A Modified Cultural Algorithm based on Genetic Algorithm for Solving Global Optimization Problems

Deam James Azevedo da Silva  
Federal University of Pará (UFPa)  
Belém, Pa, 8617 Brazil.  
djames@ufpa.br

Roberto Célio Limão de Oliveira  
Federal University of Pará (UFPa)  
Belém, Pa, 8617 Brazil  
limao@ufpa.br

Abstract—This paper introduce one innovations into the world of multimodal function optimization: a new Cultural Algorithm (CA) based on Genetic Algorithm (GA) with a new adaptive mutation (AM-CGA). The idea is to use knowledge about local optima found during the search with other techniques utilized for escape from local minimum in multimodal functions. The search knowledge was maintained using a Cultural Algorithm structure, which is updated by behaviors of individuals and is used to actively guide the search. The test results show that the algorithm can get comparable or superior results to that of some current well-known unconstrained numerical optimization. The results also point to the potential of introducing of techniques such as Hill Climbing and Simulated Annealing for improving the search space of algorithm. Simulation results on benchmark indicate that AM-CGA improved the benchmark results.

Keywords: Global Optimization, Genetic Algorithm, Cultural Algorithm, Optimization methodologies.

I. INTRODUCTION

The concept of optimization of any task or process is directly related to their efficiency. This efficiency can be assessed in many ways, depending on the type of task to be carried out. Optimization is thus a science that is always in demand, as it is directly or indirectly related to capital and is used in all fields of applications such as civil engineering, mechanical, automobile, aviation, financial, electronics, chemical, etc. [1]

Evolutionary Computation (EC) methods have been successfully used to solve many diverse problems in search and optimization, mainly in numerical problems of global optimization.

This success is in part due to the unbiased nature of their operations, which can still perform well in situations with little or no domain knowledge [2].

Consider the global minimization problem described by Xin Yao and Yong Liu [3] for the purpose of development of new search algorithm. According to Yao and Liu, the problem can be formalized as a pair of real valued vectors \((S, f)\), where \(S \subseteq \mathbb{R}^n\) is a bounded set in \(\mathbb{R}^n\) and \(f: S \rightarrow \mathbb{R}\) is an \(n\)-dimensional real-valued function. The problem is to find a vector \(x_{\text{min}} \in S\) such that \(f(x_{\text{min}})\) is a global minimum on \(S\). More specifically, it is required to find an \(x_{\text{min}} \in S\) such that:

\[
\forall x = (x_1, x_2, \ldots, x_n) \in S: f(x_{\text{min}}) \leq f(x)
\]

Here \(n \in \mathbb{Z}\) and \(x_i \in [\text{inf}_i, \text{sup}_i]\), where \(\text{inf}\) and \(\text{sup}\) is lower and upper limits respectively.

Although people have made vast and in-depth scientific research on Global Optimization, and proposed a lot of methods and algorithms, they cannot satisfy the need of practical application. It is a timeless research subject and study directions that find a optimization algorithm that can be reliable converge at global optimization solution.

At present, the global optimization methods can be classified into two main categories: deterministic and probabilistic method [4]. Most of deterministic methods involve the application of heuristics to escape from local minima, such as trajectory methods and penalty-based methods. On one hand, these methods cannot guarantee to find the optimal solution and the search efficiency is rather lower. Probabilistic methods rely on probabilistic judge to determine whether search should depart from the neighborhood of a local minimum. Almost all the probabilistic methods can be proved that the algorithms can converged asymptotically at global optimal solution in the sense of probability. The Evolutionary Algorithms (EA) such as: Genetic Algorithms (GA) and Evolutionary Programming (EP) belong to the probabilistic methods and have been successfully applied in optimization of problems to find approximate solutions.

In this article is proposed a Cultural Algorithm (CA) based on Genetic Algorithm (GA) with a new adaptive mutation for global optimization problems “Adaptative Mutation Cultural Genetic Algorithm” (AM-CGA).

