Nonlinear Dynamics and Systems Theory, **18** (1) (2018) 42–52

On Efficient Chaotic Optimization Algorithm Based on Partition of Data Set in Global Research Step

R. Bououden [1,2]* and M.S. Abdelouahab [2]

[1] *Department of Mathematics, Mentouri University, Constantine, Algeria.*
[2] *Laboratory of Mathematics and their Interactions, Abdelhafid Boussouf University Center, Mila, Algeria.*

Received: April 18, 2017; Revised: December 18, 2017

Abstract: The great difficulty facing the optimization algorithms is the easiness of trapping into local optima. Many researchers have benefited from the good characteristics of chaotic mappings to overcome this difficulty, but for some complex functions the problem persists. In this paper, we attempt to avoid this problem by proposing a new chaos optimization technique based on partition of data set in global research step. The numerical results show that the proposed algorithm provides the best results as compared to other ones.

Keywords: *chaos optimization; test functions; probability density function; Lozi map.*

Mathematics Subject Classification (2010): 65P20, 65Q10, 37N40, 65K10, 80M50, 37E05.

1 Introduction

Chaos theory has been successfully developed since its early years through wide applications in other sciences such as physics, mechanics, electronics, biology, economy, astronomy, meteorology, optimization, secure communication, ... etc [1–7]. As far as optimization problems of some usual functions that are continuously differentiable are concerned, some traditional optimization algorithms such as the Newton method, the gradient method and the Hessians method [8, 9] can get their global optimal points with the advantage of speed convergence and high precision. However, these traditional optimization algorithms will easily trap into local optimum when solving optimization problems of some multi-modal functions.

* Corresponding author: mailto:rabouden@yahoo.fr

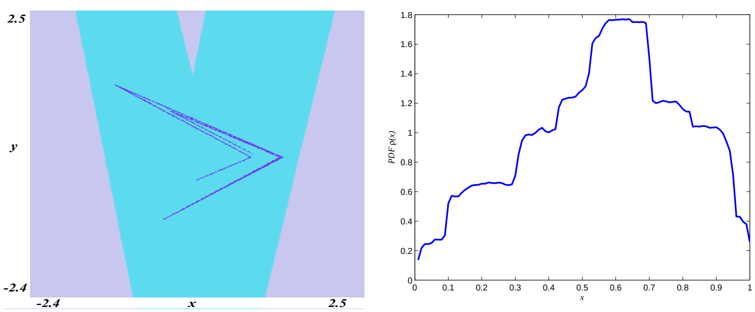
This is due to the several important dynamical characteristics of chaos, namely: the sensitive dependence on initial conditions, ergodicity, pseudo-randomness, and strange attractor with self-similar fractal pattern. Many researchers use the chaotic mappings in the optimization algorithm in order to avoid falling into local optimum [10, 11].

Recently, researchers have focused on developing the hybrid algorithms by combining heuristic algorithms with chaos searching technique to solve non linear system of equations and optimization problems such as chaotic Monte Carlo optimization, chaotic BFGS, chaotic particle swarm optimization, chaotic genetic algorithms, chaotic harmony search algorithm, chaotic simulated annealing, gradient based methods and so on [12–14].

Among those who tried to find a solution to the problem of trapping in local minima are L.S. Coelho in [15] and T. Hamaizia et al in [16]. They have resolved this problem for a large range of objective functions but for some complex functions the problem persists as we will explain later. In this paper, we recall the algorithm proposed by T. Hamaizia et al in [16] and we propose some modifications in order to improve it. The chaotic variables are generated by using the Lozi map [17] defined by the function L as follows:

$$L\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} L_1 \begin{pmatrix} x \\ y \end{pmatrix} \\ L_2 \begin{pmatrix} x \\ y \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 - a \mid x \mid + by \\ x \end{pmatrix}. \tag{1}$$

It is a $2-d$ invertible iterated map that gives a chaotic attractor called the Lozi attractor which is obtained for $a = 1.4$ and $b = 0.3$ as shown in Figure 1 (a). Numerical computation of the density $\rho(s)$ of iterated values $x(k)$ is displayed in Figure 1 (b). In this figure, the iterated values $x(k)$ are normalized in the range $[0, 1]$ i.e. $\int_0^1 \rho(x)\, \mathrm{d}x = 1$ and we notice that the highest value of $\rho(x)$ is approximately 1.8 when x is in the neighbourhood of 0.6.



