Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor

Tayeb Hamaizia, René Lozi

To cite this version:
Tayeb Hamaizia, René Lozi. Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor. date. hal-00606509v2

HAL Id: hal-00606509
https://hal.archives-ouvertes.fr/hal-00606509v2
Preprint submitted on 16 Jul 2011
Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor

Tayeb Hamaizia · René Lozi

Abstract Recently, chaos theory has been used in the development of novel techniques for global optimization, and particularly, in the specification of chaos optimization algorithms (COA) based on the use of numerical sequences generated by means of chaotic map.

In this paper, we present an improved chaotic optimization algorithm using a new two-dimensional discrete multifold mapping for optimizing nonlinear functions (ICOM). The proposed method is a powerful optimization technique, which is demonstrated when three nonlinear functions of reference are minimized using the proposed technique.

Keywords Chaos optimization algorithms · Nonlinear test functions · 2-D Discrete map · multifold chaotic attractor.

1 Introduction

Application of chaos in industrial and applied problems is a very important and urgent research topic [1–5]. In particular, in the theory of control, cryptography and more recently in global optimization algorithms where introduction of chaotic numbers instead of random ones leads to better results [6]. In general, chaos has three main dynamics properties [7]: sensitive dependence on initial conditions assessed by Lyapunov exponents [8–10], stochasticity and ergodicity. Taking advantage of properties as ergodicity and stochasticity of chaos, some new algorithms called chaos optimization algorithm (COA) and hybridization with other techniques are presented in the literature: gradient-based methods [11], genetic algorithms [12, 13], particle swarm optimization [14–18], differential evolution [19, 20], clonal algorithms [21], artificial immune systems [22, 23], bee colony algorithms [24] and simulated annealing [25].

The aim of this paper is to present a new optimization algorithm chaotic based on new 2-D discrete chaotic system (map) with multifold attractor. The paper is organized as follows: in Sec. 2 we present a new strategy based on locally averaged strategy of the global search and multifold chaotic attractor, in Sec. 3 we analyse the effectiveness of the proposed algorithm on a benchmark suite of 3 well-known nonlinear test functions which are optimized. Finally, we propose a conclusion.

2 Chaotic optimization method

2.1 Multifold chaotic attractor

Since the pioneer chaotic map introduced by Hénon [26] in 1976, many other chaotic maps have been studied [27–29]. Among these maps, some display multifold patterns [30, 31]. In this paper, we have used the map recently introduced by Zeraoulia and Sprott, as a modification of Hénon map.

\[
\begin{align*}
  y_1(k) &= 1 - a(\sin y_1(k-1)) + by(k-1) \\
  y(k) &= y_1(k-1)
\end{align*}
\]

(1)

where \( k \) is the iteration number.

The essential motivation to replace the quadratic term \( x^2 \) in the Hénon map by the nonlinear term in \( \sin x \) is to develop a \( C^\infty \) mapping that is capable of generating chaotic attractors with multifolds via a period-doubling bifurcation route to chaos which has not been studied before in the literature.
The fact that this map is $C^\infty$ in some ways simplifies the study of the map and avoids some problems related to the lack of continuity or differentiability of the map. The choice of the term $\sin x$ has an important role in that it makes the solutions bounded for values of $b$ such that $|b| \leq 1$, and all values of $a$, while they are unbounded for $|b| > 1$. The chosen parameters values are $a = 4$ and $b = 0.9$ as suggested in [30]. For this values the observed attractor (see Fig. 1) belongs to the square $(y_1, y) \in [-8.588, 27.645]^2 = D \subset R^2$. However, even if for all $a$ in $R$ and $|b| < 1$ and all initial conditions $(y(0), y_1(0))$, the orbits of (1) are bounded (see cite 26, theorem 5), this map exhibits very complicated dynamical behaviors with coexisting attractors. Hence in order to choose initial conditions for ICOMM for which the attractor is observed we choose it in a subset of $D$. The optimization algorithm needs to normalize the variable $y(k)$ in the range $[0, 1]$ using the transformation

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

(2)

where $[\alpha, \beta] = [-8.588, 27.645]$

Numerical computation leads to the density $d(s)$ of iterated values of $y(k)$ displayed on Fig. 2. In this figure, the density is normalized to 1 over the whole interval $[0, 1]$ i.e.

$$\int_{0}^{1} d(s)ds = 1.$$  

Remark: Contrary to the theoretical proof that Lozi map exhibits a strange chaotic attractor [32] there is only weak numerical evidence that (1) has chaotic attractors. It is possible that Fig. 1 displays only a transient regime which leads eventually to a periodic orbit. Numerically, with any initial condition in the basin of attraction defined for $a = 4$ and $b = 0.9$ one finds a period 6 attractor when $k \geq 150,000$

