GA and ACO-based Hybrid Approach for Continuous Optimization

Zhiqiang Chen*
Chongqing Engineering Laboratory for Detection Control and Integrated System, School of Computer Science and Information Engineering
Chongqing Technology and Business University
19, Xuefu Avenue, Nan’an, Chongqing, 400067, China

Ronglong Wang
Country Graduate School of Engineering University of Fukui
Fukui-shi, Japan 910-8507

Abstract—This paper presents a hybrid algorithm based on genetic algorithm and ant colony optimization for continuous optimization, which combines the global exploration ability of the former with the local exploiting ability of the later. The proposed algorithm is evaluated on several benchmark functions. The simulation results show that the proposed algorithm performs quite well and outperforms classical ant colony optimization and genetic algorithm for continuous optimization, which efficiently balances two contradictory aspects of its performance: exploration and exploitation.

Keywords-component; hybrid; GA; ACO; continuous Optimization

I. INTRODUCTION

Many real-world application problems in engineering, science and technology can be formulated as continuous optimization problems (CnOPs) [1,2]. The continuous optimization problems always have local as well as global optima. Mostly, the user is interested in determining the global minima [3]. However, it is more difficult to determine the global minima rather than local minima for a lot of multimodal problems. As a result, most algorithms are very easy to be trapped in the local minima. Besides, for some non-separable benchmark problems, in view of the correlations information among the variables, the existing algorithms are difficult to converge to the global optimum, especially when the scale of the problem becomes larger [3,4].

Metaheuristics are a family of optimization techniques that have seen increasingly rapid development and have been applied to CnOPs over the past few years. Among them are genetic algorithm (GA) and ant colony optimization (ACO). GA has been used in many engineering applications since it was introduced as a robust and efficient search technique. The popularity of this method is based on simply solving multidimensional and multimodal optimization problems without requiring any additional information such as the gradient of an objective function. Although the origin of this method proposed binary number for encoding, over the past ten years, there were a surge of studies related to real-coded genetic algorithms (RCGA) for continue domain problem [3-6]. ACO is inspired by the ants’ foraging behavior and it was first applied to solve discrete optimization problems [7-9]. The ant colony optimization was extended to the continuous domains by Socha [2], called ACOx.

In this paper, we combine the conditionally breeding genetic algorithm model (CGA [4]) with ACOx[2] and develop a new hybrid algorithm for CnOPs. Several classical test problems available in the global optimization literature are used to test the performance of the proposed algorithm.

II. METHODOLOGIES FOR CnOPS

A. Conditionally Breeding Genetic Algorithms model

The original conditionally breeding genetic algorithms (CGA) is firstly defined in [10], in which crossover and mutation behaviors are performed by difference-degree between individuals instead of given probability. The CGA is a binary coded GA and thus it was applied to combinatorial optimization problems [12,13]. In the literature [4], we extend the CGA for continue optimization problems (CGAR). In the CGAR, an important parameter controlling crossover and mutation is called setting difference-degree $D_s$ ($0 < D_s < 1$).

$$D_s(t) = \mu D_s(t)$$

(1)

where $t$ expresses $t^{th}$ generation, $\mu \in (0,1)$ is a constant variable called cooling ratio. The difference-degree between individuals is calculated as follows: Given two chromosomes $x_1$ and $x_2$, the difference-degree between $x_1$ and $x_2$ is defined as follows:

$$d_j = \frac{||x_1 - x_2||}{2}$$

(2)

where $e_1 = x_i/||x_i||$, $e_2 = x_j/||x_j||$ and $||e_1 - e_2||$ is the distance between $e_1$ and $e_2$. The criterion of crossover and mutation in GA is by the difference-degree between or not by probability of conventional GA.
B. ACOR