(a) (b)

Figure 1: (a) Chaotic attractor of Lozi map (1) and its attractive basin obtained for $a = 1.7$ and $b = 0.5$. (b) Density of $x(k)$ in (1) over the interval $[0, 1]$ splitted into 100 boxes for 10, 000, 000 iterated values.

2 The ICOLM Algorithm

In [16] T. Hamaizia and R. Lozi have used a sampling mechanism to coordinate the research methods based on chaos theory, and they refined the final solution using a second method of local search. The obtained results show that the ICOLM algorithm is fast and converges to a good optimum compared with the COLM algorithm. But for some complex functions the problem persists. In order to avoid this problem, we will give some modifications of this method so that to improve it. We can describe this algorithm as follows:

Firstly, we choose a map and adopt it to have a chaotic behavior in order to use it to generate several sequences of points by using different initial conditions.

Secondly, every sequence $\{y(i), i = 1, 2, ..., n\}$ is normalized in the range $[0, 1]$ as follows:

$$z(i) = \frac{y(i) - \alpha}{\beta - \alpha}$$

for all $i = 1, 2, ...n$, where $\alpha = \min\{(y(i), i \geq 1\}$, $\beta = \max\{(y(i), i \geq 1\}$. The rest are:

Algorithm 2.1 Inputs:
M_g**:** max number of iterations of chaotic global search.
Mgl_1**:** max number of iterations of first chaotic local search in global search.
Mgl_2**:** max number of iterations of second chaotic local search in global search.
M_l**:** max number of iterations of chaotic local search.
$Mt = M_g(Mgl_1 + Mgl_2) + M_l$**:** stopping criterion of chaotic optimization method in iterations.
λ_{gl1}**:** step size in first global-local search.
λ_{gl2}**:** step size in second global-local search.
λ**:** step size in chaotic local search.

Outputs:
$\bar{x}$**:** best solution from current run of chaotic search.
$\bar{f}$**:** best objective function (minimization problem).

Step 1: Initialization of the numbers M_g, Mgl_1, Mgl_2, M_l of steps of chaotic search and initialization of parameters λ_{gl1}, λ_{gl1}, λ and initial conditions. Set $k = 1$, $y_1(1)$, $y_2(1)$, $a = 1.7$ and $b = 0.3$. Set the initial best objective function $\bar{f} = +\infty$.

-Step 2: Algorithm of chaotic global search:
while $k \leq M_g$ **do**
$x_i(k) = L_i + z_i(k)(U_i - L_i)$, $i = 1, 2, ..., n$
if $f(x(k)) < \bar{f}$, **then**
$\bar{x} = x(k)$, $\bar{f} = f(x(k))$
end if

-Step 2-1: Sub algorithm of first chaotic global-local search:
while $j \leq M_{gl1}$ **do**
for $i = 1$ to n **do**
if $r \leq 0.5$ **then (where** r **is a uniformly distributed random variable with range** $[0, 1]$**)**
$x_i(j) = \bar{x_i} + \lambda_{gl1} z_i(j)(U_i - \bar{x_i})$
else
$x_i(j) = \bar{x_i} - \lambda_{gl1} z_i(j)(\bar{x_i} - L_i)$
end if
end for
if $f(x(j)) < \bar{f}$, **then**
$\bar{x} = x(j)$, $\bar{f} = f(x(j))$
end if

