An Effective Chaotic Cultural-Based Particle Swarm Optimization for Constrained Engineering Design Problems

Yongquan Zhou, Lingzi Liu

College of Mathematics and Computer Science
Guangxi University for Nationalities, Nanning 530006, P.R. China
yongquanzhou@126.com

Keywords: cultural-based particle swarm optimization; chaotic local search strategy; constrained optimization; engineering design

Abstract. In this paper, a novel chaotic cultural-based particle swarm optimization algorithm (CCPSO) is proposed for constrained optimization problems by employing cultural-based particle swarm optimization (CPSO) algorithm and the notion of chaotic local search strategy. In the CCPSO, the shortcoming of cultural-based particle swarm optimization (CPSO) that it is easy to trap into local minimum be overcome, the chaotic local search strategy is introduced in the influence functions of cultural algorithm. Simulation results based on well-known constrained engineering design problems demonstrate the effectiveness, efficiency and robustness on initial populations of the proposed method.

1 Introduction

Generally, a constrained optimization problem can be described as follows:

Find $\mathbf{x}$ to minimize $f(\mathbf{x})$

Subject to: $g_i(\mathbf{x}) \leq 0, i = 1,2,...,n,$

$h_j(\mathbf{x}) = 0, j = 1,2,...,p,$

Where $\mathbf{x} = [x_1,x_2,...,x_d]^T$ denotes the decision solution vector, $n$ is the number of inequality constraints and $p$ is the number of equality constraints. In common practice, equality constraints $h_j(\mathbf{x}) = 0$ can be replaced by a set of inequality constraints $h_j(\mathbf{x}) \geq \delta$ and $h_j(\mathbf{x}) \geq -\delta$ ($\delta$ is a small tolerant amount). Thus, all constraints can be transformed to $N = n + 2p$ inequality constraints.

Many engineering design problems can be formulated as constrained optimization problems. As we all known, the penalty function method has been the most popular constraint-handling techniques due to its simple principle and easy implementation. However, it is often not easy to set suitable penalty factors. So, an evolutionary computation technique is proposed [1], i.e. genetic algorithm (GA) [2], co-evolutionary particle swarm optimization (CPSO) [3]. Another kind of constraint-handling techniques is inspired by multi-objective optimization techniques, i.e. a bubble-sort like algorithm [4]. Recently, Coello and Montes applied a similar feasibility-based rule to propose a multi-member evolution strategy for constrained optimization problems [5]. In their work, the CPSO algorithm better than those previously reported in the Reference for constrained engineering design problems.

In this paper, a novel chaotic cultural-based particle swarm optimization algorithm (CCPSO) is proposed for constrained optimization problems by applying cultural-based particle swarm optimization (CPSO) algorithm and employing the notion of chaotic local search strategy. Simulation results show that the CCPSO are effectiveness, efficiency and robustness.
2 Basic Algorithms

2.1 Cultural Algorithms

Cultural algorithm (CA) was proposed by Reynolds in 1994 [6]. A framework for a CA can be depicted in Fig 1. CA models two levels of evolution: the population and the belief space. Besides a population space, CA has a belief space in which the problem-solving knowledge acquired from the evolving population can be stored and integrated. All individuals in the population space evolved according to a certain rule. The accept() function is used to generate beliefs by gleaning the experience of selected individuals from the population. In return, the influence() function can make use of the problem-solving knowledge to guide the evolution of the population component. The belief space also can be improved by update() function. The generate() function is an operation function to enable the evolution of individuals in the population space. The objective() function is an objective function. The select() function is used to select part of the individual to form the next generation individuals. A pseudo-code description of CA is described as follows:

Cultural Algorithm (CA)
Begin
  t=0;
  Initialize Population POP(0);
  Initialize Belief Network BLF(0);
  Initialize Communication Channel CHL(0);
  Evaluate (POP(0));
  t=1;
  Repeat
    Communicate (POP(0), BLF(t));
    Adjust (BLF(t));
    Communicate (BLF(t), POP(t));
    Modulate Fitness (BLF(t), POP(t));
    t = t+1;
    Select POP(t) from POP(t-1);
    Evolve (POP(t));
    Evaluate (POP(t));
  Until (termination condition)
End

This article will use two of them to complete the whole solution process. Main population space and solution belief space inherit parent information respectively. The evolution of population space guided by the knowledge conserved in the belief space.
2.2 A basic PSO Algorithm

Particle swarm optimization (PSO) is first introduced in Reference [7] as a novel evolutionary computation technique. Each individual is regarded as a particle without volume and weight. The trajectory of each particle in the search space is dynamically adjusted by updating the velocity of each particle, according to its own flying experience as well as the experience of neighboring particles. The particles $x_i$ are update the velocity and position at each generation through tracking the local best historical position $p_i$ and the global best historical position $p_g$. In standard PSO algorithm, the new velocity and position of each particle is calculated as follows:

$$v_{ij}(k+1) = w v_{ij}(k) + c_1 r_1 (p_{ij} - x_{ij}(k)) + c_2 r_2 (p_g - x_{ij}(k)), j = 1, 2, ..., d,$$  

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), j = 1, 2, ..., d.$$  

Where $c_1$ and $c_2$ are two positive constants called acceleration coefficients, $w$ is called the inertia factor, $r_1$ and $r_2$ are two independent random numbers uniformly distributed in the range of $[0, 1]$.