The first algorithm that can be classified as an ACO algorithm for continuous domains is ACO$_R$[2]. In ACO$_R$, the discrete probability distributions used in the solution construction by ACO algorithms for combinatorial optimization are substituted by probability density functions (PDFs). ACO$_R$ uses a solution archive [14] for the derivation of these PDFs over the search space. Additionally, ACO$_R$ uses sums of weighted Gaussian functions to generate multimodal PDFs. ACO$_R$ initializes the solution archive with $k$ solutions that are generated uniformly at random. Each solution is a $D$-dimensional vector with real-valued components $x_i \in [x_{\text{min}}, x_{\text{max}}]$, with $i = 1, 2, ..., D$. In this paper, we assume that the optimization problems are unconstrained except possibly for bound constraints of the $D$ real-valued variables $x_i$. The $k$ solutions of the archive are kept sorted according to their quality (from best to worst) and each solution $S_j$ has associated a weight $\omega_j$. This weight $\omega_j$ is calculated using a Gaussian distribution as [15]:

$$\omega_j = \frac{1}{qk \sqrt{2\pi}} e^{-\frac{(\text{rank}(j) - 1)^2}{2qk^2}}$$

where $\text{rank}(j)$ is the rank of solution $S_j$ in the sorted archive, and $q$ is a parameter of the algorithm. By computing $\text{rank}(j)$, the best solution receives the highest weight. The weights are used to choose probabilistically a guiding solution around which a new candidate solution is generated. The probability of choosing solution $S_j$ as guiding solution is given by (2) [15]:

$$P_j = \frac{W_j}{\sum_{i=1}^{k} W_i}$$

So that the better the solution, the higher are the chances of choosing it. Once a guiding solution $S_{\text{guide}}^j$ is chosen, the algorithm samples the neighborhood of the $i$-th real-valued component of the guiding solution $S_{\text{guide}}^j$, using a Gaussian PDF with $\mu_{\text{guide}}^j$ and $\sigma_{\text{guide}}^j$ equal to

$$\sigma_{\text{guide}}^j = \frac{1}{k-1} \sum_{r=1}^{k} |S_{r}^i - S_{\text{guide}}^i|$$

which is the average distance between the value of the $i$-th component of $S_{\text{guide}}^j$ and the values of the $i$-th components of the other solutions in the archive, multiplied by a parameter $f$ [15]. The process of choosing a guiding solution and generating a candidate solution is repeated a total of $N_p$ times (corresponding to the number of “ants”) per iteration. Before the next iteration, the algorithm updates the solution archive keeping only the best $k$ of the $k+N_p$ solutions that are available after the solution construction process.

C. CGA and ACO-based Hybrid Approach

Our past researches showed CGA$_R$ is a genetic algorithm with excellent ability of global search. However, it does not provide a good mechanism to tune the near-optimal in promising space for some non-separable function and multimodal problems. To improve the performance of CGA$_R$ and balance between two contradictory aspects of their performance: exploration and exploitation, we utilize the exploiting mechanism of ACO$_R$ to develop a hybrid approach as Figure I. BLX-$\alpha$ crossover operator [16], and non-uniform mutation operator are used, which are as same as in the literature [4].

Algorithm:

Input Parameters : $N_P$, $N_C$, $D_h$, $m$, $D$, $\varepsilon$, $\cdots$

Initialize Population: $P = (P_1, P_2, \cdots, P_{NP})$

for $j = 1 : k$

for $i = 1 : D$

$P_j^i = \text{normrnd}(x_{\text{min}} + 2\varepsilon (m+j) L), L)$

end

end

while (termination criterion is not satisfied)

//Generate set G with $N_C$ new solutions using CGA$_R$

for $k = 1 : N_C$

Randomly Select a pair $p_i$ with two Solution from $P$;

Calculate difference-degree $d$ of pair $p_i$;

if $d > D_h$

Crossover are performed on pair $p_i$;

Store and evaluate newly generated solution into $G$; $N_C += 2$;

else

Mutation are performed on pair $p_i$;

end

end

Update population $P$ with the best $N_P$ solutions of $P+G$;

//Generate set M with $m$ new solutions using ACO$_R$

for $j = 1 : m$

Select solution $S_j$ from $P$ according to weights;

Generate a new solution based on (5);

Store and evaluate newly generated solution;

end

Update population $P$ with the best $N_P$ solutions of $P+M$;

Update $D_h$;

end

III. EXPERIMENT AND DISCUSSION

A. Experimental setup

In order to verify the effectiveness of the proposed algorithm, we use the following four test functions. Sphere function $f_1$ is the basic function to evaluate the algorithm [5]. For the non-separable function we choose Rosenbrock function ($f_2$) [17]. For the multimodal functions, the Schwefel function ($f_3$) and the Rastrigin function ($f_4$) are chosen. The dimensionality is set to 30 for all test functions.
1. Sphere function

\[ \min f(x) = \sum_{i=1}^{n} x_i^2, \]
\[-5.12 \leq x_i \leq 5.12, \quad x^* = (0,0,...,0), \quad f(x^*) = 0. \]

