Revista Avances en Sistemas e Informática
ISSN: 1657-7663
avances@unalmed.edu.co
Universidad Nacional de Colombia
Colombia

Velásquez Henao, Juan D.
An introduction to chaos-based algorithms for numerical optimization
Revista Avances en Sistemas e Informática, vol. 8, núm. 1, marzo, 2011, pp. 51-60
Universidad Nacional de Colombia
Medellín, Colombia

An introduction to chaos-based algorithms for numerical optimization

Una introducción a los algoritmos basados en caos para optimización numérica

Juan D. Velásquez Henao Ph.D.
Profesor, Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia
jdvelasq@unal.edu.co

Recibido para revisión 03 de septiembre de 2010, aceptado 03 de enero de 2011, versión final 09 de febrero de 2011

***Abstract*— Use of chaotic sequences as part of optimization algorithms is a novel approach to the problem. In this paper, we present, review, discuss and analyze the main chaos optimization algorithms used when the objective function is defined in a compact domain. We found that the algorithms developed in the literature are composed by one or several building blocks: the first wave carrier block, the second wave carrier block and a gradient-based optimization block. Experimentation, allow us to conclude that first wave carrier block is unnecessary, and the successful of algorithms is due to the combination of second wave carrier block and gradient based optimization.**

***Keywords*— Chaos optimization algorithms, nonlinear test functions, minimization, evaluation of algorithms.**

***Resumen*— El uso de secuencias caóticas como parte de algoritmos de optimización es una aproximación novedosa al problema. En este artículo, se revisan, presentan, discuten y analizan los principales algoritmos de optimización basados en caos cuando la función objetivo es definida en un dominio compacto. Se encontró que los algoritmos desarrollados en la literatura están compuestos por uno o más bloques constructivos: un bloque de primera onda de transporte, una segunda onda de transporte y un bloque de optimización basado en gradientes. La experimentación realizada permite concluir que el primer bloque es innecesario y que el éxito del algoritmo es debido a la combinación del bloque de la segunda ola de transporte y la optimización basada en el gradiente.**

***Palabras Clave*— Algoritmos de optimización basados en caos, funciones no lineales de prueba, minimización, evaluación de algoritmos.**

I. INTRODUCCION

In many real application cases, nonlinear optimization problems are stated as $\min f(x)$ subject to $L \leq x \leq U$, where x, L, U are vectors of $n \times 1$, and $f(\cdot)$ is a nonlinear function, such that $f: \Re^n \rightarrow \Re$. Usually, $f(\cdot)$ is a complex rough function with multiple local minima where gradient-based optimization algorithms are trapped.

Thus, heuristic algorithms based on stochastic optimization are gaining popularity [1] due to they do not use the information of the gradient of the objective function, and they have been successful for solving nonlinear complex problems; well-known techniques include simulated annealing [2], random search [3] [4] and genetic algorithms [5]. Recently, new paradigms have emerged with the promise of overcome the limitations of classical techniques; new methodologies include, among others, artificial immune systems [6] [7], the fast clonal algorithm [8] and harmony search [9].

Chaos optimization algorithms (COA) are a novel heuristic technique for global optimization developed recently [10] [11] [12], which are based in the generation of chaotic sequences of numbers instead of random number generators. However, there are not mathematical proofs about the benefits of use chaotic series [13]. There are three research trends about this topic: first, the solution of combinatorial problems, such as the traveling salesman problem, by means of a transiently neural network with chaotic annealing [14] [15]. Second, the development of COA based on the use of numerical sequences generated by

means of a chaotic map [10] [12] [16] [17] [32]. And third, the hybridization of classical optimization algorithms, including gradient-based techniques [17] [18] [19], genetic algorithms [20] [21], interior point methods [22], simulated annealing [23], particle swarm optimization [24] [25], taboo search [26] and evolutionary algorithms [27], among others. Applications of COA include mainly the training of different types of neural networks [28] [29], but they would be used for solving other continuous optimization problems.

This paper focuses on nonlinear optimization problems solved by means of COA based on the use of chaotic sequences. This paper deals specifically with methodologies based on the use of chaotic sequences for seeking the global optimum of f(x) in the compact domain defined by $L \le x \le U$. The aim of this paper is to help to identify, understand, classify and characterize the main emerging methodologies and their application to the state optimization problem. For this, we realize numerical simulations with the aim of compare the behavior of main proposed COA. Also, we present a survey of most important publications in this area. In Section 2, we describe the main COA presented in the literature and our experimental setup; in Sections 3 and 4, we present and discuss the obtained results; in Section 4, the main conclusions are summarized.

II. MATERIALS AND METHODS

A. Chaotic maps

Chaos is understood as the complex, bounded and unstable behavior caused by a simple deterministic nonlinear system or chaotic map, such that the generated sequences are quasi-random, irregular, ergodic, semi-stochastic, very sensible to the initial value [30].

The use of chaotic sequences instead of quasi-random number generators seems to be a powerful strategy for improving many traditional heuristic algorithms, and their main use is in escape of local minima points [27]. Commonly used one-dimensional non-invertible chaotic maps are presented in Table 1. In Figure 1, we present a sequence of 200 points generated by the logistic chaotic map with $\lambda = 4$ and an initial random value.

