A Clustering-based Adaptive Parameter Control Method for Continuous Ant Colony Optimization

Yue-jiao Gong, Rui-tian XU and Jun Zhang
Department of Computer Science
SUN yat-sen University
Guangzhou, P.R.China, 510275
junzhang@ieee.org

Ou Liu
The School of Accounting and Finance
The Hong Kong Polytechnic University
Kowloon, Hong Kong
Ou.Liu@inet.polyu.edu.hk

Abstract—Ant colony optimization (ACO) has been widely and successfully applied to NP-hard combinatorial optimization problems for its strong searching ability and robustness. Recently, several extended ACO algorithms have also been proposed to deal with continuous optimization problems. However, the ACO algorithms always have slow convergence speed and encounter premature convergence in engineering applications. This paper proposes a novel adaptive parameter control method for continuous ACO algorithms. Clustering analysis is used to judge the optimization state of the algorithm and the flexible adjustment of the parameters is based on these optimization states during the training process. As an example, the adaptive control method is used to improve the performance of the continuous orthogonal ant colony (COAC). Experimental results demonstrate that the clustering-based adaptive parameter control scheme contributes to both faster convergence speed and higher solution accuracy. The proposed adaptive control method has great practical value and bright prospect.

Keywords—ant colony optimization (ACO), continuous orthogonal ant colony (COAC), adaptive parameter, clustering analysis

I. INTRODUCTION

Ant colony optimization (ACO) was first proposed in 1990s for solving NP-hard combinatorial optimization problems. Because of its strong searching ability and robustness, ACO has been widely and successfully applied in solving problems such as the traveling salesman problem [1][2], scheduling problem [3][4], protein folding problem [5], and other problems [6][7]. Recently several extended ACO algorithms have also been proposed to deal with continuous optimization problems [8][9].

However, the ACO algorithms, especially the continuous ACO algorithms, always have slow convergence speed and encounter premature convergence. The parameter setting strategies for ACO algorithms are deficient. In order to overcome these shortcomings, many researchers have made great efforts to study on the parameter settings, and have proposed some adaptive parameter control schemes[10]-[13]. Dorigo and Gambardella [2] put forward a method for choosing the proper number of ants \( m \) based on the value of the pheromone decay parameter \( \rho \) and the probability for performing exploitation \( q_0 \) in their ant colony system. Zeng and He [10] designed a formula to reduce the \( q_0 \) during the training process in order to strengthen the random node selecting probability. Cai and Huang [11] proposed an automatic setting method to adjust the \( \rho \) and the relative weight of heuristic value \( \beta \) according to the amount of pheromone and the solution value. In the paper of Qin et al. [12] the adjustment of the relative weight of pheromone trail \( \alpha \) and \( \beta \) is based on a pheromone distributing weight.

However, most of the previous studies only concentrate on finding the parameter control strategies based on the conventional ACO which deals with the combinatorial optimization problems. They are not applicable for continuous ant colony algorithms. Moreover, almost all the previous schemes use deterministic strategies to adjust the parameters. They are not adaptive to different evolutionary environments, resulting in inefficient optimization when dealing with complex problems. Therefore, research into adaptively controlling the parameters of ACO, especially the continuous ACO, has become significant and promising.

In this paper, a clustering-based adaptive parameter control scheme for continuous ant colony algorithms is proposed. Clustering analysis [14] is used to judge the optimization state of the algorithm. Flexible adjustment of the parameters is based on the optimization states during the training process. The proposed scheme has advantages of robustness, versatility and effectiveness. As an example, the scheme is used to improve the performance of the continuous orthogonal ant colony (COAC) [9], and the experimental results show that it contributes to the COAC with faster convergence speed and higher solution accuracy.

The rest of this paper is organized as follows. The next section presents brief review on the COAC. In Section III, we introduce the clustering-based adaptive parameter control scheme in detail. Then, in Section IV, seventeen benchmark functions are tested, experimental results are reported. At last, conclusions are drawn in Section V.