$j = j + 1$
end while
- Step 2-2: Sub algorithm of second chaotic global-local search:
while $s \leq M_{gl2}$ **do**
for $i = 1$ to n **do**
if $r \leq 0.5$ **then**
$x_i(s) = \bar{x_i} + \lambda_{gl2} z_i(s)(U_i - \bar{x_i})$
else
$x_i(s) = \bar{x_i} - \lambda_{gl2} z_i(s)(\bar{x_i} - L_i)$
end if
end for
if $f(x(s)) < \bar{f}$, **then**
$\bar{x} = x(s),\ \bar{f} = f(x(s))$
end if
$s = s + 1$
end while
$k = k + 1$
end while
- Step 3: Algorithm of chaotic local search:
while $k \leq M_l$ **do**
for $i = 1$ to n **do**
if $r \leq 0.5$ **then**
$x_i(k) = \bar{x_i} + \lambda z_i(k)(U_i - \bar{x_i})$
else
$x_i(k) = \bar{x_i} - \lambda z_i(k)(\bar{x_i} - L_i)$
end if
end for
if $f(x(k)) < \bar{f}$, **then**
$\bar{x} = x(k),\ \bar{f} = f(x(k))$
end if
$k = k + 1$
end while.

Although this method was developed to find a solution to trapping into local optimization when solving optimization problems of some multi-modal functions, the success was partial because if the objective function is not smooth, this method will easily trap into local minima as we are going to clarify. If in step k in global search the optimal solution of our problem is $f(x^*)$, then all the points $x(s), s > k$ in the red part of Figure 2 will be ignored during the search; but it is possible that the global minima will be in the neighbourhood of one point of the red part. To solve this problem we suggest to divide the number of iterations in global search into packs and at the beginning of each pack we set the best objective function $\bar{f} = +\infty$.

On the other hand, due to the non-repetition of chaos, the chaotic research can carry out overall searches at higher speed than stochastic ergodic searches that depend on probabilities. Motivated by this idea, we will replace the step of local search (random step) by a chaotic local search as we will explain later. This is why we will call this new method Pure Chaotic Optimization Algorithm (PCOA).

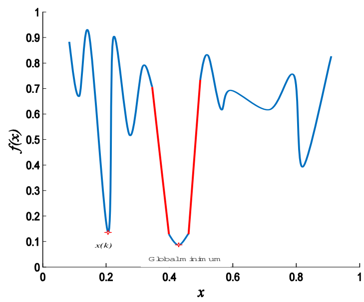


Figure 2: Example of the trapping into local minima.

3 Pure Chaotic Optimization Algorithm

As mentioned in the previous section, the fundamental changes that will be undertaken on the ICOLM are:

At first, we divide the data set that will be used in global search into packs and at the beginning of each pack we set the best objective function $\bar{f} = +\infty$ in order to go out of the local minima.

The second change is in the global local search and local search where we use chaotic search instead of random search. To apply the global local search, we use a linear transformation to project the points of chaotic sequences in the neighbourhood of the point of global search and the same idea will be used in the local search. In the following we give an example to illustrate this idea.

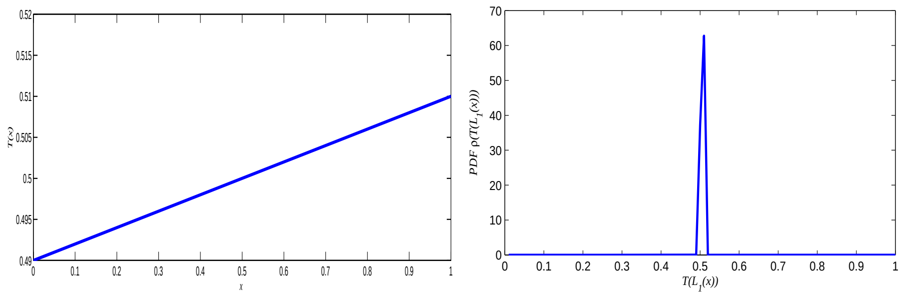
Example 3.1 In order to facilitate the process suppose that the search domain is $[l, u] = [0, 1]$ and we need to do a local search in the neighbourhood of the point $x^* = 0.5$ (i.e. the interval of local search is $[x^* - \lambda, x^* + \lambda]$, but if $x^* - \lambda < l$ (resp $x^* + \lambda > u$), the interval of local search is $[l, x^* + \lambda]$ (resp $[x^* - \lambda, u]$)). To project all the points in the neighbourhood of the point $x^* = 0.5$ we use the following linear transformation:

$$T(x) = \frac{2\lambda}{u - l} x + (x^* - \lambda).$$

Figure 3 (a) shows the plot of transformation T where we see that all the points of the interval $[l, u]$ are transformed into the interval $[x^* - \lambda, x^* + \lambda]$ $(\lambda = 0.01)$ and Figure 3 (b) shows the probability density function of $T(L_1)$.

