iterations starting from x_0 and moving toward x_6. The search is performed in the directions north, south, east and west and the first point found that makes the object function decreases is selected for the next step. The process starts on x_0. Using the step size α_0, the direction to the south is tested without success. When the point at the north from x_0 is verified, the function experiences a better result and set the point to be x_1. Then, after failing with the north and south directions from x_1, x_2 is found at the east. Similarly, x_3 is defined. In this step, the search failed in all directions and the solution is to set x_4 = x_3 and to decrease the step size. The process then continues through x_5 until it finds x_6.

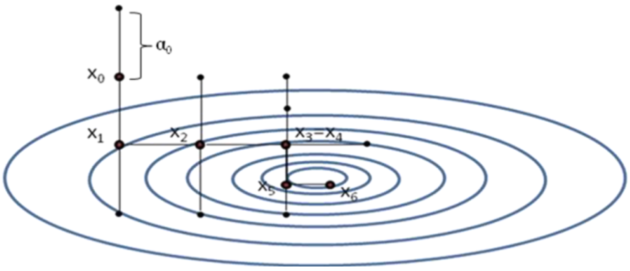

Figure 1. Example of some iterations of a directional direct search algorithm

The simplest directional method is the Coordinated Search (CS). It consists in using the maximal positive basis to proceed in the directional direct search algorithm. According to Konda et al. (2003), these methods are easy to implement and have an initially fast progress toward the solution, but may take a comparatively long time to converge.

Generalized Pattern Search (GPS) are discussed in (Audet and Dennis Jr., 2003). The key difference is the use of sets of positives bases instead of only the maximal one. Also there is an additional step called Search Step. The search step is optional and has no implications on the convergence of the method. It consists in calculating the objective function value for a finite number of points arbitrarily chosen within the provided pattern.

The Mesh Adaptive Direct Search (MADS) was proposed in (Audet and Dennis Jr., 2006). The main difference to the other methods is that the local exploration of the space of variables is not restricted to a finite number of directions which makes this class of algorithms capable of achieving convergence in the nonsmooth case.

2.2.2 Simplicial Direct Search

The central idea of this method is to build a non-degenerate simplex in $\mathbb{R}^n$ and use it to conduct the search. A simplex is a polyhedron in $\mathbb{R}^n$ with n+1 sides and it is said non-degenerate when the set $\{\mathbf{x}_1\text{-}\mathbf{x}_i, \mathbf{x}_2\text{-}\mathbf{x}_i, \ldots, \mathbf{x}_{n+1}\text{-}\mathbf{x}_i\} \forall i=1,\ldots,n$ is linearly independent.

Originally, the simplicial direct search algorithm consisted in a single movement of an isometric reflection from the worst vertex in relation to the centroid of the *n* best vertices. If the reflected vertex still is the worst vertex, then the second worst one is picked up and the process is repeated.

When the objective function decreases on the reflected vertex in relation to the best vertex, then it is considered a new candidate for the minimization problem. On the other hand, when reflection is worse than the worst vertex, there is the alternative to reduce both adjacent edges lengths from the best vertex, consequently reducing the search space.

Some new movements were then introduced: expansion and contraction (for inside or outside). Also, if these movements fail there is yet a possibility to shrink toward the best vertex. Figure 2 represents an example of these simplex operations.

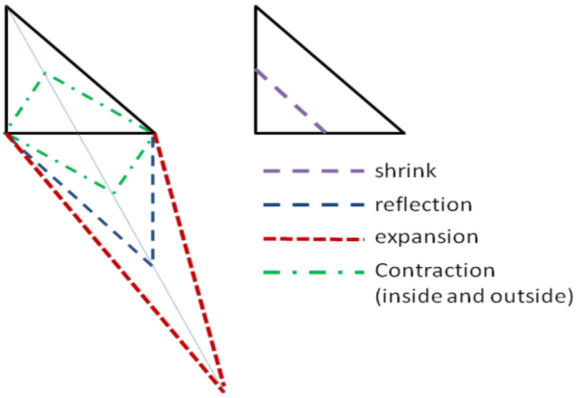

Figure 2. Movements of Nelder-Mead method

The algorithm begins with the generation of a regular simplex in $\mathbb{R}^n$ with n+1 vertices and the evaluation of the objective function. Then, the worst vertex of the simplex ($\mathbf{v}_w$) must be replaced by a new point so as to approximate to the solution. Thus, the centroid ($\mathbf{v}_c$) of the n best vertices has to be calculated.

The next step is determining the reflected vertex ($\mathbf{v}_r$). If this new point indicates a good progression then the expanded vertex ($\mathbf{v}_e$) is calculated. Then, according to the objective function, it is choose between the reflected and the expanded vertex as the substitute for the worst vertex.