Table 1. Some common chaotic maps

Name	Equation	
Logistic	$\gamma_{n+1} = \lambda\gamma_n(1-\gamma_n)$. for $0 < \lambda \le 4$	
Tent	$\gamma_{n+1} = 2\min(\gamma_n, 1-\gamma_n)$	
Sine	$\gamma_{n+1} = \frac{a}{4}\sin(\pi\gamma_n)$ where $0 < a \le 4$	0.689067
Cubic	$\gamma_{n+1} = 3\gamma_n(1-\gamma_n^2)$	
Gauss	$\gamma_{n+1} = \frac{1}{\gamma_n} \bmod 1$	
Spence	$\gamma_{n+1} = \lvert\ln\gamma_n\rvert$	
Cusp	$\gamma_{n+1} = 1 - 2(\lvert\gamma_n\rvert)^{0.5}$	

B. Chaos optimizing algorithms

1. Basic chaos optimizing algorithm (first wave carrier)

This is the most elementary procedure; it consists in generating candidate points x_c inside of the feasible region; the optimum, x_l, is the candidate point with the lowest value of $f(x_c)$. The process is schematized in Algorithm 1. Candidate points x_c (line 03) are generated in the domain [L,U] by means of the vector of chaotic sequences γ_1. For this, each component of γ_1, $\gamma_1(i)$, is mapped linearly to the interval $[L(i),U(i)]$. In the Algorithm 1, we assume that the components of γ_1 are restricted to the interval [0, 1] such it occurs for the logistic map. At each iteration, a new vector of chaotic sequences is generated using the chaotic map $H()$ (line 7); for example, each component is generated using $\gamma_1(i)=4\gamma_1(i)[1-\gamma_1(i)]$, if $H()$ is the logistic map. Current local optimum x_l is saved in each iteration (line 6). This algorithm is the so-called first carrier wave and it is similar to the Monte Carlo optimization technique, which converges slowly and obtains the global optimum with low probability.

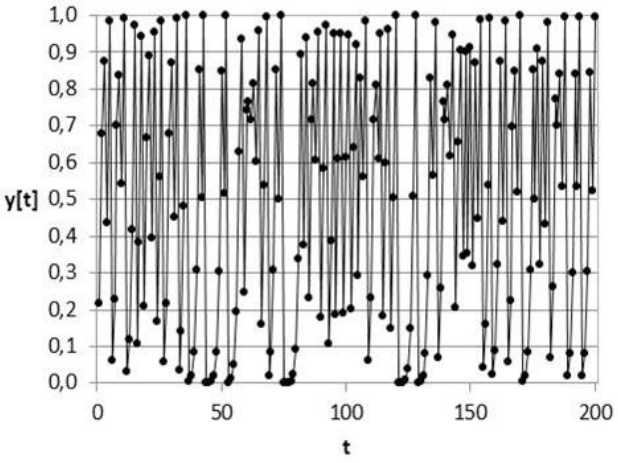

Figure 1. Chaotic sequence generated using the logistic map with $\lambda = 4$.

Algorithm 1. Basic chaos optimization algorithm – First carrier wave.

```
01  initialize γ1
02  for (m1 = 1, ..., M1) {
03      let xc = L + γ1 (U − L)
04      if (m1 == 1) let xl = xc
05      let Δf = f(xc) − f(xl)
06      if (Δf < 0) let xl = xc
07      let γ1 = H(γ1)
08  }
09  # end of algorithm
```

2. Second wave carrier

Due to the convergence problems of the Algorithm 1, it is considered a rough search procedure, and several enhancements have been proposed for improving the search procedure. An approach is to use the Algorithm 1 for obtaining the local optimum x_l, and after, to use x_l as the initial point of a local search procedure called second carrier wave [10], which is described in Algorithm 2. γ_2 is a vector of chaotic sequences, and r is a scalar parameter related to the radius of search around of x_l. For each iteration, r is decreased by means of a function $P()$ (line 6). Each candidate point is generate inside of the hypercube $[x_l - r, x_l + r]$, since each component of γ_2 (with domain [0, 1]) is mapped to the interval $[-r, r]$. The local optima is updated each time that a better point is found (line 5), such that, the procedure continues seeking around of the new optimum. The search procedure is similar is some fashion to the simulated annealing technique when ascending movements are not allowed. In [29] is suggested that $r = 0.1$ as initial value (line 1), and $P(r) = \lambda r$, with $0 < \lambda < 1$, as an useful scheme for reducing r (line 6). In addition, it is necessary to define the minimal value for r, such that, $r > 0$.

Algorithm 2. Second carrier wave.

```
01  initialize r and γ2
02  for (m2 = 1, ..., M2) {
03      let xc = xl + r(2γ2 − 1)
04      let Δf = f(xc) − f(xl)
05      if (Δf < 0) { let xl = xc }
06      let r = P(r)
07      let γ2 = H(γ2)
08  }
09  # end of algorithm
```

3. First wave carrier with gradient optimization

Other method for improving the first wave carrier (Algorithm 1) is to use a gradient based technique instead of Algorithm 2 [17]; see Algorithm 3. In addition, the complete process is repeated several times trying to escape of local optimal points (line 2). Here $\kappa()$ is a gradient-based technique for improving the current optimal point. Common choices are the weighted gradient method [31] and the BFGS algorithm [17]. For the Algorithm 3, Yang et al [17] only allows that the components of γ_1 take values in the interval $[u, 1-u]$ when the logistic map is used. Gradient-based optimization only is used when the algorithm is able to obtain a point better than the current optimal point (lines 3, 8 and 14).