II. BRIEF REVIEW ON THE COAC ALGORITHM

The COAC was proposed by Hu et al. [9] as an extension of the ACO algorithm in order to solve the continuous optimization problems. In COAC, the continuous domain is discretized into \( \mu \) regions, \( m \) ants are distributed to several of them. The principle of the problem is how to direct these ants to find the best region and the optimal point.
In COAC, every region has four properties, the coordinate of its center, the searching radiuses, the amount of pheromone, and the ranks of its desirability [9]. It works as follows: first, μ regions are randomly generated; m ants are randomly positioned on these regions. Then, each ant chooses its next region according to the amount of pheromone, and performs orthogonal exploration in the region. Once all the ants have finished a round of state transition and orthogonal exploration, the pheromone on the visited regions is updated by applying a global modulation rule. An overall flowchart of COAC is illustrated as Fig. 1, where MAXFES is predefined to control the number of iterations.

**A. State Transition**

Each ant chooses its next region for exploration by applying the state transition rule. That is, the next region $A$ according to the amount of pheromone, and performs orthogonal exploration in the region. Once all the ants have finished a round of state transition and orthogonal exploration, orthogonal exploration in the region. Once all the ants have finished a round of orthogonal exploration, an overall flowchart of COAC is illustrated as Fig. 1, where MAXFES is predetermined to control the number of iterations. The state transition rule makes ants tend to search the best region in the whole domain (biased exploration). An overall flowchart of COAC is illustrated as Fig. 1, where MAXFES is predefined to control the number of iterations.

**B. Orthogonal Exploration**

When an ant reaches a region, it performs orthogonal exploration to search better points in the region. The coordinate and radius of the region are updated according to the exploration result.

Use an orthogonal array $L(N, k, s)$, where $N$ is the number of orthogonal combinations which stands for how many points in the region are explored, $k$ is the number of orthogonal factors which must satisfy $k \geq n$ (in order to choose $n$ columns without duplication as a new orthogonal array), and $s$ stands for each factor has $s$ levels. Here $n$ is the dimension of the domain.

Suppose the current center of the region is $x^m = (x_{11}, x_{12}, \ldots, x_{1n})$, and the current radius is $r^m = (r_1, r_2, \ldots, r_n)$. The orthogonal exploration for an ant is defined as follows [9]:

**Step 1:** Randomly choose $n$ different columns in $L(N, k, s)$ as a new orthogonal array.

**Step 2:** Generate $N$ neighboring points $x^i_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$, $i = 1, 2, \ldots, N$ by the orthogonal design method where

$$x^i_j = x^i_k + \left[ \frac{L_{jk} - 1}{s - 1} \right] r^i_j \cdot \text{rand} j,$$

if $x^i_j < \text{lbound}_k$, $x^i_j = \text{lbound}_k,$ .

if $x^i_j > \text{ubound}_k$, $x^i_j = \text{ubound}_k,$ .

$L_{jk}$ is the level of the $k$-th column in combination $i$ for the chosen orthogonal array, and $\text{rand} j$ is a uniform random number in $[0, 1]$. Once a better point is found, move the center coordinate of this region to the point, so that more optimal region can be obtained.

**Step 3:** Adjust the radius by applying the rule given below.

If a better point has been found, expand the radius of the region to search for more diversity [9].

$$r^i_j = r^i_k \cdot \text{shrink}, \quad k = 1, 2, \ldots, n.$$

Otherwise, if no better point is found by the orthogonal exploration, shrink the region to achieve a higher sensitivity [9].

$$r^i_k = r^i_k \cdot \text{shrink}, \quad k = 1, 2, \ldots, n.$$

Here shrink is a pre-defined parameter in (0, 1).

**C. Global Modulation**

After all $m$ ants have finished this round of state transition and orthogonal exploration, a global modulation rule is proposed to update the pheromone value for each region in the domain. The desirability of every region is judged, some undesirable regions are discarded.