Otherwise, the vertices contracted for inside and outside ($\mathbf{v}_{ci}$ and $\mathbf{v}_{co}$) are calculated and compared with the two worst vertices of the simplex. If one of the contracted vertices is better than the second worst vertex, this will replace the worst vertex in the simplex. If not, a shrink should be executed.

The shrink operation fixes the best vertex and replaces each of the other vertices by the midpoint of the segment that joins the best. After the new vertices of the simplex are established, a new iteration is then performed.

It is important to point out that this method has some convergence problems. Some variants are presented in literature. One version can be found at (Conn et al., 2009) that is based on the control of the simplex geometry, through its diameter and its normalized volume.

2.3 Objective Function

The evaluation function works comparing the segmentation result a set of parameters as input and compares its result with a given reference image that represent some objects of interest. At the end, the function returns a numerical value that represents the degree of dissimilarity between them. The intention in this case is to minimize this function in order to obtain the best set of parameters.

For a better understanding of the objective function it is worth to define some concepts, as illustrated in Figure 3. Given a set of *n* reference objects, R_i (i=1, 2, ..., *n*) is assumed to be the i-th reference object. The object resulting from the segmentation process that has the largest intercession with R_i is denoted as S_i. The number of pixels in S_i that not belong to R_i is called false positives fp_i, the number of pixels in R_i that not belong to S_i is defined as false negatives fn_i, and the number of pixels in S_i that belong to R_i is called true positives.

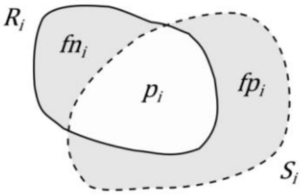

Figure 3. The reference and the segmentation object

The objective function selected for estimating the dissimilarity metric in this work is the Reference Bounded Segments Booster (RBSB) proposed in (Feitosa et al., 2006) and expressed by equation 1:

$$F(S,P) = \frac{1}{n}\sum_{i=1}^{n}\frac{fp_i + fn_i}{tp_i + fn_i} \qquad (1)$$

If the segmentation matches perfectly with the reference then the degree of dissimilarity is zero. Also, it is important to point out that identical values can be assigned to different segmentation results.

3. METHODOLOGY

A group of experiments were performed with the objective to analyse, for each method, the required number of executions of the segmentation program and the dissimilarity value reached at the end of the optimization process.

The segmentation program used in our tests is an implementation of the region growing algorithm proposed in

(Baatz and Schäpe, 2000) and the parameters being optimized were: the scale parameter, the color/shape weight, the compactness/smoothness weight, and the bands weights.

Two images have been selected to perform our tests. IM1 is a clipping from an image extracted from Google Earth (219 x 217). IM2 is a clipping from a Quickbird image (418 x 599 pixels) where the five original bands were merged into three by applying principal component analysis. These images are presented in Figure 3.

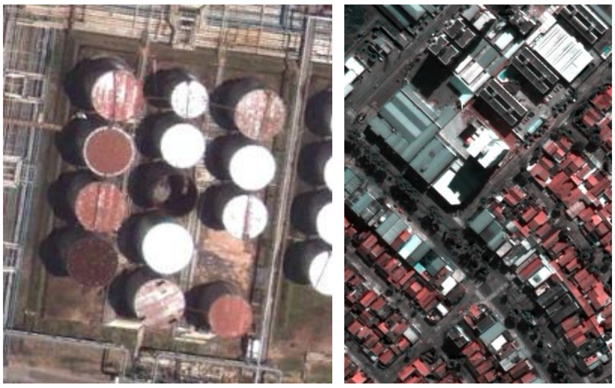

Figure 3. Images used as input in segmentation process: IM1 (left) and IM2 (right)

For IM1, five circular objects were selected to represent the reference segmentation result. For IM2, nine spectrally homogeneous objects were delineated manually to correspond to to the ideal outcome. In the first case, the spatial attributes were the distinguishing characteristics while in the last case the spectral ones. The reference objects are shown in Figure 4.