4. Gradient based chaos optimization

Tavazoei & Haeri [31] combines the second carrier wave with the use of gradient-based techniques (See Algorithm 4). For the first stage, they use Algorithm 1 to select a good initial point for the local search (lines 1 to 8). For the second stage, an iterative procedure is used to alternate gradient-based optimization and second carrier wave. Thus, a new search (line 11) is made when a new optimum is found (line 178). The proposed algorithm is described in Algorithm 4.

5. Other approximations

A novel approach consists in search along each coordinate axis (as in the cyclic coordinate search algorithm), but generating random points by means of a chaotic map [32]. In each complete cycle around all components of x_l, the current optimum is refined using the BFGS algorithm. However, this approach is out the aim of this study, and it is not considered here.

Algorithm 3. Gradient based chaos optimization.

```
01  initialize γ1
02  for (c = 1, ..., C) {
03      let found = FALSE
04      for (m1 = 1, ..., M1) {
05          let xc = L + γ1(U − L)
06          if (c == 1) let xl = xc
07          let Δf = f(xc) − f(xl)
08          if (Δf < 0) {
09              let xl = xc
10              let found = TRUE
11          }
12          let γ1 = H(γ1)
13      }
14      if (found == TRUE) let xl = κ(xl) else break
15  }
16      # end of algorithm
```

Algorithm 4. Gradient based chaos optimization.

```
01  initialize γ1
02  for (m1 = 1, ..., M1) {
03      let xc = L + γ1(U − L)
04      if (m1 == 1) let xl = xc
05      let Δf = f(xc) − f(xl)
06      if (Δf < 0) let xl = xc
07      let γ1 = H(γ1)
08  }
09  let found = TRUE
10  initialize γ2
11  while (found == TRUE) {
12      let found = FALSE
13      let xl = κ(xl)
14      initialize r
15      for (m2 = 1, ..., M2) {
16          let xc = xl + r(2γ2 − 1)
17          let Δf = f(xc) − f(xl)
18          if (Δf < 0) {
```

Revista Avances en Sistemas e Informática, Vol.8 No.1, marzo de 2011 - Medellín. ISSN 1657-7663

```
19                let x_l = x_c; let found = TRUE; break }
20            let r = P(r)
21            let γ_2 = H(γ_2)
22        }
23        if (found == TRUE)  let x_l = κ(x_l)
24    }
25    # end of algorithm
```

C. *Test function*

The algorithms described here are applied to the following test functions with the aim to understand in a better way its behavior and to clarify the influence of the different building blocks:

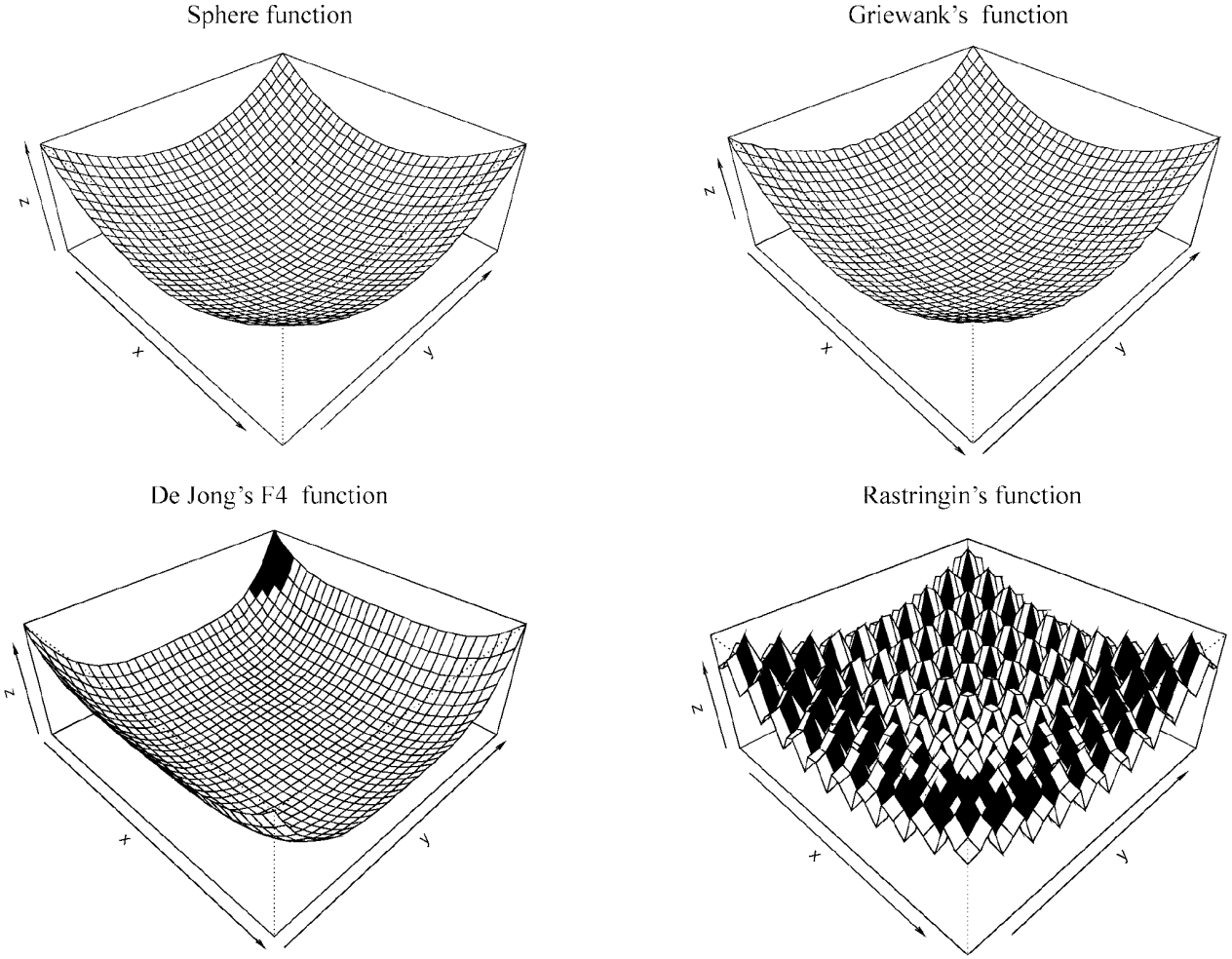

Figure 2. Plots of test functions used in this study.

Sphere function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} x_i^2$$

De Jong's F4 function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} i \cdot x_i^2$$

Griewank's function:

$$f(\mathbf{x}) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4\,000} + \prod_{i=1}^{N} \cos\frac{x_i}{\sqrt{i}}$$

Rastrigin's function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} [10 + x_i^2 - 10\cos 2\pi x_i]$$

The first two functions (Sphere and De Jong's F4) have a unique global minimum. Griewank's function has many irregularities but there is only one unique global minimum. The Rastrigin's function has many local optimal points and one unique global minimum. Table 2 resumes the global optimum, the function value at global optimum and the search range used for each test function. Figure 2 presents the plot for each test function.

III. RESULTS

For this study, N was fixed in 30 (dimensions). For each function and each algorithm considered, we use 50 random start points (50 runs); each run was limited to 15000 evaluations of the test function. In Table 3, the results for each function and each algorithm are presented.

Table 2. Test functions used in this study

Name	Global Optimum	Function value at optimum	Search range
Sphere	(0,0,…,0)	0.0	[-50,50]
DeJongF4	(0,0,…,0)	0.0	[-20,20]
Griewank	(0,0,…,0)	0.0	[-600,600]
Rastrigin	(0,0,…,0)	0.0	[-5.12,5.12]

A. Algorithm 1

In first experiment, test functions are minimized using only the first wave carrier described in Algorithm 1. Thus, candidate points are generated, using the logistic map, inside the search range. Results are reported in Table 3. The best run is plotted in Figure 3.

B. Algorithm 2

In our second experiment, we apply only the Algorithm 2 (second wave carrier) to the test functions. In this case, the initial value for r is 0.2 and we use a linear strategy to reduce its value to 0.0001 at last iteration.

C. Algorithms 1+2

In the third experiment, we combine Algorithms 1 and 2, such it is presented in the literature. We use 5000 iterations for Algorithm 1, with the aim of generates a good starting point for the Algorithm 2. 10000 iterations were used for the second wave carrier. Figure 5 presents the best run for each function.

D. Algorithm 3

In our fourth experiment, we use the Algorithm 3. For this case, we apply 1000 iterations of the first wave carrier (M_1=1000 in line 4) and then the BSFG algorithm is used to refine the current best point. The process is repeated 15 times (C=15 in lines 2). BSFG technique is applied only when the first wave carrier is able to improve the current optimal point.

Algorithm 4

In the fifth experiment, we apply Algorithm 4 to the four test functions used in this study. This algorithm combines Algorithms 1 and 2, and the BSFG gradient-based technique. The main difference with the previous experiment is the use of the second wave carrier as a mechanism to improve the current optimum point.

IV. DISCUSSION

A. Algorithm 1

It is clear that the Algorithm 1 made a rough search inside of the feasible search region; however, the generation mechanism of the next candidate point no exploits the accumulated knowledge of the function surface, and, thus, it is very difficult to obtain a better point. Numerical results in Table 3 confirms our reasoning: the best found function value is very far of the global optimal point for each function; mean best value, calculated as the mean of the best value obtain in each run, is far from the best point, and standard deviation of the sample of best points is huge. Figure 3 reveals that the method is ineffective, in the sense that it is necessary many tries to found a better point that the current. For the sphere function, Algorithm 1 was not able to improve the initial point for the best run.

Table 3. Comparison among different algorithms. All results have been averaged over 50 runs. "Best value" indicates the minimum value of the objective function over 50 runs. "Mean best value" indicates the mean of the minimum values. "Std. Dev" is the standard deviation of the minimum values.