The best region in $S_n$ is assigned with rank 1, then, the second-best region is assigned with rank 2, and so on, until $\mu$ regions are marked, where $\phi \in [0,1]$ is the elitist rate. In each iteration, only $\phi \cdot \mu$ regions are reserved, and the pheromone on the selected region $j$ is updated by

$$\tau_j = (1 - \alpha) \cdot \tau_j + \alpha \cdot T_0 \cdot (\phi \cdot \mu + 1 - \text{rank}_j + \text{visit}_j),$$

where $\tau_j$ is the pheromone value in region $j$, $\text{rank}_j$ is the rank of the visited region, and $\text{visit}_j$ is the number of iterations. The desirability of every region is judged, some undesirable regions are discarded.

The state transition rule makes ants tend to search the regions with large amount of pheromone.
where $\alpha \in (0, 1)$ is the pheromone decay parameter, $T_0$ is the initial pheromone value and $visit_j$ is the number of times that region $j$ has been visited. The better the region is and the more times it has been visited, the more pheromone is deposited in the region [9].

At last, other unmarked regions are re-initialized.

Although the COAC is useful for discovering the optimal solution for continuous problems, the total iteration number required to find the optimum is unsatisfactory. Therefore, we devise an adaptive parameters control method to improve its performance, which is presented in the next section.

III. ADAPTIVE PARAMETERS CONTROL METHOD

In conventional ACO, the parameter values such as $q_0$ and $\alpha$ are fixed. However, the requirement of these values depends on the optimization state during the convergence process, so that the training efficiency by using these fixed parameters may deteriorate. Moreover, for different cases, different parameter settings are required. It brings in a major burden of specifying the value of these parameters. In order to overcome these shortcomings, using adaptive parameter control schemes are promising.

In this paper, a clustering-based adaptive parameter control method is proposed to improve the efficiency of continuous ant colony algorithms, and the COAC is taken as an example. The adaptive process is partitioned into two parts: the clustering to judge the optimization state and the adjustment of the parameters. The procedures are described as follows.

A. Judging the Optimization State

In COAC, each region stands for a candidate solution, therefore the training process of the algorithm is the process of all the regions in the domain gathering to the region which represents the optimum solution. Thus, the optimization state can be judged by measuring the current distribution of regions.

By applying a $K$-means algorithm [14], the regions in the domain is clustered into several groups according to the distribution information in each iteration. The clustering process is defined as below, here $K$ is the number of clusters.

**Initialize** $K$ clusters $C_1, C_2, ..., C_K$.
Each cluster is associated with a cluster center $c_j (j = 1, 2, ..., K)$.

**Loop**
For each region $p_i$ (1, 2, ..., $\mu$) in the domain, calculate the distance between $p_i$ and $c_j (j = 1, 2, ..., K, j \neq i)$, then assign $p_i$ to its closest cluster.
For each cluster $C_j$, count the number of regions $n_j$ and compute new cluster center by:

$$c_j = \frac{1}{n_j} \sum_{p \in C_j} p_i, \quad j = 1, 2, ..., K.$$ (7)

**Until** End condition.

After the clustering process, the cluster containing the best region is defined as $C_{\text{best}}$, while the cluster containing the worst region is defined as $C_{\text{worst}}$. Based on the size of $C_{\text{best}}$ and $C_{\text{worst}}$, the convergence process is divided into three optimization states, as Table I shows. If $C_{\text{worst}}$ is the largest cluster, the current process is considered to be in the initial state. Otherwise, if $C_{\text{best}}$ is the largest cluster, the current process is considered to be in the matured state. Otherwise, if $C_{\text{best}}$ and $C_{\text{worst}}$ are both small, the current process is considered to be in the maturing state.

<table>
<thead>
<tr>
<th>Optimization State</th>
<th>$q_0$</th>
<th>$\varphi$</th>
<th>shrink</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial State</td>
<td>$+$</td>
<td>$-$</td>
<td>$-0.3$</td>
</tr>
<tr>
<td>Maturing State</td>
<td></td>
<td>$+$</td>
<td></td>
</tr>
<tr>
<td>Matured State</td>
<td>$+$</td>
<td>$-0.4$</td>
<td>$-0.3$</td>
</tr>
</tbody>
</table>

B. Adjustment of the Parameters

Adjustment of the parameters is based on the training state. In COAC, there are four parameters including $q_0$, $\alpha$, $\varphi$ and shrink can be adjusted. $q_0$ determines the probability of exploitation towards biased exploration. $\alpha$ is the pheromone decay parameter. $\varphi$ is the elitist rate, in each iteration only $\varphi \cdot \mu$ regions are reserved. shrink is a parameter that controls the expansion and shrinking rate of the regions. In this paper, the adaptive control strategies for the parameters $q_0$, $\alpha$, $\varphi$ and shrink are proposed. The value of $\alpha$ is fixed and equal to 0.1, because experimental studies show that, the value of $\alpha$ has little impact on the efficiency in COAC.