2. Rosenbrock function

\[ \min f(x) = \sum_{i=2}^{n} 100(x_i - x_{i-1}^2)^2 + (1-x_{i-1})^2, \]
\[-2.048 \leq x_i \leq 2.048, \quad x^* = (0,0,...,0), \quad f(x^*) = 0. \]

3. Schwefel problem

\[ \min f(x) = 418.9829 \cdot n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|}), \]
\[-500 \leq x_i \leq 500, \quad x^* = (420.97,420.97,...,420.97) \text{ and } f(x^*) = 0. \]

4. Rastrigin function

\[ \min f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i)), \]
\[-5.12 \leq x_i \leq 5.12, \quad x^* = (0,0,...,0), \quad f(x^*) = 0. \]

The parameters used in the proposed algorithm are as follows: population size \( N_p \), the number of offspring generated by CGA_R: \( N_C \), the number of offspring generated by exploit scheme: \( m \), the parameter \( \xi \) of standard deviation \( \sigma_f \), the setting difference-degree \( D_s \), and cooling ratio \( \mu \). In this work, all parameters are set as follows: \( N_p = 180 \), \( N_c = 30 \), \( m = 30 \), \( \xi = 0.76 \), \( D_s = 0.1 \) and \( \mu = 0.999 \).

B. Performance Evaluation

To investigate the performance of the proposed algorithm, the convergence properties on four typical functions are analyzed, in comparison with the CGA_R [3] and ACO_R [2]. As shown in Fig.2–5, the proposed hybrid approach can be fastest to find the global optimum than CGA_R and ACO_R.

For each test functions we performed 25 independent runs using each algorithm. The stopping criterions are as follows: \( |f(x) - f(x^*)| < 10^{-7} \) (\( x^* \) is the global optimal solution), and the maximum number of function evaluations (MaxFEs) is set to 4E+6. It means that if the error accuracy does not reach \( 10^{-7} \) within 4E+6 FEs, the simulation run is considered to an unsuccessful run. CGA_R+FPDD-LX [3] and the differential evolution (DE) [18] are employed to compare with the proposed algorithm. The CGA_R+FPDD-LX is another CGA_R with local search mechanism. The DE is the state-of-the-art algorithm that is useful for the real world application, and we select the classical DE approach called DE/rand/1 to compare with the proposed algorithm. The mean numbers of the FEs of the proposed algorithm is far fewer than CGA_R, ACO_R, DE/rand/1 and CGA_R+FPDD-LX.
TABLE I. COMPARING WITH OTHER ALGORITHMS, D = 30

<table>
<thead>
<tr>
<th>Func</th>
<th>This work</th>
<th>CGA_R</th>
<th>ACO_R</th>
<th>DE/rand/1</th>
<th>CGA_R+FPDD-LX</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_1</td>
<td>1.31e+4</td>
<td>1.36e+4</td>
<td>1.84e+4</td>
<td>4.39E+4</td>
<td>1.26E+4</td>
</tr>
<tr>
<td>f_2</td>
<td>4.48e+5</td>
<td>--</td>
<td>1.20e+6</td>
<td>--</td>
<td>3.40E+5</td>
</tr>
<tr>
<td>f_3</td>
<td>1.93e+4</td>
<td>--</td>
<td>2.54E+5</td>
<td>5.0E+5</td>
<td>7.43E+5</td>
</tr>
<tr>
<td>f_4</td>
<td>2.38e+4</td>
<td>2.27e+5</td>
<td>--</td>
<td>8.43E+4</td>
<td>2.40E+5</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

In this paper, we proposed an effective hybrid algorithm based on genetic algorithm and ant colony optimization for continuous optimization in continuous domains. To evaluate the proposed algorithm, we have carried out a lot of simulations on several benchmark problems. Simulation results showed the proposed scheme distinctly improved the performance of CGA_R and ACO_R, especially for the non-separable functions and multimodal functions. The proposed algorithm has been compared with some evolutionary algorithms. From the results, we can see that the proposed algorithm outperforms the other algorithms.

ACKNOWLEDGMENT

This work was supported by Natural Science Foundation Project of CQ CSTC (No.cstc2012jjA40041) and Science Research Fund of Chongqing Technology and Business University (No. 2011-56-05).

REFERENCES