Algorithm	Best value	Mean best value	Std. Dev.
	Sphere function		
1	11775.8433	18772.5616	1580.6817
2	0.1604	0.5462	0.1952
1+2	0.1992	0.6115	0.2263
3	8.82×10^{-39}	7.33×10^{-38}	4.14×10^{-38}
4	1.69×10^{-38}	9.65×10^{-29}	2.33×10^{-28}
	DeJongF4 function		
1	4592347.0545	8072397.2825	1342851.3951
2	0.0029	0.0449	0.0420
1+2	0.0050	0.0791	0.0860
3	2.17×10^{-10}	4.37×10^{-10}	1.62×10^{-10}
4	2.05×10^{-10}	4.20×10^{-10}	1.53×10^{-10}
	Griewank function		
1	594.8358	688.0459	41.0269
2	0.8373	0.9778	0.0522
1+2	0.8503	0.9942	0.0440
3	0.0025	0.0850	0.1519
4	0.0025	0.1724	0.2944
	Rastrigin function		
1	386.1310	428.9419	14.9182
2	67.8730	133.0823	38.6321
1+2	92.7028	191.3053	48.5231
3	110.4401	232.5600	56.4270
4	145.2634	249.3351	38.5667

Revista Avances en Sistemas e Informática, Vol.8 No.1, marzo de 2011 - Medellín. ISSN 1657-7663

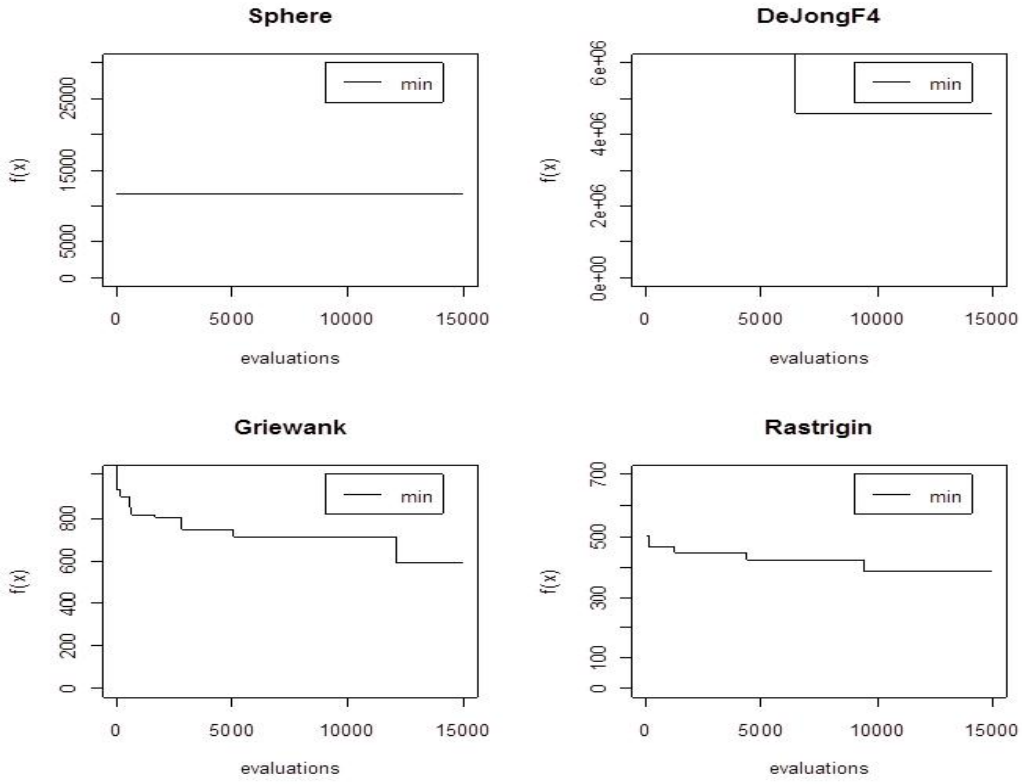

Figure 3. Best run for Algorithm 1

B. Algorithm 2

This optimization scheme remember us the simulated annealing algorithm. Algorithm 2 is a local search method where the base point for the perturbation is the best point at moment. This structure allows exploring the neighborhood of the best current point, such that, local information is used to generate the next candidate point. This scheme is more effective than the Algorithm 1; evidences presented in Table 3 support this conclusion. First, the best optimal point is far, in quality, from the obtained using only the Algorithm 1. Second, the dispersion of the best values for Algorithm 2 is very low when they are compared with Algorithm 1. Figure 4 shows a rapid convergence and a high rate of successful points improving the value of each test function.

C. Algorithms 1 + 2

It is notorious that, the first part (Algorithm 2) is not able to peak up a good initial point; and there is a vertical line indicating the starting point of the second wave carrier. Numerical evidence in Table 3, allow us to conclude that the first wave carrier it is unnecessary and to worse the performance of the algorithm. In all test functions, the best value in all 50 runs is worse that the obtained using only the Algorithm 2.

D. Algorithm 3

For the best run (Figure 6), the strategy proposed in Algorithm 3 is able to locate an optimal value in the first main cycle, except for the Griewank function where the optimum is cycle 2. Thus, the first wave carrier component is ineffective, such as it occurred in the first experiment.

For the two first functions, characterized by the presence of a unique global optimum, the gradient-based technique is responsible for the location of the global minimum independently of the behavior of the first carrier wave. For the Griewank function, the behavior of the Algorith 3 is similar to the previous functions. For the Rastrigin function, the BSFG component converge to a near optimal local point, but Algorithm 1 is very weak to find good initial points. This is evidenced in the fact that optimal point found by the algorithm 3 is worse than alternatives using Algorithm 2; in this sense, the BSFG component is used to refine the points generated in the first wave carrier.