In the following we are going to describe the pure chaotic optimization algorithm.

Algorithm 3.1 Inputs:
N**:** max number of iterations of chaotic global search.
N_p**:** max number of packets of global search.
M_g**:** max number of iterations of chaotic global search for any packets.
M_{gl}**:** max number of iterations of chaotic local search in global search.
M_l**:** max number of iterations of chaotic local search.
$Mt = N_p(M_g M_{gl} + M_l)$**:** stopping criterion of chaotic optimization method in iterations.



(a) (b)

Figure 3: (a) Transformation T. (b) Probability density function of $T(L_1)$.

λ_{gl}**:** the width of the interval in chaotic local search in global search.
λ**:** the width of the interval in chaotic local search.

Outputs:

$\bar{x}$**:** best solution from current run of chaotic search.
$\bar{f}$**:** best objective function (minimization problem).

Step 1: Initialization of the numbers M_g, M_{gl}, M_l of steps of chaotic search and initialization of parameters λ_{gl}, λ and initial conditions. The Lozi map (1) is adopted to have a chaotic behavior in order to use it for generating several sequences of points by using different initial conditions (the number of sequences is equal to dimension of the objective function) after every sequence $\{y(i), i = 1, 2, ...n\}$ is normalized in the range $[0, 1]$ as follows:

$$z(i) = \frac{y(i) - \alpha}{\beta - \alpha}$$

for all $i = 1, 2, ...n$, where $\alpha = min\{(y(i), i \geq 1\}$, $\beta = max\{(y(i), i \geq 1\}$.

-Step 2-1: Algorithm of chaotic global search:

for $t = 1 : N_p$
Set the initial best objective function $f(\bar{t}) = +\infty$.
while $k \leq M_g$ **do**
$x_i(k) = L_i + z_i(k)(U_i - L_i)$, $i = 1, 2, ..., n$
if $f(x(k)) < \bar{f}$, **then**
$\bar{x} = x(k)$, $\bar{f} = f(x(k))$

- Step 2-2: Sub algorithm of chaotic global-local search:

Transform the points generated by Lozi map in the neighbourhood of the point $\bar{x}$ and we begin the search
while $j \leq M_{gl}$ **do**
if $f(x(j)) < \bar{f}$, **then**
$\bar{x} = x(j)$, $\bar{f} = f(x(j))$
end if
$j = j + 1$
end while
end if
$k = k + 1$

end while
end for
- Step 3: Algorithm of chaotic local search:
Transform the points generated by logistic map in the neighbourhood of the point $\bar{x}$ and we begin the search
while $k \leq M_l$ **do**
if $f(x(k)) < \bar{f}$, **then**
$\bar{x} = x(k)$, $\bar{f} = f(x(k))$
end if
$k = k + 1$
end while.

During the chaotic local search, the step size λ (resp λ_{gl}) is an important parameter in convergence behavior of optimization method which adjusts small ergodic ranges around X*. The step sizes λ and λ_{gl} are employed to control the impact of the current best solution on generating a new trial solution. The small λ and λ_{gl} tend to perform exploitation to refine results by local search, while the large ones tend to facilitate a global exploration of search space.

4 Numerical Examples and Discussion

In order to test this new method vs the previous one in very tough conditions, the simulation results are obtained with the following four objective functions.

4.1 Some test functions

1. $$f_1(x_1, x_2, ..., x_n) = \frac{\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)}{2},$$
   where $-5 \leq x_i \leq 5$ for $1 \leq i \leq n$.

2. $$f_2(x_1, x_2) = x_1^4 - 7x_1^2 + x_2^4 - 9x_2^2 - 5x_2 + 11x_1^2x_2^2 + 99\sin(71x_1) + 137\sin(97x_1x_2) + 131\sin(51x_2),$$
   where $-10 \leq x_1 \leq 10$ and $-10 \leq x_2 \leq 10$.