![Fig. 1 Chaotic multifold attractor of the map (1) obtained for $a = 4$ and $b = 0.9$.](image)

![Fig. 2 density of iterated values of $y(k)$ of equation (1) over the interval $[0, 1]$ splitted in 100 boxes for 100,000 iterated values.](image)

$$y_1(k + 6) = y_1(k) = 10.9694028942956052.$$  

$$y_1(k + 1) = 13.0613259136267086.$$  

$$y_1(k + 2) = 8.97249334266406606.$$  

$$y_1(k + 3) = 11.0071070225514713.$$  

$$y_1(k + 4) = 13.0749780033934186.$$  

$$y_1(k + 5) = 8.95855079898761808.$$  

computation being done with double precision numbers. Again there is no proof that (1) possesses orbit shifted shadowing property as proved for generalized Lozi map [33]. However, as for optimization algorithm only few iterates (i.e. $k \leq 10,000$) are needed, property of transient regime (which is ergodic and stochastic within its range of value) worthes for ICOMM.

2.2 Locally averaged strategy

COMM is mainly COLM (chaotic optimization method based on Lozi map) defined by Coelho [34] in which Lozi map is replaced by the map of Zeraoula and Sprott. In order to improve COLM we have done [35] some modification in the global step of research. This new algorithm is called ICOLM (Improved COLM). Now we introduce the modification to COMM in order the global search converges. The ICOMM algorithm is then defined as:

Find $X$ to minimize $f(X), X = [x_1, x_2, \ldots, x_n]$ subject to $x_i \in [L_i, U_i]$.  

Where $f$ is the objective function, and $X$ is the decision solution vector consisting of $n$ variables $x_i \in R^n$ bounded by lower ($L_i$) and upper limits ($U_i$). The chaotic search procedure based on two-dimensional maps can be illustrated as
follows [34, 36, 37]:

Inputs:

- $M_g$: max number of iterations of chaotic Global search.
- $M_{gl1}$: max number of iterations of first chaotic Local search in Global search.
- $M_{gl2}$: max number of iterations of second chaotic Local search in Global search.
- $M_l$: max number of iterations of chaotic Local search.

- $M_g \times (M_{gl1} + M_{gl2}) + M_g$: stopping criterion of chaotic optimization method in iterations.
- $\lambda_{gl1}$: step size in first global-local search.
- $\lambda_{gl2}$: step size in second global-local search.
- $\lambda$: 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 $M_g$, $M_{gl1}$, $M_{gl2}$, $M_l$ of steps of chaotic search and initialization of parameters $\lambda_{gl1}$, $\lambda_{gl2}$, $\lambda$ and initial conditions. Set $k = 1$, $y(0)$, $y_1(0)$, $a = 4$ and $b = 0.9$. Set the initial best objective function $\bar{f} = +\infty$

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

  ```plaintext
  while $k \leq M_g$ do
    $x_i(k) = L_i + z_i(k).Z_i$;
    \text{if } f(X(k)) < \bar{f} \text{ then}
      \bar{X} = X(k); \bar{f} = f(x(k))
    \end{if}
  \end{while}

  - **Step 2-1**: Sub algorithm of first chaotic global-local search:

    ```plaintext
    while $j \leq M_{gl1}$ do
      for $i = 0$ to $n$ do
        if $r \leq 0.5$ then
          $x_i(j) = \bar{x}_i + \lambda_{gl1}z_i(j).Z_i$
        end if
        else
          $x_i(j) = \bar{x}_i - \lambda_{gl1}z_i(j).Z_i$
        end if
      end for
      \text{if } f(X(j)) < \bar{f} \text{ then}
        \bar{X} = X(j); \bar{f} = f(x(j))
      end if
    end while
  end while
  ```

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

  ```plaintext
  while $s \leq M_{gl2}$ do
    for $i = 0$ to $n$ do
      if $r \leq 0.5$ then
        $x_i(s) = \bar{x}_i + \lambda_{gl2}z_i(s).Z_i$
      end if
      else
        $x_i(s) = \bar{x}_i - \lambda_{gl2}z_i(s).Z_i$
      end if
    end for
  end while
  ```