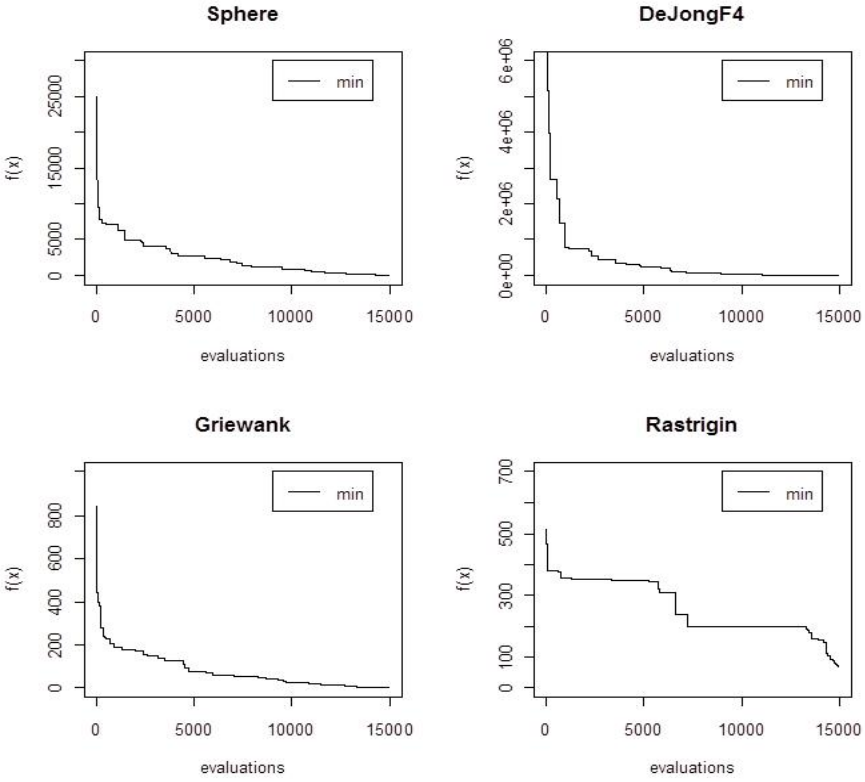

Figure 4. Best run for Algorithm 2

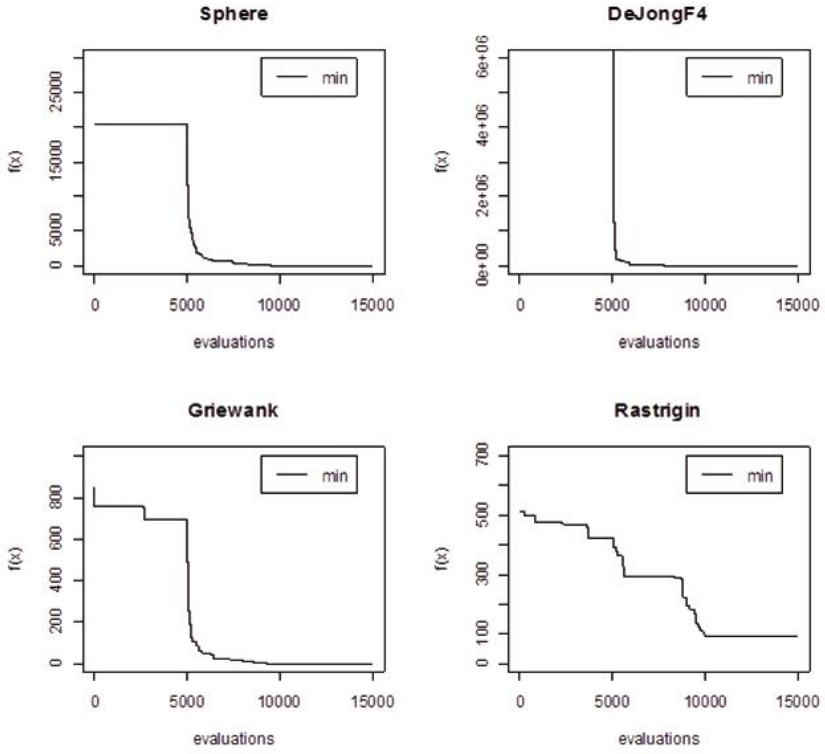

Figure 5. Best run for Algorithm 1+2

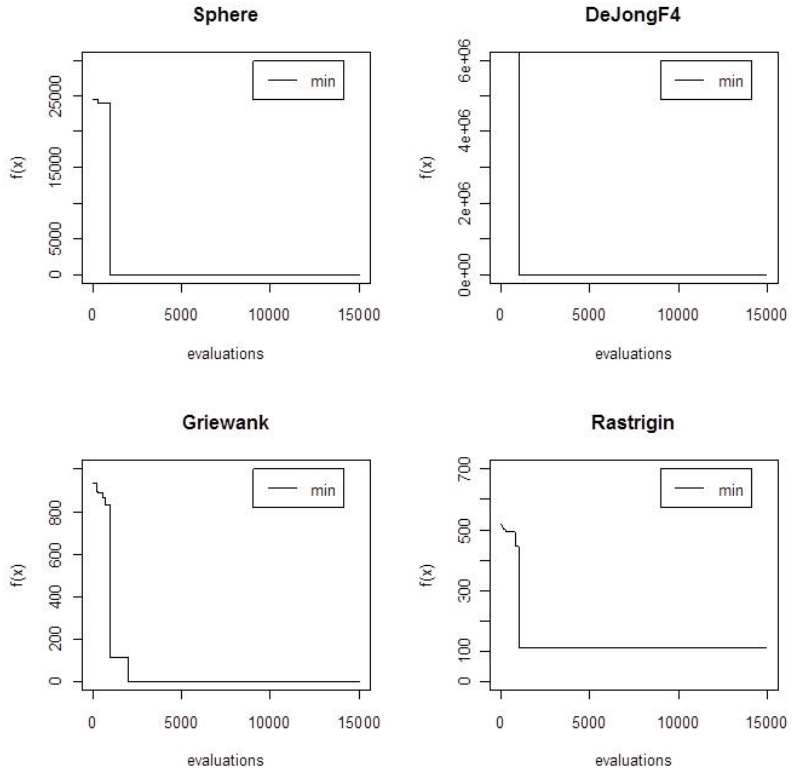

Figure 6. Best run for Algorithm 3

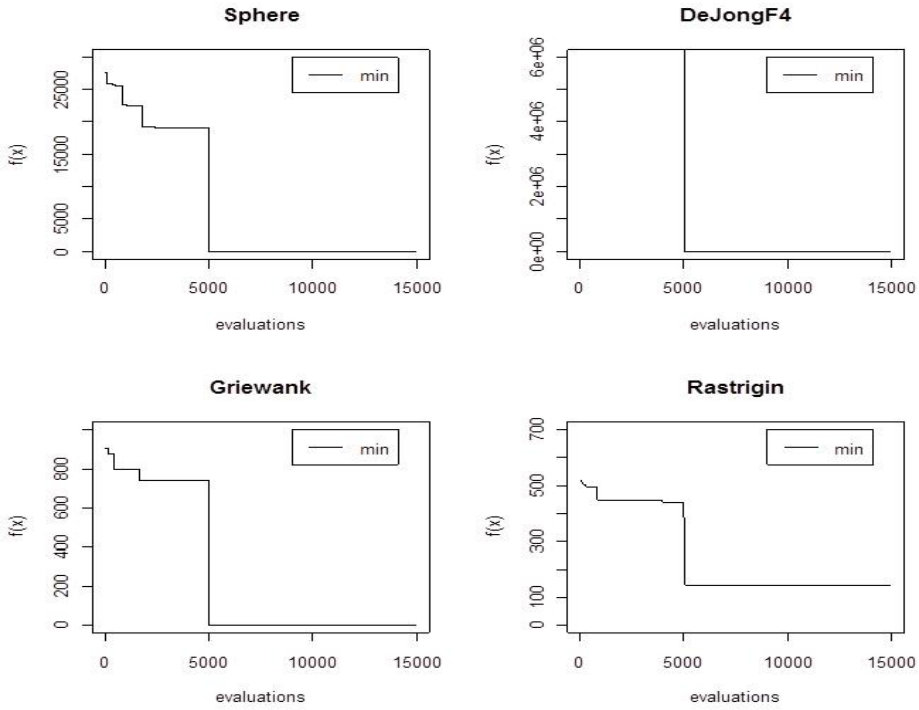

Figure 7. Best run for Algorithm 4

E. Algorithm 4

As in the previous experiment, BSFG component is responsible of the convergence of the algorithm to the global optimum for the Sphere and DeJongF4 functions (Table 3). However, Algorithm 3 is superior in terms of the mean best value and the standard deviation of the best values found. For the Rastringin function, the use of second wave carrier (Algorithm 2) is harmful, and the search capacity of the algorithm is reduced. Figure 7 shows that in any case, the second wave carrier was able to found a better point that the found in the first component of the Algorithm 4.

V. CONCLUSIONS

In this paper, we present, discuss and analyze the most important algorithms for nonlinear optimization based on chaotic sequences. Our analysis is restricted to the case when optimization is realized in a compact domain. We identify three basic building blocks: the first wave carrier block, the second wave carrier block and a gradient-based optimization block. Thus, five different algorithms are presented. Also, we use these algorithms for optimizing four complex nonlinear test functions and we discuss in detail the behavior of each algorithm. Our main conclusion is that the first wave carrier block is unnecessary, and the successful of algorithms is due to the combination of second wave carrier block and gradient based optimization.

VI. REFERENCES

[1] Pardalos, P. M. and Resende, M. G. C., 2002. Editors. Handbook of Applied Optimization. Oxford University Press, New York.

[2] Kirkpatrick, S.; Gelatt, C. D. and Vecchi, M. P., 1983. Optimization by simulated annealing. Science, Vol. 220, No. 4598, pp. 671–680.

[3] Matyas, J., 1965. Random optimization. Automation and Remote Control, Vol. 26, pp. 246–253.

[4] Solis, F.J. and Wets, J.B., 1981. Minimization by random search techniques. Mathematics of Operations Research, Vol. 6, No. 1, pp. 19–30.

[5] Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA

[6] Farmer, J.D.; Packard, N. and Perelson, A. 1986. The immune system, adaptation and machine learning. Physica D, Vol. 2, pp. 187–204.