3. $$f_3(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)],$$
   where $-2 \leq x_1 \leq 2$ and $-2 \leq x_2 \leq 2$.

4. $$f_4(x_1, x_2) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01\,|x_1 + 10|\,,$$
   where $-15 \leq x_1 \leq -5$ and $-3 \leq x_2 \leq 3$.

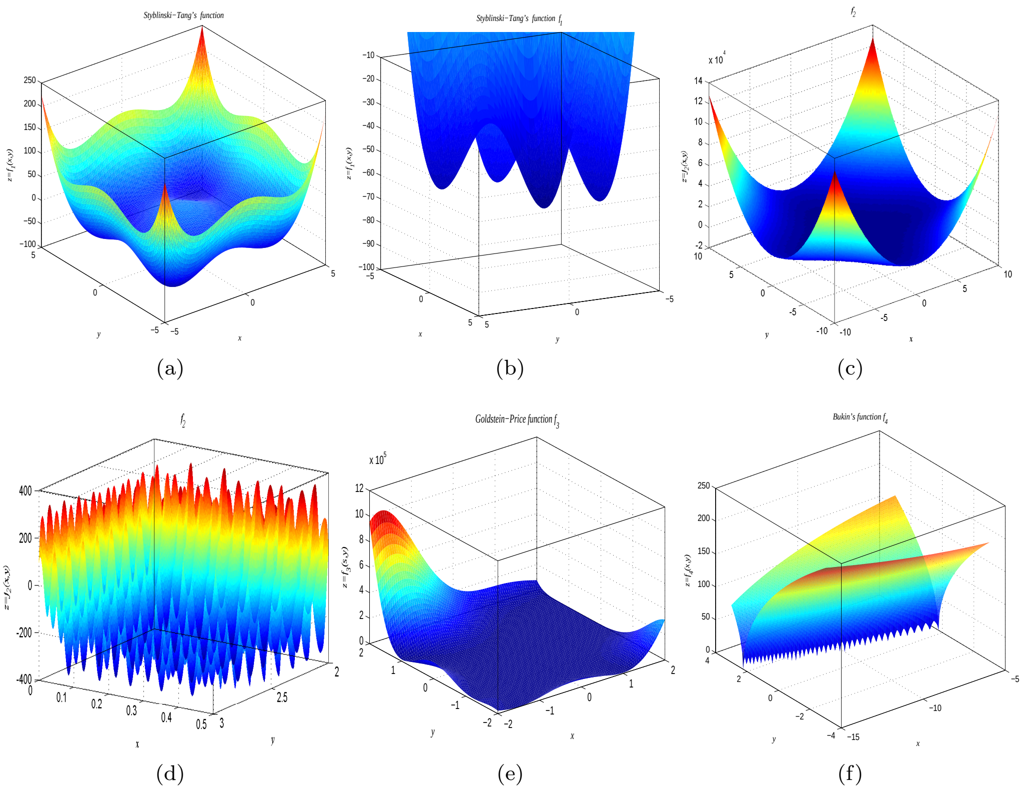


Figure 4: (a) Styblinski-Tang's function f_1. (b) Magnification of Styblinski-Tang's function f_1. (c) Function f_2. (d) Magnification of function f_2. (e) Goldstein-Price function f_3. (f) Bukin function f_4.

Figures 4 (a) and (b) show the 3D plots of the Styblinski-Tang function f_1 which is a d-dimensional function, usually evaluated on the hypercube $x_i \in [-5, 5]$, for all $i = 1, ..., d$. It has a global minimum

$$-39.16617 \times d \leq f_4(-2.903534, ..., -2.903534) \leq -39.16616 \times d.$$

Concerning f_2 shown in Figures 4 (c) and (d), it possesses hundreds of local minima [16], but its global minimum is not yet theoretically known.

f_3 is the Goldstein-Price function usually evaluated on the rectangle

$$(x_1, x_2) \in [-2, 2] \times [-2, 2],$$

it has a lot of local minima and one global minimum $f_3(0, -1) = 3$ and the 3D plot of this function is in Figure 4 (e).

f_4 is the Bukin function which is usually evaluated on the rectangle

$$(x_1, x_2) \in [-15, -5] \times [-3, 3],$$

it has a lot of local minima and one global minimum $f_4(-10, 1) = 0$, see Figure 4 (f).