The outline of the paper is as follows. In Section 2, we introduce the main features of CAs and GAs. Section 3 and 4, points out an overview of the evolutionary techniques applied to global optimization problems, as well as the theoretical basis for technical proposal (AM-CGA algorithm). Our experimental results are shown in Section 5 such as benchmark functions used and finally we finish with some conclusions.
II. EVOLUTIONARY COMPUTATION (EC)

As part of the evolutionary computation (EC), the Genetic Algorithm (GA) is a special technique for function optimization initially introduced in the seventies by John Holland [5]. It is a powerful tool in solving search and optimization problems and originates from the idea of natural selection and natural genetic process. Since GA searches in colony and completes individual's information communication based on the operators of crossover and mutation, it has the advantage of implicit simultaneity and the search efficiency is significantly improved. The evolutionary process ends when the desired target is achieved or a preset amount of resources has been exhausted.

In the process of evolution, there are also problems, such as the global convergence. The convergence is not good enough, and sometimes is easy to trap into local minimum or maximum [5]. However, there can be considerable improvement in EC performance when knowledge that is acquired during evolution is used to bias the problem-solving process in order to identify patterns in the performance environment [6,7]. These patterns are used to influence the generation of candidate solutions, promote more instances of desirable candidates, or to reduce the number of less desirable candidates in the population.

Other important method in EC is Cultural algorithm (CA). In human societies, culture can be viewed as a vehicle for the storage of information that is potentially accessible to all members of the society, and that can be useful in guiding their problem solving activities. Cultural algorithms (CAs) have been developed in order to model the evolution of the cultural component of an evolutionary computational system over time as it accumulates experience [8]. As result, CAs can provide an explicit mechanism for global knowledge and a useful framework within which to model self-adaptation in an EC system [9,10,11]. The CAs is based on knowledge of an evolutionary system that implements a dual mechanism of inheritance. This mechanism allows the CAs explore as much microevolution as macroevolution. The microevolution is the evolution that happens in the population level. The macroevolution occurring on the culture itself, i.e. the beliefs space evolution. The belief space goal is to guide individuals in search of better regions (see Figure 1).

In the CAs the evolution occurs more quickly than in population without the mechanism of macroevolution. The characteristics and behavior of individuals are represented in the Population Space. This representation can support any population-based computational model such as Genetic Algorithms, Evolutionary Programming, Genetic Programming, Differential Evolution, the immune system, among others [12,13, 14].

![Fig. 1. Framework of Cultural Algorithms.](image1)

The communications protocols dictate the rules about individuals that can contribute to knowledge in the Belief Space (function of acceptance) and how the Belief Space will influence new individuals (Function of Influence).

The two most used ways to represent knowledge in the belief space are: Situational Knowledge and Normative Knowledge.

Situational Knowledge represents the best individuals found at certain time of evolution. According [15] it contains a number of individuals consider as set of exemplars to the rest of the population. The number of examples may vary according to the implementation, but usually is small. In Figure 2 has been an example of the structure used to represent this type of knowledge. Each individual is stored within its parameters and fitness value.

![Fig 2. Representation of Situational Knowledge.](image2)

The Situational Knowledge is updated when find the best individual of population.

Normative Knowledge represents a set of intervals that characterize the range of values given by the features that make the best solutions. These intervals are used to guide the adjustments (mutations) that occur in individuals.

The Figure 3 shows the structure used by Reynolds and his students, where are stored the minimum and maximum values on the individuals characteristics.

With these minimum values, (li) and maximum (ui), the fitness values also are stored. This value is result from the individuals that produced each extreme Li and Ui respectively [15].

![Fig 3. Representation of Normative Knowledge.](image3)
The adjustment of the range of Normative Knowledge varies according to the best individual. That is, if the individual was accepted by the acceptance function and its range is less than the range stored in the belief space, the range is adjusted, and vice versa.

The resolution of problems produces experiences from individual in the population space, which are selected to contribute to the acceptance by the belief space, where the knowledge is generalized and stored. In the initial population, the individuals are evaluated by the fitness function. Then, the information on the performance of the function is used as a basis for the production of generalizations for next generations. The experiences of the individuals selected will be used to make the necessary adjustments on the knowledge of the current belief space.

III. RELATED WORKS

In approaches about Global Optimization problem, Yong Xin Yao and Liu [16] emphasize two important characteristics of functions: unimodal and multimodal.

The unimodal functions are functions that have a minimum or maximum point, there isn’t, therefore, the presence of intermediaries points (local minimum or maximum) [17].

According to the outlines of a lower and upper range for X, it is observed that X1 is the lowest possible value achieved by function (global minimum). Already X2 is a point in the function F(X) defined as local minimum, because it does not represent the lowest possible value.