- **Step 3**: Algorithm of chaotic local search:

  ```plaintext
  while $k \leq M_{gl} \times (M_{gl1} + M_{gl2}) + M_l$ do
    for $i = 0$ to $n$ do
      if $r \leq 0.5$ then
        $x_i = \bar{x}_i + \lambda z_i X_i Z_i$;
      end if
      else
        $x_i = \bar{x}_i - \lambda z_i X_i Z_i$;
      end if
    end while
  end while
  ```

- **Step 1**: Initialization of the numbers $M_g$, $M_{gl1}$, $M_{gl2}$, $M_l$ of steps of chaotic search and initialization of parameters $\lambda_{gl1}$, $\lambda_{gl2}$, $\lambda$ and initial conditions. Set $k = 1$, $y(0)$, $y_1(0)$, $a = 4$ and $b = 0.9$. Set the initial best objective function $\bar{f} = +\infty$

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

  ```plaintext
  while $k \leq M_g$ do
    $x_i(k) = L_i + z_i(k).Z_i$;
    \text{if } f(X(k)) < \bar{f} \text{ then}
      \bar{X} = X(k); \bar{f} = f(x(k))
    \end{if}
  \end{while}

- **Step 2-1**: Sub algorithm of first chaotic global-local search:

  ```plaintext
  while $j \leq M_{gl1}$ do
    for $i = 0$ to $n$ do
      if $r \leq 0.5$ then
        $x_i(j) = \bar{x}_i + \lambda_{gl1}z_i(j).Z_i$;
      end if
      else
        $x_i(j) = \bar{x}_i - \lambda_{gl1}z_i(j).Z_i$;
      end if
    end for
    \text{if } f(X(j)) < \bar{f} \text{ then}
      \bar{X} = X(j); \bar{f} = f(x(j))
    end if
  end while
  ```

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