4.2 Numerical experiments

In order to enrich our study, we are going to use different values of step sizes and different values of the number of iterations for both methods that are presented in Tables 1 and 2. Each optimization method was implemented in Matlab (MathWorks). All the programs were run on a 2.53 GHz, $i3$ processor with 4 GB of random access memory. Since the ICOLM algorithm gives random results, in each case study 50 independent runs are made involving 50 different initial trial conditions and all the results are summarised in Table 3; however the pure chaotic optimization algorithm is a deterministic method, therefore one run is made involving 50 different initial trial conditions and all the results are summarised in Table 4.

We generally believe that the use of large number of steps will lead us closer to the global minimum for all test functions. But this is not true as shown in Table 3 because of the trap of local minima mentioned in Section 2.

Concerning the optimization results by using the PCOA we have:

- For the function f_3 the global minimum is easily reached in few steps and little time compared with the ICOLM algorithm as explained in Tables 3 and 4.
- Concerning f_1, the global minimum is obtained by using configurations C3.
- For f_2 which possesses hundreds of local minima, the best result is obtained using configurations C3 and the global minimum is not yet theoretically known.
- Finally, the best result for f_4 is obtained using configurations C3.
  We note that the PCOA converges faster than the ICOLM as shown in Tables 3 and 4.

	λ	λ_{gl1}	λ_{gl2}	M_g	M_{gl1}	M_{gl2}	M_l
C1	0.01	0.04	0.01	30	5	5	20
C2	0.01	0.04	0.01	100	5	5	50
C3	0.001	0.04	0.01	500	10	10	100

Table 1: The set of parameter values for every run of the ICOLM algorithm.

	λ	λ_{gl}	N_p	M_g	M_{gl}	M_l
C1	0.001	0.01	100	10	100	100
C2	0.002	0.05	100	100	200	200
C3	0.005	0.08	1000	100	200	200

Table 2: The set of parameter values for every run of the PCOA algorithm.

T Fun	Case	Op So	Op Pts	Mean val	Std.Dev	T/s
f_1	C1	-103.3610	(2.7455, -2.8977,-2.9069)	-103.3383	0.0136	6.461655
	C2	-117.4956	(-2.8970,2.9005,-2.8926)	-117.4806	0.0114	20.458741
	C3	-117.4983	(-2.9046, -2.9000,-2.9038)	-117.4867	0.0082	180.311540
f_2	C1	-392.9923	(0.2443,2.0614)	-383.8462	7.6147	7.619080
	C2	-395.8094	(0.2434,2.0632)	-389.7800	5.6617	27.695888
	C3	-395.7769	(0.2434,2.0640)	-387.4540	6.1347	253.251734
f_3	C1	3.0669	(0.0108,-1.0068)	3.7525	0.2849	3.561953
	C2	3.0004	(-0.0007, -1.0010)	3.0064	0.0052	11.280039
	C3	3.0001	(0.0006, -1.0001)	3.0039	0.0026	105.905089
f_4	C1	0.1027	(-9.4415,0.8914)	0.7547	0.4245	3.562340
	C2	0.02794	(-9.4132,0.8861)	0.4295	0.1159	12.257843
	C3	0.0487	(-9.5870,0.9191)	0.3587	0.2091	109.698371

Table 3: Optimization results over 50 runs for 3 parameter configurations using ICOLM algorithm.

Test Function	Cases	Optimal solution	Optimal point	T/s
f_1	C1	-117.4772	(-2.8830, -2.8759, -2.9111)	2.648436
	C2	-117.4924	(-2.8869,-2.8949,-2.9014)	8.289850
	C3	-117.4985	(-2.9034, -2.9026,-2.8952)	47.761714
f_2	C1	-390.2672	(0.0622,1.8189)	2.237673
	C2	-395.8622	(0.2433,2.0638)	5.072239
	C3	-395.8742	(0.2432,2.0636)	49.7400
f_3	C1	3.0000	(-0.0001,-0.9999)	1.202800
	C2	3.0000	(-0.0000, -1.0000)	3.343877
	C3	3.0000	(-0.0000, -1.0000)	18.763473
f_4	C1	0.0322	(-10.7807,1.1622)	1.122277
	C2	0.0108	(-9.6809, 0.9372)	2.754711
	C3	0.0086	(-10.2723,1.0552)	25.725318

Table 4: Optimization results over one run for 3 parameter configurations using PCOA algorithm.