3 A chaotic cultural-based particle swarm optimization algorithm (CPPSO)

3.1 Chaotic local search algorithm

In this section, a high precision chaotic optimization strategy is applied to the influence operation of cultural algorithm and the deficiency of traditional methods are effectively overcome which are easily being trapped in local optima. This paper uses the famous one dimensional mapping as follows:

$$x_{k+1} = \mu \cdot x_k (1 - x_k), 0 \leq x_k \leq 1$$  

(6)

Where $u$ is control parameters, $x_k$ is variable, $k = 0, 1, 2, ...$; The steps of chaotic local search are as follows:

Step 1. Let $k = 0$. The decision variable $x_j^k (x_{k+1} = \mu \cdot x_k (1 - x_k), 0 \leq x_k \leq 1)$ by type (7) is mapped into chaotic variables $cx_j^k$ between $0$ and $1$.

$$cx_j^k = \frac{x_j^k - x_{\text{min},j}}{x_j^k - x_{\text{max},j}}, j = 1, 2, ..., n.$$  

(7)

Where $x_{\text{max},j}$ and $x_{\text{min},j}$ are respectively for searching up-lower bound of the variable of the dimension $j$.

Step 2. Obtain next iterative chaotic variables $cx_j^{k+1}$ by calculating formula (6) based on $cx_j^k$.

Step 3. Make chaotic variables $cx_j^{k+1}$ transform into decision variables $x_j^{k+1}$ based on formula (8)

$$x_j^{k+1} = x_{\text{min},j} + cx_j^{k+1} (x_{\text{max},j} - x_{\text{min},j}), j = 1, 2, ..., n.$$  

(8)

Step 4. Assess properties of the new explanation according to the decision variables $x_j^{k+1} (j = 1, 2, ..., n)$.

Step 5. If the new solution should be better than the initial solution $X^{(0)} = [x_1^{(0)}, ..., x_n^{(0)}]$ or chaotic searching should have reached the default maximum iteration numbers, the new solution would use as the searching results of CLS. Otherwise, let $k = k + 1$ and return to step 2.

3.2 CCPSO Algorithm

Step 1. Initialize population individual with random method and the related parameters.

Step 2. Initialize the belief space: Using the same way to encode with the population space. Population size in the belief space is 30% of the population space. Calculating the generations of the population space and judging whether or not to make evolution of the belief space. Executing acceptance operation when it runs AcceptStep generation. Otherwise turn to Step 5.

Step 3. Update the belief space by accepting individuals using the acceptance function, and take the best individuals of the current population space to replace the worst individual. This paper adopts the evolutionary operation of particle swarm optimization to promote the evolution of the belief space.
**Step 4.** Influence operation: Execute CLS search to the better individual of belief space when PSO runs AffectStep generation every time based on the belief space, then updating its $p_i$ and $p_g$.

$$Affectstep = N_1 + Endstep - k \times Endstep \times N_2$$  \hspace{1cm} (9)

Where Endstep are the defaults maximum evolutionary generations of PSO, $k$ are the current generations of particle swarm evolution, $N_1$ and $N_2$ are respectively 15 and 100.

**Step 5.** Evaluate the objective values of all particles in the population space.

**Step 6.** For each particle, compare its current objective value with the objective value of its $p_g$. If the current value is better, then update $p_g$ and its objective value with the current position and objective value.

**Step 7.** Repeat step 4-step 5 until the termination condition is achieved.

4 Numerical example

**Example 1 Welded Beam Design Problem**

This problem was described in Reference [8]. Welded Beam construction can be show in Fig 2. There are four design variables, $h(x_1), l(x_2), t(x_3)$ and $b(x_4). A welded beam is designed for minimum cost subject to constraints on shear stress $\tau$, bending stress in the beam $\sigma$, buckling load on the bar $P_c$, end deflection of the beam $\delta$, and side constraints. The problem can be stated as follows:

![Fig 2. The welded beam used for an example 1](image)

$$\min f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2).$$ \hspace{1cm} (10)

Subject to: $g_1(x) = \tau(x) - 13600 \leq 0,$ \hspace{1cm} (11)

$g_2(x) = \sigma(x) - 30000 \leq 0,$ \hspace{1cm} (12)

$g_3(x) = x_1 - x_4 \leq 0,$ \hspace{1cm} (13)

$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x^2) - 5.0 \leq 0,$ \hspace{1cm} (14)

$g_5(x) = 0.125 - x_1 \leq 0,$ \hspace{1cm} (15)