Table II and Fig. 2 depict the adaptive control schemes for different optimization states. Constantly adjust the control parameters according to the optimization state makes the parameters meet the need of the evolutionary process better. The searching course of the algorithm can be more flexible and intelligent.

![Figure 2. Adaptive Control Scheme](image-url)
In the initial state, good individuals in the population are very few, the searching direction is determined. In order to search in the whole domain and establish an initial searching direction, the value of \( q_0 \) and the \( \phi \) are reduced, and the shrink is set small to increase the flexibility of the orthogonal exploration.

In the maturing state, the number of good individuals in the population is a little more. So, increase the value of \( q_0 \) step by step to guide the search. However, the value of the \( \phi \) should still be reduced in order to prevent the COAC from getting trapped at a local optimum. Because in this optimization state the probabilities of finding better points during the orthogonal exploration are very high, shrink should be increased in order to keep the search steady.

In the matured state, most of the regions in the domain have swarmed together near the currently best region, the searching direction is determined. So increase the value of \( q_0 \) and the \( \phi \) in order to speed up the convergence. Reduce the shrink to make the final orthogonal search more diverse.

These measures can realize the twin goals of maintaining diversity in the domain and enhancing the convergence capability. The comparisons and analysis between the COAC with and without the proposed adaptive method are presented in the next section.

### IV. EXPERIMENTAL STUDIES

In this paper, an adaptive parameters control scheme for continuous ant colony algorithms based on clustering analysis is proposed. As an example, it is used to improve the performance of the COAC algorithm [9]. In the following, the clustering-based adaptive COAC is abbreviated to CCOAC. In order to compare the performance of CCOAC with COAC, fourteen benchmark functions [14] are used.

#### A. Benchmark Functions

The fourteen benchmark functions are given in Table III. \( f_1 \)-\( f_4 \) are unimodal functions. \( f_5 \)-\( f_{14} \) are multimodal functions, among which \( f_8 \) are rotated multimodal functions. The dimensions of these functions are all set as four.

The parameter settings of COAC are the same as [9]. In the CCOAC, the number of regions \( \mu \) and the number of ants \( m \) are set as 30 and 20 for all the fourteen functions, other parameter settings are the same as COAC.

For each function, 100 independent trials are carried out by applying the COAC and CCOAC under the same circumstances.

<table>
<thead>
<tr>
<th>Function</th>
<th>Domain</th>
<th>Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_i(x) = \sum_i x_i^2 )</td>
<td>([-100,100]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left</td>
<td>x_i \right</td>
<td>+ \sqrt{\sum_i \left</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \sum_j x_i x_j )</td>
<td>([-100,100]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ 100(x_{i-1} - x_i)^2 + (x_i - 1)^2 \right] )</td>
<td>([-100,100]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i (x_i + 0.5)^2 )</td>
<td>([-100,100]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i x_i^4 )</td>
<td>([-1.28,1.28]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(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 + \epsilon )</td>
<td>([-32.0],1.28]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ 10 \sin^3(3\pi x_i) + \sum_{j \neq i} \left( x_j - 1 \right)^2 \left[ 1 + \sin^2(3\pi x_j) \right] \right] + \frac{\mu}{100} \sum_i x_i^4 )</td>
<td>([-50,50]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = 0.1 \sin^3(3\pi x) + \sum_{i=1}^n (x_i - 1)^2 \left[ 1 + \sin^2(3\pi x_i) \right] + \frac{\mu}{100} \sum_i x_i^4 )</td>
<td>([-50,50]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ 1 + \left( \frac{x_i}{b} + 1 \right) \left( \frac{x_i}{b} + 1 \right)^2 \right] )</td>
<td>([-5,5]^n)</td>
<td>(3.075 \times 10^{-4})</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ (x_i - a_i)^2 \right] c_i )</td>
<td>([0,10]^n)</td>
<td>(-10.1532)</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ (x_i - a_i)^2 \right] c_i )</td>
<td>([0,10]^n)</td>
<td>(-10.409)</td>
</tr>
<tr>
<td>( f_i(x) = \sum_i \left[ (x_i - a_i)^2 \right] c_i )</td>
<td>([0,10]^n)</td>
<td>(-10.5664)</td>
</tr>
<tr>
<td>( f_i(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n y_i^2}) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi y_i\right) + 20 + \epsilon )</td>
<td>([-32.0],1.28]^n)</td>
<td>0</td>
</tr>
</tbody>
</table>
B. Experimental Results and Comparison