In studies conducted in the course of this work, we draw a conclusion that a Genetic Algorithm (GA) is sufficient for solving problems of unimodal functions. However, they are inefficient in several problems that have minimum or maximum local point (multimodal functions).

This is the tendency of the algorithm in approaching when a local minimum or maximum, because the chromosomes with unfavorable fitness are reject during the process of optimization. In this situation, which seems to be the best solution becomes an unsatisfactory result. It happens that such solutions tend to be closer to areas that have local minimum or maximum.

For example, in Figure 5 can be seen that chromosomes with the X-axis points in the corresponding position in the 'a' and 'b' are unfavorable to the selection process, because yours fitness value tend to increase.

In this case, their corresponding values of fitness in F(X) are greater than the amount found in local minimum.

This means that values of X that are distant from X2, are less selected. Consequently occur in a predominant population of values closer to X2, making it difficult to escape the local minimum.

Most traditional techniques to optimize the function using techniques such as "hill climbing" This technique start in a point, generated randomly, a search in an appropriate direction, according to a set distance for each step of search, until finally reaching an optimum point. Direction and distance of the step can be adjusted adaptively. The use of the technique of "hill climbing" in multimodal functions does not guarantee an optimal solution, because tends to found a local minimum and make multiple visits to other points already discovered.

Other local optimization techniques such as Nelder-Mead Simplex [18] have common characteristics with respect to the GA, such as not using successive derivatives of the objective function and work with a population of points instead of a single point.

A "simplex" is a geometrical figure consisting, in n dimensions, of n of (n +1) points: S0…Sn. If any point of a simplex is taken as the origin, the n other points are vector definitions of directions that span then-dimension vector space.

If we randomly draw as initial starting point S0, then we generate the other n points si according to the relation 

$$S_i = S_0 + \lambda E_j$$

where $E_j$ are n unit vectors, and $\lambda$ is a constant which is typically equal to one (but may be adapted to the problem characteristics).
Through a sequence of elementary geometric transformations (reflection, contraction, expansion and multi-contraction), the initial simplex moves, expands or contracts. After each transformation, the current worst vertex is replaced by a better one. The movements shown in Figure 7 are generated in accordance with the basic operations: $S_r$ (reflection), $S_e$ (expansion) and $S_c$ (contraction), where $\alpha$, $\beta$, $\gamma$ are constants:

$$
\overline{S} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} S_i
$$

- reflection: $S_r = (1 + \alpha) \overline{S} - \alpha S_n + 1$
- expansion: $S_e = \gamma S_r + (1 - \gamma) \overline{S}$
- contraction: $S_c = \beta S_{x+1} + (1 - \beta) \overline{S}$

At the beginning of the algorithm, one moves only the point of the simplex, where the objective function is worst (this point is called "high"), and one generates another point image of the worst point. This operation is the reflection. If the reflected point is better than all other points, the method expands the simplex in this direction, otherwise, if it is at least better than the worst, the algorithm performs again the reflection with the new worst point [18].

Consider $\Delta$ the variation in value of the evaluation function $f(\cdot)$ to move to a neighboring candidate solution $s'$, i.e $\Delta = f(s') - f(s)$. The method accepts the movement and the

The temperature $T$ is initially with a value, which after a fixed number of iterations (until the system reaches the thermal equilibrium at a given temperature) is gradually reduced by a cooling rate. The basic pseudo-code method is described show in Figure 8.

1. Let $X :=$ initial config
2. Let $E :=$ Eval($X$)
3. Let $i = random$ move from the moveset
4. Let $E_i :=$ Eval(move($X,i$))
5. If $E < E_i$ then
   $X := move(X,i)$
   $E := E_i$
Else with some probability,
accept the move even though things get worse.
   $X := move(X,i)$
   $E := E_i$
6. Goto 3 unless bored

Fig. 8 Simulated Annealing Algorithm.

It is important to observe the graph of function $f(x)$ as in Figure 9 to understand the SA process. Suppose you want to determine the global minimum $B$ of $f(x)$ from the point $x_0$. The SA can converge from $x_0$ to $x_1$ and then from $x_1$ to $B$ (global minimum). However, it is possible that the optimization process converge to $x_2$ and so to $A$ [21].

The algorithm starts with a high temperature $T$. So it accepts any type of configuration and measure the value of the temperature decreases, the settings that have a higher energy has decreased the probability of acceptance. Thus the process working initially with a great acceptance without limit settings, so do not tend to move only a local minimum.