5 Conclusion

In this paper, we have presented a new technique of chaotic optimization algorithm inspired by ICOLM methods [16]. In order to test the numerical performance of this new technique, the four non linear multi modal benchmark functions are employed. More detailed analysis on this new technique by using other maps and testing them on a large number of test functions in higher dimension will be provided in near future.

Acknowledgment

The authors wish to thank the editor and anonymous reviewers for their valuable suggestions and comments.

References

[1] Abdelouahab, M.S. and Lozi, R. Hopf Bifurcation and Chaos in Simplest Fractional-Order Memristor-based Electrical Circuit. *IJIAM* **6** (2) (2015) 105–119.

[2] Illing, L. Digital communication using chaos and non linear dynamics. *Nonlinear Anal* **71** (12) (2009) 2958–2964.

[3] Bischi, G.I., Gardini, L. and Kopel, M. Analysis of global bifurcations in a market share attraction model. *J. Econ. Dyn. Control* **24** (5) (2000) 855–879.

[4] Lin, Q., Wong, K.W. and Chen, J. An enhanced variable-length arithmetic coding and encryption scheme using chaotic maps. *J. Syst. Softw* **86** (5) (2013) 1384–1389.

[5] Lopez-Mancilla, D. and Cruz-Hernandez, C. Output Synchronization of Chaotic Systems: Model-Matching Approach with Application to Secure Communication. *Nonlinear Dynamics and Systems Theory* **5** (2) (2005) 141–156.

[6] Khan, A. and Ahmad Bhat, M. Hybrid Projective Synchronization of Fractional Order Chaotic Systems with Fractional Order in the Interval (1,2). *Nonlinear Dynamics and Systems Theory* **16** (4) (2016) 350–365.

[7] Woon, S.F., Rehbock, V. and Loxton, R.C. Global Optimization Method for Continuous-Time Sensor Scheduling. *Nonlinear Dynamics and Systems Theory* **10** (2) (2010) 175–188.

[8] Liu, J. and Li, S. J. New hybrid conjugate gradient method for unconstrained optimization. *Appl. Math. Comput.* **245** (2014) 36–43.

[9] Chen, TW.C. and Vassiliadis, V.S. Solution of general nonlinear optimization problems using the penalty/modified barrier method with the use of exact Hessians. *Comput. Chem. Eng.* **27** (4) (2003) 501–525.

[10] Ying, L., Yun-chang, H. and Hong-duo, C. Steeped-up chaos optimization algorithm and its application. *J. System Eng.* **17** (1) (2002) 41–44.

[11] Jovanovic, V.T. Chaotic descent method and fractal conjecture. *Internat. J. Numer. Methods Eng* **48** (1) (2000) 137–152.

[12] Turgut, O.E. Hybrid Chaotic Quantum behaved Particle Swarm Optimization algorithm for thermal design of plate fin heat exchangers. *Applied Mathematical Modelling* **40** (1) (2016) 50–69.

[13] Machado, J. A.T. Optimal tuning of fractional controllers using genetic algorithms. *Nonlinear Dyn.* **62** (12) (2010) 447–452.

[14] Bunnag, D. and Sun, M. Genetic algorithm for constrained global optimization in continuous variables. *Appl. Math. Comput.* **171** (1) (2005) 604–636.

[15] Coelho, L.D.S. Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach. *Chaos, Solitons and Fractals* **39** (4) (2009) 1504–1514.

[16] Hamaizia, T., Lozi, R. and Hamri, N. Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor. *Applied Mathematics and Computation* **219** (1) (2012) 188–196.

[17] Lozi, R. Un attracteur étrange du type attracteur de Hénon. *Journal de Physique, Colloque* **39** (C5) (1978) 9–10.