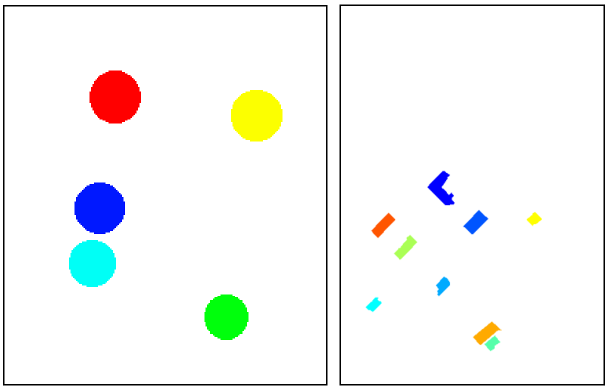

Figure 4. Reference objects for IM1 (left) and for IM2 (right)

Functions implementing the methods available in the optimization toolbox of MATLAB were used throughout our experiments. With the exception of the NM, which was executed with its default values, some parameters of the other methods were varied as described hereafter. For GPS and MADS, it was modified the number of basis vectors (which influences the quantity of search directions), the polling order (that defines the order of directions to move in the parameter space), the mesh expansion factor (that influences the step size after a successful poll) and the mesh contraction factor (that influences the step size after a failed poll). The tested configurations are presented in Table 1.

Name	Size of Basis	Polling Order	Expansion Factor	Contraction Factor
GPS1	N+1	Consecutive	2	0.5
GPS2	N+1	Success	2	0.5
GPS3	N+1	Consecutive	1	0.5
GPS4	N+1	Consecutive	2	0.25
GPS5	2N	Consecutive	2	0.5
GPS6	2N	Success	2	0.5
GPS7	2N	Consecutive	1	0.5
GPS8	2N	Consecutive	2	0.25
MADS1	N+1	Consecutive	2	0.5
MADS2	N+1	Success	2	0.5
MADS3	N+1	Consecutive	1	0.5
MADS4	N+1	Consecutive	2	0.25
MADS5	2N	Consecutive	2	0.5
MADS6	2N	Success	2	0.5
MADS7	2N	Consecutive	1	0.5
MADS8	2N	Consecutive	2	0.25

Table 1. Configurations of GPS and MADS

For GA execution, variants of a basic configuration were tested involving the selection function, the crossover function and the mutation function. The population size was also adjusted since it is directly related to the number of executions. Table 2 shows the configurations used for the experiments using GA.

Name	Population	Selection	Crossover	Mutation
GA1	8	stochastic uniform	scattered	default
GA2	8	roulette	scattered	default
GA3	8	stochastic uniform	two point	default
GA4	8	stochastic uniform	scattered	uniform
GA5	10	stochastic uniform	scattered	default
GA6	10	roulette	scattered	default
GA7	10	stochastic uniform	two point	default
GA8	10	stochastic uniform	scattered	uniform

Table 2. Configurations of GA

4. EXPERIMENTAL RESULTS

The results using IM1 are shown in Table 3. ND seems to have achieved local minima and did not deliver a good result in terms of dissimilarity. GPS reached good dissimilarity values using few executions. MADS required more evaluations, but obtained better values. GA took a higher number of executions and produced almost the same dissimilarity values as GPS when the population was set to 10.

The results using IM2 are shown in Table 4. The first important aspect revealed by Table 4 is that ND also achieved the worst dissimilarity values among all methods. Second, GA was the worst method in terms of executions of the segmentation program. On the other hand, GPS and MADS presented a good balance between both performance metrics. According to these results, GPS and MADS produced nearly the same dissimilarity values as GA, requiring roughly as little segmentation evaluations as ND.

Method Name	# Executions	Dissimilarity Value
GPS1	176	0.1419
GPS2	179	0.1419
GPS3	155	0.1191
GPS4	101	0.2034
GPS5	262	0.2808
GPS6	265	0.2808
GPS7	246	0.1191
GPS8	182	0.1199
MADS1	224	0.1248
MADS2	236	0.1315
MADS3	219	0.2362
MADS4	210	0.1343
MADS5	338	0.1241
MADS6	347	0.1244
MADS7	436	0.1117
MADS8	283	0.1563
ND	122	0.6791
GA1	416	0.2523
GA2	416	0.1910
GA3	416	0.1799
GA4	416	0.2493
GA5	520	0.1197
GA6	520	0.1299
GA7	520	0.1906
GA8	520	0.2232

Table 3. Result of the optimization methods with different configurations on image IM1

Method Name	# Executions	Dissimilarity Value
GPS1	217	0.1475
GPS2	217	0.1475
GPS3	151	0.1775
GPS4	93	0.1853
GPS5	277	0.1744
GPS6	279	0.1744
GPS7	226	0.1769
GPS8	159	0.1697
MADS1	183	0.1630
MADS2	238	0.1647
MADS3	193	0.1731
MADS4	227	0.1911
MADS5	304	0.2002
MADS6	376	0.1433
MADS7	324	0.1902
MADS8	413	0.1736
ND	180	0.4268
GA1	416	0.2602
GA2	416	0.1744
GA3	416	0.1877
GA4	416	0.2016
GA5	520	0.1526
GA6	520	0.1601
GA7	520	0.1721
GA8	520	0.2269

Table 4. Result of the optimization methods with different configurations on image IM2

It can be noticed that for both GPS and MADS, N+1 bases led to a fewer executions. On average, it also delivered in better dissimilarity values, with the exception of MADS for IM1. Likewise, it can be verified that GPS mostly presented better dissimilarity results than MADS, with the exception of MADS working with 2N bases on IM1 that achieved the better dissimilarity values.