An important feature of Simulated Annealing is the acceptance of settings that have higher energy, which may seem worse, or allows the acceptance of a configuration that provides a "worst" value for the objective function thus avoiding convergence to a minimum location. This acceptance is determined by a random number being controlled by a probability $P = e^{\Delta/T}$ [22].

IV. DESCRIPTION OF OUR APPROACH

The proposal presented in this work is based on a cultural algorithm used to resolve problems of global optimization.
Thus, it proposes a new adaptive mutation based on techniques used in problems of minimum and maximum point without the use of derivatives.

Unimodal functions for the traditional AC showed good results with what exists in the literature. Consequently the AC was unable to escape from local points in the functions of the modal.

Were initially reviewed some key points for implementation as the use the knowledge stored in CA to find a way to identify the exact time of occurrence of a local minimum for multimodal functions and Identification and analysis of possible techniques that could be applied to the AC, for escape from local minimum

Through various simulations and observations on the situational and normative knowledge discovered that the AC reached a point of local minimum or maximum (or even a global minimum or maximum) when the module of the difference of the average values from high and lower normative knowledge is near zero. Thus, a variable identified as probability of disturbance is increased to allow small movements around the current chromosome, even occurring in the opposite direction to the desired (based on acceptance of configurations with higher energy of the Simulated Annealing). The equations (1), (2) and (3) show the definition.

\[
\mu_L = \frac{1}{n} \sum_{i=1}^{n} L_i \quad (1)
\]

\[
\mu_U = \frac{1}{n} \sum_{i=1}^{n} U_i \quad (2)
\]

\[|\mu_U - \mu_L| \approx 0 \quad (3)
\]

In equation (1) and (2) are the fitness averages of Normative Knowledge Lower and Upper limits interval respectively, and \(L_i\) e \(U_i\) are the fitness from lower and upper limits for each variable \(i\) of the function.

Equation (3) represents the module of the difference of the average values from high and lowers normative knowledge when near zero.

The disturbances around all chromosomes occur in a range \(I \in [-1,0,1,0]\). This range is divided into \(N\) parts (NParts) that will define the size of the disturbance \(T\) in equation (4).

\[
T = \frac{(I_{SUP} - I_{INF})}{n} \quad (4)
\]

Through various tests it was defined the value of \(N\) equal to 200. Thus, the size \(T\) is equal to 0.01 as the calculation of the equation (4).

Each element \(T\) will be used as increment (+T) and decrement (-T) characterizing the distance \(D\) of the disturbance. The distance is increased the signal that determines the direction for the two sides \(\pm D\) \(\in [0.01,1.0]\). For example (see Figure 10), consider a point of chromosome \(P_0\). The value of \(D\) will determine the intensity of disturbance for both \(P_1\) and for \(P_2\).

These disturbances were inspired in the techniques of “hill climbing” and Simulated Annealing. One of the features inherited from the Simulated Annealing is the use of disturbance probability, which becomes zero in that the intervals are covered in full. This occurs after the end of the 200 cycles variations defined for this work.

![Figura 10: intensity of disturbances.](image)

The simplex of Nelder-Mead was used only for tests and studies of functions and its behaviors. The algorithm test was used with the CA in the sequence CA + Simplex. The idea was to optimize the use of Simplex, from the configurations of chromosomes give by the last generation of the CA. For convergence reference, the Simplex has escape from local minimum and maximum, but the computational cost was very high and the tests were abandoned.

In preliminary tests, it was possible to observe the behavior of disturbances carried out around the chromosomes. Thus, it was found that the use of small disturbances allows obtain the minimum global value in most executions. After preliminary tests was required some adjustments and refinements on algorithm to improve convergence for multimodal functions to escape the local minimum. For it, were defined some strategies as: in the mutation process is performed \(K\) copies of each chromosome. After each copy of chromosome receive disturbances. Finally it chooses the best chromosome disorders from \(k\) copies through the fitness evaluation.