[7] Bersini, H. Y. and Varela, F.J. ,1990. Hints for adaptive problem solving gleaned from immune networks. Parallel Problem Solving from Nature, First Workshop PPSW 1, Dortmund, FRG.

[8] Khilwani, N.; Prakash, A.; Shankar, R. and Tiwari, M.K. 2008. Fast clonal algorithm. Engineering Applications of Artificial Intelligence, Vol. 21, No. 1, pp. 106-128.

[9] Lee, K. S. and Geem, Z. W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering, Vol. 194, No. 36–38, pp. 3902–3933.

[10] Li, B. and Jiang, W.-S., 1997. Chaos optimization method and its application. Control Theory and Application, Vol. 14, No. 4, pp. 613–615.

[11] Li, B. and Jiang, W.S.,1998. Optimizing complex function by chaos search. Cybernetics and Systems, Vol. 29, No.4, pp. 409–419.

[12] Choi, C. and Lee, J.J., 1998. Chaotic local search algorithm. Artificial Life and Robotics, Vol. 2, No. 1, pp. 41–47.

[13] Bucolo, M.; Caponetto, R.; Fortuna, L.; Frasca, M. and Rizzo, A., 2002. Does chaos work better than noise?. IEEE Circuits and Systems Magazine, Vol. 2, No. 3, pp. 4–19.

[14] Chen, L. and Aihara, K., 1999. Global searching ability of chaos neural networks. IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, Vol. 46, No. 8, pp. 974–993.

[15] Tokuda, I; Aihara, K. and Nagashima, T., 1998. Adaptive annealing for chaotic optimization. Physical Review E, Vol. 58, No. 4, pp. 5157–5160.

[16] Tavazoei, M. S. and Haeri, M., 2007a. An optimization algorithm based on chaotic behavior and fractal nature. Journal of Computational and Applied Mathematics, Vol. 206, No. 2, pp. 1070–1081.

[17] Yang, D.; Li, G. and Cheng, G., 2007. On the efficiency of chaos optimization algorithms for global optimization. Chaos, Solitons and Fractals, Vol. 34, pp. 1366–1375.

[18] Liu, S.S. and Hou, Z.J., 2002. Weighted gradient direction based chaos optimization algorithm for nonlinear programming problem. In: Proceedings of the Fourth World Congress on Intelligent Control and Automation, Vol. 3, pp. 1779–1783.

[19] Fazayeli, F.; Wang, L. and Liu, W., 2008. Back-propagation with chaos. In: Proceedings of the IEEE International Conference on Neural Network and Signal Processing (ICNNSP2008), pp. 5–8.

[20] Yong, Y.; Wanxing, S. and Sunan, W., 2002. Study of chaos genetic algorithms and its application in neural networks. In: Proceedings of the IEEE Region 10 Conference on Computers, Communications, TENCON '02, Vol. 1, pp. 232-235.

[21] Qi, R.; Qian, F.; Li, S. and Wang, Z., 2006. Chaos-Genetic Algorithm for Multiobjective Optimization. Proc. Intelligent Control and Automation WCICA, Vol. 1, pp. 1563–1566.

[22] Shengsong, L.; Zhijian, H. and Min, W., 2002. A hybrid algorithm for optimal power flow using the chaos optimization and the linear interior point algorithm. Proceedings of the International Conference on Power System Technology, Vol. 2, No. 793–797.

[23] Ji, M.J. and Tang, H. W., 2004. Application of chaos in simulated annealing. Chaos, Solitons & Fractals, Vol. 21, pp. 933–941.

[24] Liu, B.; Wang, L.; Yin, H. Y.; Tang, F. and Huang, D.X., 2005. Improved particle swarm optimization combined with chaos. J Chaos, Solitons & Fractals, Vol. 25, No. 5, pp. 1261–1271.

[25] Xiang, T.; Liao, X. and Wong, K., 2007. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Applied Mathematics and Computation, Vol. 190, No. 2, pp. 1637–1645.

[26] Changzhi, S.; Zhifei, C. and Hongmei, L., 2003. Chaotic optimization and Taboo search algorithms for design of underwater thruster motor. Electrical Machines and Systems, ICEMS 2003, Sixth International Conference on, Vol. 1, No. 1, pp. 149–152.

[27] Caponetto, R.; Fortuna, L.; Fazzino, S. and Xibilia, M. G., 2003. Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation, Vol. 7, No. 3, pp. 289–304.

[28] Khoa, T. Q. D. and Nakagawa, M., 2007. Training Multilayer Neural Network by Global Chaos Optimization Algorithms. International Joint Conference on Neural Networks, IJCNN 2007, Vol. 1, pp. 136–141.
[29] Lu, H.; Zhang, H. and Ma, L., 2006. A new optimization algorithm based on chaos. Journal of Zhejiang University [Science A], Vol. 7, No. 4, pp. 539–542.
[30] Strogatz, S. H., 2000. Nonlinear dynamics and chaos. Massachussetts: Perseus Publishing.
[31] Tavazoei, M. S. and Haeri, M., 2007b. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. Applied Mathematics and Computation, Vol. 187, No. 2, pp. 1076–1085.
[32] Velásquez, J.D., 2010. An Enhanced Hybrid Chaotic Algorithm using Cyclic Coordinate Search and Gradient Techniques. Revista de Ingeniería, Universidad de los Andes. En prensa.