In Table IV, the solution accuracy compared between COAC and CCOAC are listed. It can be seen that to most of the fourteen functions, both the COAC and the CCOAC can find the optimum in all trials. In solving \( f_4 \), \( f_7 \) and \( f_{10} \), the CCOAC can achieve better mean values, and in \( f_4 \), \( f_7 \), \( f_{10} \), \( f_{11} \) and \( f_{13} \), it can achieve better standard deviation. The CCOAC provides higher global optimization ability and stability than the COAC. Especially in solving \( f_4 \), the differences are significant.

Fig. 3 illustrates the mean convergence characteristics during the training in solving \( f_1 \), \( f_4 \), \( f_7 \), \( f_{10} \), \( f_{13} \) and \( f_{14} \). The vertical axis is the mean function value of 100 trials, and the horizontal axis is the number of function evaluations. Fig. 3 shows that the CCOAC can provide faster convergence speed as well as higher solution accuracy than the COAC.

The comparison results of the convergence speed are listed in Table V. Error is defined to judge whether the optimization meets the precision requirement. During the optimization process, if the absolute value of the current best minus optimum is smaller than the value of Error (\(|\text{best} - \text{optimum}| < \text{Error}\)), the trial is regarded as successful and the “errFES” of this trial is obtained. The column of “errFESs” lists the average function evaluations in obtaining the Errors in the 100 trials. The column of “stopFESs” lists the average function evaluations in obtaining the last optimal results in the 100 trials. The column of “%ok” lists the successful rates within the Errors.

As shown in Table V, the improvement of the convergence speed is remarkable. The errFESs in CCOAC is smaller than that in COAC when solving all the fourteen functions, the stopFESs in CCOAC is smaller than that in COAC only except for solving \( f_{11} \) to \( f_{15} \). The result shows that the CCOAC has extraordinary faster convergence speed for almost all the functions than the COAC. The successful rates in CCOAC are all 100%, which means the encounter premature phenomenon does not occurred when applying the CCOAC to solving the fourteen functions.

![Figure 3. Convergent Curves of COAC and CCOAC](image-url)
The experimental results and comparisons demonstrate that the clustering-based adaptive parameters control scheme contributes to both higher solution accuracy and faster convergence speed to continuous ant colony algorithms.

V. CONCLUSIONS

In this paper, an adaptive parameters control method for continuous ant colony algorithms has been proposed. Clustering analysis is used to judge the optimization state and the adjustment of the parameters is based on the optimization states during the training process.

As an example, the clustering-based adaptive parameter control method is used to improve the performance of the continuous orthogonal ant colony (COAC). From the experimental results, it can clearly be seen that the method not only improves the accuracy and stability of the algorithm, but also improve the convergence speed.

The proposed clustering-based adaptive parameter control method has practical value and bright prospect. Future work is applying the method to conventional ACO algorithms for solving combinatorial optimization problems.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China under Project 60573066, in part by the NSF of Guangdong under Project 5003346, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, China, in part by the National Natural Science Foundation of China Joint Fund with Guangdong under Key Project U0835002, in part by the National High-Technology Research and Development Program of China, No. 2009AA01Z208. Jun Zhang is the corresponding author, e-mail: junzhang@ieee.org.

REFERENCES