  ```plaintext
  while $s \leq M_{gl2}$ do
    for $i = 0$ to $n$ do
      if $r \leq 0.5$ then
        $x_i(s) = \bar{x}_i + \lambda_{gl2}z_i(s).Z_i$;
      end if
      else
        $x_i(s) = \bar{x}_i - \lambda_{gl2}z_i(s).Z_i$;
      end if
    end for
  end while
  ```

3 Experiments and analysis

In this section, the benchmark suite consists in three nonlinear multimodal functions that differ in terms of various characteristics. They are used to evaluate application performance of ICOMM. To examine the effectiveness of this method involving the multifold map, we apply ICOMM for each function. We use different values of steps size $\lambda$, $\lambda_{gl1}$, and $\lambda_{gl2}$. For each trial we use 48 random initial points (48 runs); on a 3.2 GHz Pentium IV processor with 2 GB of RAM. For all the studied cases, the four configurations, numbered from C1 to C4, that are used are presented in Tab. 1.
3.1 Multimodal test functions

We test ICOMM in 2D optimization problem using:

3.1.1 Function $f_1$ (See Fig. 4)

The function $f_1$ is the Easom function [2, 5]

$$f_1 = -\cos(x_1) \cos(x_2) e^{-\left((x_1-\pi)^2 + (x_2-\pi)^2\right)}$$

its characteristics are:

- search domain: $-10 \leq x_i \leq 10, i = 1, 2$.
- number of local minima: several local minima.
- one global minimum: $\bar{x} = (\pi, \pi), f(\bar{x}) = -1$.

3.1.2 Function $f_2$ (See Fig. 5)

The function $f_2$ is the Rosenbrock’s function [2, 5]

$$f_2 = 100(x_1^2 + x_2)^2 + (1-x_1)^2$$

its characteristics are defined as follows:

- search domain: $-2.048 \leq x_i \leq 2.048, i = 1, 2$.
- number of local minima: several local minima.
- The global minimum: $\bar{x} = (0, 0), f(\bar{x}) = 0$.

3.1.3 Function $f_3$ (See Fig. 6)

The function $f_3$ is more complex [35, 38] than $f_1$ and $f_2$

$$f_3 = x_1^4 - 7x_1^2 - 3x_1 + x_1^4 - 9x_2^2 - 5x_2$$

$$+ 11x_1^2x_2^2 + 99 \sin(71x_1) + 137 \sin(97x_1x_2) + 131 \sin(51x_2)$$

- search domain: $-10 \leq x_i \leq 10, i = 1, 2$.
- number of local minima: several local minima.

The essential feature of this benchmark function is that location of minima is not symmetric. In a forthcoming paper we will extend our numerical analysis in higher dimension with an extended benchmark suite [38].

3.2 Numerical results

The numerical results are displayed in Tab. 2. For both functions $f_1$ and $f_2$ the global minimum is easily reached in few steps. Configurations C1 and C2 are fast and efficient. Concerning $f_3$ which possesses hundreds of local minima, the best results are obtained using configurations C3 and C4. The global minima is not yet theoretically known, however extended numerical computations give some clues that the values of $f_3$ found using both C3 and C4 are not far from the value of $f_3$ on the global minimum. The locally averaged strategy of ICOMM is illustrated on Fig. 7 on which the result of every step 2-2 is plotted.

Table 1 The set of parameters values for every run on the benchmark suite defined in Sec. 2.2

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\lambda_{M_{c1}}$</th>
<th>$\lambda_{M_{c2}}$</th>
<th>$M_c$</th>
<th>$M_t$</th>
<th>$M_{c1}$</th>
<th>$M_{c2}$</th>
<th>$M_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>90</td>
</tr>
<tr>
<td>C2</td>
<td>0.01</td>
<td>0.4</td>
<td>0.01</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>90</td>
</tr>
<tr>
<td>C3</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>1050</td>
</tr>
<tr>
<td>C4</td>
<td>0.001</td>
<td>0.04</td>
<td>0.01</td>
<td>200</td>
<td>100</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 3 Heuristics of locally-averaged strategy.

Fig. 4 graph of test function $f_1$ in the search domain.

Fig. 5 graph of test function $f_2$ in the search domain.

Fig. 6 graph of test function $f_3$ in the search domain.
Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor

$$f_2(x_1, x_2) = (x_1^2 + x_2)^2 + (1 - x_1)^2$$

Fig. 5 graph of test function $f_2$ in the search domain

$$f_3(x_1, x_2, x_3, x_4, x_5, x_6, \ldots) = 100 (x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + \ldots)$$

Fig. 6 graph of test function $f_3$ in the search domain

Fig. 7 Locally-averaged strategy of chaotic search. Results of every Step 2-2 for $f_3$

<table>
<thead>
<tr>
<th>$f_1$</th>
<th>Best value</th>
<th>Mean value</th>
<th>Std.Dev</th>
<th>$(x, y)$</th>
<th>T/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0000</td>
<td>-0.9999</td>
<td>0.0001</td>
<td>3.1443, 3.1443</td>
<td>1.9490</td>
<td></td>
</tr>
<tr>
<td>-1.0000</td>
<td>-0.998</td>
<td>0.0002</td>
<td>3.1458, 3.1446</td>
<td>1.9499</td>
<td></td>
</tr>
<tr>
<td>-1.0000</td>
<td>-0.9999</td>
<td>0.0001</td>
<td>3.1453, 3.1445</td>
<td>27.8084</td>
<td></td>
</tr>
<tr>
<td>-1.0000</td>
<td>-1.0000</td>
<td>0.0000</td>
<td>3.1420, 3.1420</td>
<td>55.5564</td>
<td></td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>(0.9986, 0.9978)</td>
<td>1.8380</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>(0.9988, 0.9977)</td>
<td>1.8386</td>
<td></td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>(0.9999, 0.9998)</td>
<td>25.9905</td>
<td></td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>(1.0001, 1.0002)</td>
<td>52.1532</td>
<td></td>
</tr>
<tr>
<td>$f_3$</td>
<td>-373.2600</td>
<td>-362.8730</td>
<td>5.8505</td>
<td>(-0.2926, -2.6142)</td>
<td>2.1350</td>
</tr>
<tr>
<td>-391.1240</td>
<td>-362.9798</td>
<td>10.7292</td>
<td>(-2.0556, -2.4995)</td>
<td>2.1736</td>
<td></td>
</tr>
<tr>
<td>-395.5435</td>
<td>-390.5618</td>
<td>5.1234</td>
<td>(-0.2897, -0.2786)</td>
<td>31.3117</td>
<td></td>
</tr>
<tr>
<td>-395.5870</td>
<td>-391.3068</td>
<td>4.7932</td>
<td>(-0.2034, 0.0920)</td>
<td>62.2426</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 optimization results over 48 runs for 4 parameter configurations

4 Conclusion

In this paper, we have presented a new chaotic optimization algorithm inspired by COLM methods, chaos optimization algorithms based on new 2-D discrete multifold chaotic attractor. This algorithm is tested on a benchmark suite consisting in three well know nonlinear reference functions. The presented study allows us to conclude that the proposed method is fast and converges to a good optimum. because we used a sampling mechanism to coordinate the research methods based on chaos theory, and we refined the final solution using a second method of local search. Further research is needed to gain more confidence and better understanding of the proposed methodology. The proposed algorithm has to be evaluated for a large number of test functions in higher dimension.

References