The consecutive polling order behaved similar to the success polling order for GPS. However, it presented a substantial difference in terms of number of executions for MADS on IM2. For lower contraction factor the number of executions tended decrease for GPS. Though, this fact was not observed for MADS on IM2.

The differences in terms of dissimilarity value when the population size of the GA was varied were relatively high for IM1. Using the population size equal to eight, the best dissimilarity value were almost 0.18. For the population size set to 10, the results became closer to the GPS and MADS. Considering the different configurations for GA, the selection set to "roulette" (GA2 and GA6) was the one that reached a better average dissimilarity. The worst result was obtained for the mutation set to "uniform".

5. CONCLUSION

These results strongly suggest that the GPS and MADS can be very effective for automatic segmentation parameter tuning as alternatives to Genetic Algorithm, specially in what refers to the computational cost. Both methods achieved an overall better accuracy at less number of executions. Moreover, GA still requires a great number of parameters and options that have to be adjusted before executed. Although MADS is more frequent in the recent literature, GPS achieved better results in most of our experiments when considering the number of executions.

Analysing the dissimilarity values, the best absolute value as well as the best average value were obtained by the MADS. However, the difference is relatively small when comparing with GPS. ND did not deliver any good result.

Since GPS presented the best balance between number of executions and dissimilarity value among the tested methods, it seems to be the most recommended method for tuning segmentation parameters.

More experiments involving different images, objective functions, reference objects, as well as other derivate-free optimization methods, are planed for the continuation of this research.

ACKNOWLEDGEMENTS

The authors would like to thank to the National Council for Scientific and Technological Development (CNPq, Brazil) for financial support.

REFERENCES

Audet, C. and Dennis Jr., J. E., 2003. “Analysis of generalized pattern searches”, SIAM J. Optim., 13 (2003), pp. 889-903.

Audet, C. and Dennis Jr., J. E., 2006. “Mesh adaptive direct search algorithms for constrained optimization”. SIAM J. Optim. 17(1):188–217.

Audet, C. and Orban, D., 2006. “Finding optimal algorithmic parameters using derivative-free optimization”, SIAM J. Optim., 17 (2006), pp. 642–664.

Baatz, M. and Schäpe, A., 2000.“Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation”. In: XII Angewandte Geographische Informationsverarbeitung, Wichmann-Verlag, Heidelberg.

Bhanu, B.; Sungkee Lee; Ming, J., 1995. "Adaptive image segmentation using a genetic algorithm". In: Systems, Man and Cybernetics, IEEE Transactions on , vol.25, no.12, pp.1543-1567, Dec 1995.

Conn, A. R.; Scheinberg, K.; Vicente, L. N., 2009. “Introduction to derivative-free optimization”. Philadelphia, Society for Industrial and Applied Mathematics/Mathematical Programming Society.

Coope, I. D. and Price, C. J, 2000. Positive bases in numerical optimization. *Technical Report UCDMS2000/12*, Dept of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

Darwin, C., 1859. “On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life”. London: Murray.

Feitosa, R. Q.; Costa, G. A. O. P..; Cazes, T. B.; Feijó, B., 2006. “A genetic approach for the automatic adaptation of segmentation parameters”. In: International Conference on Object-based Image Analysis – ISPRS Proceedings, v. 36, n. 4/C42.

Kolda, T.; Lewis, R.; Torczon, V., 2003. “Optimization by direct search: New perspectives on some classical and modern methods”. SIAM Review, 45:385482.

Michalewicz, Z., 1996. “Genetic Algorithms + Data Structures = Evolution Programs”. Heidelberg: Springer.

Nelder, J. A. and Mead, R., 1965. “Simplex method for function mininimization”. Comput J., 7 (1965), pp; 308-313.

Pignalberi, G.; Cucchiara, R.; Cinque, L.; Levialdi, S., 2003. “Tuning range image segmentation by genetic algorithm”. EURASIP Journal on Applied Signal Processing, n. 8, p. 780–790.

Torczon, V., 1997. “On the Convergence of Pattern Search Algorithms” SIAM Journal on Optimization 7 :1-25.

Vaz, A., 2009. “Algoritmos Genéticos e Evolucionários e Optimização Sem Derivadas”, Curso de Optimização Aplicada às Ciências e Engenharia. Universidade do Minho, Braga.