$g_6(x) = \delta(x) - 0.25 \leq 0,$ \hspace{1cm} (16)

$g_7(x) = 6000 - P_c(x) \leq 0,$ \hspace{1cm} (17)

$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''x_2 / (2R) + (\tau'')^2},$ \hspace{1cm} (18)

$\tau' = 6000 / (\sqrt{2}x_1x_2),$ \hspace{1cm} (19)

$\tau'' = MR / J,$ \hspace{1cm} (20)
\[ M = 6000(14 + x_3 / 2), \]  
\[ R = \left(\frac{x_2^2 + (x_1 + x_3)^2}{4}\right), \]  
\[ J = 2\left\{\frac{\sqrt{2}x_1x_2}{12} + (x_1 + x_3)^3 / 4\right\}, \]  
\[ \sigma(x) = 504000 / (x_1^3 x_2^3), \]  
\[ \delta(x) = 2.1952 / (x_1^3 x_2^3), \]  
\[ P_c(x) = 4.013 \cdot E \cdot (1 - 0.0282346x_1) \cdot x_1 x_4^3 / (6L^2), \]

Where, \( 0.1 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10, 0.1 \leq x_3 \leq 10, 0.1 \leq x_4 \leq 2 \). \( P = 6000, L = 14, E = 30, G = 12 \times 10^6, \)  
\( \tau_{\text{max}} = 13,600, \) \( \sigma_{\text{max}} = 30,000, \) \( \delta_{\text{max}} = 0.25. \)

The simulated experiment was compiled in Matlab 7.6, and 30 independent runs are carried out for example 1. Where the population size \( M = 50, \) Maximum number of generations \( N = 100, \) Acceleration coefficients \( c_1 = 2, c_2 = 2.05, \) The inertia factor \( \omega \) is linearly decreases, Knowledge solutions size \( p = 0.3N ; N_1 = 15, \) \( N_2 = 100. \) This experiment used of optimal value and the average as a measure of algorithm, the simulation results are shown in the charts and tables as follows:

Fig 3. Objective values resulted by this algorithm with different generations for example 1.

<table>
<thead>
<tr>
<th>Design variables</th>
<th>Best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 (h) )</td>
<td>2.057296e-001</td>
</tr>
<tr>
<td>( x_2 (l) )</td>
<td>3.470489e+000</td>
</tr>
<tr>
<td>( x_3 (t) )</td>
<td>9.036624e+000</td>
</tr>
<tr>
<td>( x_4 (b) )</td>
<td>2.057939e-001</td>
</tr>
<tr>
<td>( g_1 (x) )</td>
<td>-1.758963e+000</td>
</tr>
<tr>
<td>( g_2 (x) )</td>
<td>-3.561581e+000</td>
</tr>
<tr>
<td>( g_3 (x) )</td>
<td>-1.608713e+000</td>
</tr>
<tr>
<td>( g_4 (x) )</td>
<td>-3.432983e+000</td>
</tr>
<tr>
<td>( g_5 (x) )</td>
<td>-8.072961e+000</td>
</tr>
<tr>
<td>( g_6 (x) )</td>
<td>-2.355403e+000</td>
</tr>
<tr>
<td>( g_7 (x) )</td>
<td>-2.425622e+000</td>
</tr>
<tr>
<td>( f(x) )</td>
<td>1.724852</td>
</tr>
</tbody>
</table>
Table 2 Statistical results of different methods for the example 1

<table>
<thead>
<tr>
<th>Methods</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>1.724852</td>
<td>1.738269</td>
<td>1.814293</td>
</tr>
<tr>
<td>Reference [9]</td>
<td>1.748309</td>
<td>1.771973</td>
<td>1.785835</td>
</tr>
<tr>
<td>Reference [2]</td>
<td>1.728226</td>
<td>1.792654</td>
<td>1.993408</td>
</tr>
</tbody>
</table>

From Table 1, it can be seen that the optimal feasible solution obtained by the proposed algorithm is better than other methods of those previously reported solutions. In addition, as shown in Table 2, the average searching quality of the proposed algorithm is superior to those of other methods, and the worst solution found by this algorithm is better than the optimal solutions reported in [9, 2, 10, and 3].

5 Conclusions

This paper proposes a CCPSO for solving constrained optimization problems. The CCPSO is developed to solve slow speed of search in the latter evolution of traditional cultural algorithm. It inducts chaotic strategy in evolution course and individuals escaping from local best solutions. The experimental results show that this algorithm is effectiveness.

Acknowledgment

This work is supported by Grants 60461001 from NSF of China and the project Supported by Grants 0832082, 0991086 from Guangxi Science Foundation.

References

Information Technology for Manufacturing Systems
10.4028/www.scientific.net/AMM.20-23

An Effective Chaotic Cultural-Based Particle Swarm Optimization for Constrained Engineering Design Problems
10.4028/www.scientific.net/AMM.20-23.64

DOI References
doi:10.1109/4235.873238
doi:10.1080/03052150410001647966