Instead of making a mutation of a chromosome \(C\), run several different mutations in their copies. where only one will be chosen to be the best mutation. In the disturbances cycle are consider sub-mutation (Smut) interactions divided for 2 (two) directions (+/- Disturbance Direction) inside of range \(I\) that will influence the number of maximum steps as equation (5). The value of SC (StepCount) identifies the number of steps performed to complete any defined interval in MaxS (Maximum of disturbance steps around a point) in equation (6).
For $n = 10$ chromosomes ($\text{Smut}_{10}$) the number of steps (SC) for each mutation will be 5. Applying the value of NParts equal to 200 in equation (6) the result is 40. The condition $\left| \bar{C} - \bar{L} \right| \leq 0.01$ represents the difference between the normative knowledge of the upper and lower fitness that should be true when approaching zero. The smaller the difference greater is the probability of a local minimum or maximum. Then, the variable D (value of the disturbance increment) is initialized to the maximum, causing a higher noise.

V. EXPERIMENTAL RESULTS

To evaluate the results of the algorithm AM-CGA, it was used the six functions of the benchmark tests described in the work of Xin Yao and Yong Liu [23], which addresses problems of minimization and development of new algorithms for search. The six test functions are presented with numbers similar to [16, 24, 25], to facilitate comparison with previous results. Table I show three Unimodal functions: $f_1$ is the sphere function, $f_2$ is a test problem function and $f_5$ is the extended rosemback’s function. The other three are multimodal functions with many local minimum: $f_9$ Rastrigin function, $f_{10}$ is modified version of Ackley function and $f_{11}$ is Griewank function.

<table>
<thead>
<tr>
<th>Test function</th>
<th>n</th>
<th>$S$</th>
<th>$f_{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>30</td>
<td>$[-100, 100]$</td>
<td>0</td>
</tr>
<tr>
<td>$f_2(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>+ \prod_{i=1}^{n}</td>
<td>x_i</td>
</tr>
<tr>
<td>$f_5(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$</td>
<td>30</td>
<td>$[-30, 30]$</td>
<td>0</td>
</tr>
<tr>
<td>$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$</td>
<td>30</td>
<td>$[-5.12, 5.12]$</td>
<td>0</td>
</tr>
<tr>
<td>$f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$</td>
<td>30</td>
<td>$[-32, 32]$</td>
<td>0</td>
</tr>
<tr>
<td>$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$</td>
<td>30</td>
<td>$[-600, 600]$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table II shows the average of evolutionary processes for the six benchmark functions for the proposed algorithm (AM-CGA) and other algorithms. For the results were run 50 executions to test each function in accordance with the definitions in [23].

<table>
<thead>
<tr>
<th>Func. eval</th>
<th>AM-CGA Mean best</th>
<th>DLB2 Mean best</th>
<th>DLB1 Mean best</th>
<th>FBL Mean best</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 150.000</td>
<td>0.0</td>
<td>0.0</td>
<td>9.22E-30</td>
<td>5.85E-7</td>
</tr>
<tr>
<td>F2 200.000</td>
<td>0.0</td>
<td>4.52E-26</td>
<td>9.13E-21</td>
<td>2.30E-3</td>
</tr>
<tr>
<td>F5 2000.000</td>
<td>1.097727E-26</td>
<td>8.12E-4</td>
<td>1.61E-1</td>
<td>5.57E-1</td>
</tr>
<tr>
<td>F9 500.000</td>
<td>0.0</td>
<td>29.33</td>
<td>21.03</td>
<td>2.89</td>
</tr>
<tr>
<td>F10 150.000</td>
<td>0.0</td>
<td>7.69E-15</td>
<td>1.03E-14</td>
<td>6.33E-4</td>
</tr>
<tr>
<td>F11 200.000</td>
<td>0.0</td>
<td>1.24E-2</td>
<td>9.60E-3</td>
<td>1.27E-1</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this paper we have extended a Genetic Algorithm (GA) to Cultural Genetic Algorithm (CGA). Also was showed Cultural Algorithm in the problem of global optimization in order to improve the search performance.

We observed that the inclusion of Cultural Algorithm (CA) in canonical GA (CGA) improve the convergence reliability and search speed. However, the CGA is not enough to reach all global optimum for problems presented. It became necessary to add an adaptive solution, inspired by other techniques known to escape from local minimum or maximum. The adjustments necessary to achieve minimum or maximum points of functions are performed in the neighbors of chromosomes. Meanwhile the area of belief space records best changes made on the chromosome. This allowed to the AM-CGA provide a best performance in the benchmark tests.

The experiments suggest that systems that use cultural knowledge become very important in areas where there are limitations on system performance. In the future we will
work to investigate the performance of cultural algorithms in conjunction with other optimization techniques to evaluate their performance in more complex areas of the real world.

VII. REFERENCES


